

Universidad de las Ciencias Informáticas
FACULTAD 10.



**Título: “Análisis y diseño del sistema
nacional de programación cultural”**

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

Autores: Yusliel García Vázquez.

Luis Domínguez Cruz.

Tutor: Lic. Luis Joaquín Guzmán Hernández.

Junio de 2007

**Cuando la verdad sea demasiado débil para
defenderse tendrá que pasar al ataque.
Bertolt Brecht**

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Luis Dominguez Cruz

Yusliel García Vázquez

Luis Guzmán Hernández

Datos de Contacto

Lic. Luis Joaquín Guzmán Hernández

Graduado de Licenciatura en Ciencias de la Computación en la Universidad Central "Martha Abreu" de Las Villas en el año 2002. Trabajó durante dos años en el Centro de Información Científica de la Universidad Central como especialista en informática. Actuó como líder en varios proyectos de la rama de la gestión de información. En el año 2005 comienza a trabajar en la UCI, desempeñándose como especialista en una dirección de la infraestructura productiva. A participado en muchos proyectos de desarrollo de software, fundamentalmente en el desarrollo de aplicaciones para WEB. Actualmente se desempeña como Director de la dirección de producción # 2 de la Infraestructura Productiva.

Agradecimientos

Luis:

A Aida mi mamá cuyos abnegados esfuerzos y sacrificios me han hecho llegar donde estoy.

A José Antonio mi hermanito, por quererme tanto aunque solo lo pueda ver dos veces al año.

A Yanedi mi novia por todo lo que me ha aguantado estos años.

A Nico mi hermano por ser buen amigo y por la confianza que siempre ha tenido en mí.

A Misleidys mi cuñada por acogerme en su casa, por ser de la familia.

A Yasmín mi hermana por todas las cosas buenas que ha hecho por mí.

A Yusliel por la paciencia que ha tenido conmigo.

A Luis Guzmán por su participación.

A mis abuelos por la dedicación de toda la vida.

A Fidel por dejarnos ser parte de sus sueños hechos realidad.

A Maidelys por no dejarnos tirados cuando nos hizo falta.

A Pla y Damian mis amigos de toda la carrera, los que siempre molesté por cualquier cosa.

También quiero expresar agradecimientos a todas las amistades el aula del apartamento a todos los que nunca dudaron de mí y que siempre estarán ahí cuando los necesite.

Yusliel

A mi mamá, mi hermana y mi papá, por apoyarme y haber dado siempre lo mejor de sí para que pudiera llegar hasta aquí, por enseñarme que no hay nada más importante que la familia, por creer siempre en mí, por estar siempre ahí cuando los necesito; por hacerme saber, aunque a veces no se los demuestre, cuan importantes son para mí.

A mis tíos Orlanda y Carito, por apoyarme en mis decisiones, por haber confiado siempre en mí, por guiarme siempre por el camino correcto, por poder contar con ellos cuando los he necesitado, por quererme como un hijo.

A mi familia por quererme y preocuparse por mí y por mi carrera.

A Mairenys mi novia por soportarme durante tanto tiempo, por dedicarme su tiempo y amor, por enseñarme a valorar la cosas que tengo a mi alrededor, por haber sabido introducirse en mi corazón.

A la Revolución y nuestro Comandante Fidel, por haberme dado la oportunidad de formar parte de esta Universidad.

A la UCI, por haberme formado como profesional.

A Luis Guzman, por su ayuda en mi carrera y por haber aceptado ser tutor del presente trabajo.

A mis amigos de La Palma, por los momentos alegres que hemos pasado juntos.

A mis compañeros de aula y de apartamento, aunque no los mencione por nombre, porque los tendría que mencionar todos, han sido muy importantes en mi vida. A ellos por los consejos, las ayudas, el apoyo, por hacerme sentir como en familia aún estando lejos de la mía.

A Luis Domínguez mi compañero de tesis, por su empeño y dedicación en la elaboración de la misma.

En fin a todos los que me han dado su mano cuando la he necesitado, a todos sinceramente **muchas gracias**.

Dedicatoria

Luis:

A la memoria de **Nicolás Roque Álvarez** mi papa quien estaría muy orgullo de verme graduado.

A **Aida Ester Cruz Pozo** mi mamá, por el cariño y la dedicación de toda la vida.

Yusiel

A mi mamá, a mi hermana y a mi papá.

A mis tíos **Orlinda y Carito**.

Resumen

Este trabajo propone una solución desde el punto de vista de análisis y diseño en forma de un sistema que automatice los procesos de programación cultural del departamento de programación del Ministerio de Cultura que actualmente no cuentan con un soporte informático. Sobre todo los procesos más prioritarios entre los que se encuentran los de solicitud de agrupación, la programación desde la base y la planificación de giras nacionales de las diferentes categorías de talento artístico con que cuentan las instituciones asociadas. (ICM, CNAE, ICL, ICAIC), los procesos de información que se llevan a cabo alrededor del departamento se solucionarían con un sistema informático que garantice mayor eficiencia y control.

Por tanto el objetivo concreto de este trabajo consiste en desarrollar el análisis y el diseño una aplicación Web, segura y de interfaz amigable, que dé solución a los problemas presentados en el departamento en cuanto a los informes de programación que llegan al MINCULT desde la base, la forma y velocidad con que viaja la información, también debe encargarse de la gestión de solicitudes de talento artístico por parte de los municipios y la planificación de presentaciones de talento artístico de primer nivel en los municipios.

En el presente documento se recogen los resultados del trabajo investigativo realizado. Se identifican, describen y mejoran los procesos del departamento de programación del MINCULT, especialmente aquellos que se van a automatizar.

Tabla de contenido.

Introducción	7
Capítulo1: Fundamentación teórica	11
1.1 Introducción	11
1.2 Gestores de base de datos	11
1.3 Lenguajes de programación.....	12
1.4 Tecnologías del lado del cliente.....	14
1.5 ¿Qué es el software libre?	15
1.5.1 ¿Qué le depara a Cuba el camino del software libre?	15
1.6 Metodologías de desarrollo de software	16
1.7 Propuesta.....	18
1.8 Conclusiones	19
Capítulo 2: Características del Sistema.....	20
2.1 Introducción	20
2.2 Fundamentación de la situación problemática.....	20
2.3 Modelo del negocio.....	21
2.3.1 Modelo de casos de uso	21
2.3.2 Actores del negocio.....	22
2.3.3 Trabajadores del negocio.....	23
2.3.4 Casos de uso extendidos.....	24
2.3.4.1 CU Crear programación	24
2.3.4.2 CU Solicitar agrupación	26
2.3.4.3 CU Programar fiestas populares.....	27
2.3.4.4 CU Programar Exposiciones.....	29
2.3.4.5 CU Programar giras nacionales (Música)	29
2.3.4.6 CU Programar giras nacionales (Artes escénicas)	29
2.3.4.7 CU Crear programación cinematográfica.....	29
2.4 Requisitos	29
2.4.1 Requisitos funcionales	29
2.4.2 Requisitos no funcionales	32
2.5 Definición de los actores y casos de uso del sistema.....	33
2.5.1 Listado de actores del sistema.....	33
2.5.2 Listado de casos de uso del sistema	34
2.6 Diagrama de casos de uso del sistema	39
2.7 Casos de uso expandido	40
2.7.1 CU Autenticar usuarios	40
2.7.2 CU Gestionar unidades artísticas	42
2.7.3 CU Gestionar espacios culturales.....	46
2.7.4 CU Gestionar actividades culturales	46
2.7.5 CU Planificar fiestas populares	46
2.7.6 CU Planificar giras musicales	49
2.7.7 CU Planificar proyección cinematográfica	49
2.7.8 CU Planificar giras CNAE	49

2.7.9 CU Mostrar vistas.....	49
2.7.10 CU Mostrar estadísticas.....	50
2.7.11 CU Administrar usuarios	52
2.7.12 CU Administrar roles	55
2.7.13 CU Administrar permisos	56
2.8 Conclusiones.....	56
Capítulo 3: Análisis y diseño del sistema.....	57
3.1 Introducción	57
3.2 Modelo de análisis	57
3.2.1 Diagramas de clases de análisis.....	57
3.2.1 Diagramas de interacción	62
3.3 Modelo de diseño.....	68
3.3.1 Diagrama de clases de diseño.....	68
3.3.1.1 Descripción de las clases de diseño	73
3.3.2 Diseño de la Base de Datos.....	78
3.3.2.1 Diagrama de clases persistentes	78
3.3.2.2 Modelo de datos.....	80
3.3.2.3 Descripción de las tablas	81
3.4 Patrón de diseño utilizado.....	94
3.5 Interfaces.	95
Conclusiones	96
Recomendaciones	97
Referencias Bibliográficas	98
Bibliografía	99
Glosario de Términos.....	100

Introducción

El departamento de programación del Ministerio de Cultura se encarga de llevar los registros de las actividades culturales programadas en cada municipio, programar las presentaciones del talento artístico de primer nivel y llevar un registro con información sobre los espacios culturales de cada consejo popular. Entre sus responsabilidades está también contabilizar las actividades, mantener informadas a las instituciones nacionales, encargarse de que a cada provincia llegue una programación balanceada, atender las solicitudes de los municipios y que cada institución nacional agregue a esta programación el talento que controla. ¿Cómo están funcionando estos procesos actualmente?

A nivel municipal el programador diseña la programación cultural con el talento del municipio y el que le es asignado, esta información llega a la provincia en un documento que se envía por correo o se lleva personalmente. Para solicitar una agrupación musical desde el municipio el programador tiene que seguir una serie de pasos que van desde enviar por medio de un documento la solicitud o llamar por teléfono hasta esperar a que le sea asignada o no y se le avise por teléfono o correo postal. Cuando el Instituto Cubano de la Música o el Centro Nacional de Artes Escénicas conforman la distribución de giras nacionales del talento que controlan se ven obligadas a informar a las provincias y municipios por teléfono, correo postal u otro medio, y en esta situación están todas las instituciones y el departamento de programación. Los informes de programación que tienen que tributar las provincias al MINCULT viajan de la misma manera que todas las solicitudes y planificaciones, esto conlleva a que se comentan muchos errores de estimación y contabilización de actividades, sobre todo en verano que los informes se rinden con mayor frecuencia. También ocurre que cuando los indicadores que le interesan al MINCULT cambian, hay que crear nuevos modelos para los informes a mano.

Por todo lo planteado el departamento de programación ha tomado la decisión de automatizar los procesos que tienen lugar tanto dentro de sí como en sus alrededores (las demás instituciones), con vistas a obtener mejoras en el control y la planificación de las actividades culturales buscando eficiencia y control en el trabajo: eficiencia, porque un sistema automatizado facilitaría el trabajo hasta tal punto que se haría mucho más rápido; y control, porque toda la información quedaría almacenada en un historial, todas las actividades quedarían registradas.

Por lo tanto **el problema** a resolver está formulado dentro de la siguiente interrogante: ¿Cómo facilitar los procesos de solicitudes, planificación, información y control existentes en el departamento de programación del MINCULT y las instituciones nacionales?

El **objeto de estudio** lo constituyen los procesos de planificación, información, solicitudes y control llevados a cabo en el departamento de programación del MINCULT y las instituciones nacionales (Instituto Cubano de la Música, Centro Nacional de Artes Escénicas, Instituto Cubano del Libro) y el departamento de estadística del MINCULT.

El **campo de acción** de este trabajo abarca al departamento de programación e instituciones relacionadas con el MINCULT que necesitan que se automaticen sus procesos de trabajo.

El **objetivo general** de esta investigación es modelar un sistema informático para la automatización de los procesos de planificación, información y control existentes en el departamento de programación del MINCULT y las instituciones relacionadas con éste.

A raíz de problema planteado surgen algunas **interrogantes** las cuales son:

- ¿Cómo comprender los procesos de planificación, información y control que existen en el departamento de programación del MINCULT?
- ¿Qué tecnologías son más usadas actualmente para la solución de problemas de gestión de información de actividades culturales?
- ¿Qué tecnologías se pueden escoger para dar solución al problema de gestión de información de actividades culturales?
- ¿Cómo almacenar de forma centralizada la información que se maneja en el departamento de programación?

De esto se derivan los siguientes **Objetivos específicos**.

- Diseñar un módulo para gestionar los datos del talento artístico permitiéndole al usuario guardar, editar, eliminar y ver los datos referentes al mismo para saber con precisión a que provincia o municipio pertenecen, centro cultural en el que radican, en que manifestación artística se desempeñan, etc.

- Diseñar un módulo para gestionar los datos referentes a los espacios culturales, para permitir al usuario guardar, editar, eliminar y ver los datos de estos y saber con precisión si el espacio está cerrado o abierto, a qué tipo de público está destinado, qué agrupaciones se presentan, qué capacidad tiene, etc.
- Diseñar un módulo para gestionar las solicitudes de talento artístico por parte de los municipios, que permita a los usuarios ver si una agrupación está disponible o no, hacer la solicitud y obtener respuestas en un tiempo sumamente corto.
- Diseñar un módulo para gestionar la planificación de giras nacionales, fiestas populares y proyecciones cinematográficas, desde las entidades nacionales hacia los municipios.
- Diseñar un módulo capaz de gestionar los datos referentes al público al que están destinadas las actividades, que le permitan al usuario conocer cuánto público de cada categoría (niños, adolescentes, jóvenes, adultos, adulto mayor) se presenta en cada actividad para poder hacer conclusiones cualitativas.
- Diseñar un módulo que permita la gestión de la programación en todas sus aristas desde la base hasta el MINCULT, este debe permitir además que el programador del municipio conforme la programación de forma fácil y eficiente.
- Diseñar un generador de reportes dinámico que permita al usuario obtener la información que desee independientemente de sean cuales sean los indicadores requeridos en los informes de salida, además debe devolver el reporte en un fichero con formato PDF.

Para la realización de este sistema se plantearon las siguientes **tareas**:

- Reuniones con el cliente.
- Estudio y descripción de los procesos de planificación, información y control existentes en el Departamento de programación del MINCULT y las instituciones nacionales.
- Análisis del estado actual de las tecnologías en el marco internacional, que se utilizan para desarrollar sistemas como el que se desea.

- Selección de la tecnología, lenguaje y herramientas a utilizar, fundamentando con investigación la elección.
- Selección de una metodología de análisis y diseño de sistemas que facilite la creación y garantice la calidad de la aplicación.
- Diseño una base de datos que satisfaga las funcionalidades del sistema.
- Diseño de una aplicación Web segura con interfaz amigable que facilite los procesos de solicitudes, planificación, información y control del departamento de programación del MINCULT y las instituciones nacionales.

El contenido de este documento esta estructurado en 3 capítulos de la forma siguiente:

- Capítulo1: “Fundamentación teórica”, donde se incluyen los aspectos teóricos que soportan el trabajo.
- Capítulo2: “Características del sistema”, donde se plantea el problema y la situación problemática, el objeto de automatización, se especifica la información que se maneja, la propuesta del sistema, modelo del negocio y los requisitos funcionales del sistema.
- Captulo3: “Análisis y diseño del sistema”, donde se especifican las clases que se utilizan en la solución y la relación entre ellas, se define todo el proceso de obtención de la base de datos y se especifican los diagramas de interacción por cada caso de uso.

Capítulo 1: Fundamentación teórica

1.1 Introducción

En el presente capítulo se hace un análisis de las tecnologías actuales a nivel mundial que se pueden considerar en la propuesta de la solución del problema. Se tienen en cuenta los lenguajes de programación Web, los Sistemas Gestores de Bases de Datos más utilizados a nivel internacional, así como las tecnologías del lado del cliente.

Finalmente se seleccionan los más convincentes que garanticen el cumplimiento de los intereses de los usuarios finales y del departamento de Programación Cultural del Ministerio de Cultura.

1.2 Gestores de base de datos

Los Sistema de gestión de bases de datos están representados por software capaces de manejar un número grande de ficheros que componen las bases de datos, sirven de interfaz entre la base de datos y el usuario. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. A continuación un análisis de 4 de los principales exponentes de los SGDB. (CONFERENCIAS 2005)

Mysql

Sistema multiplataforma que ha atraído a muchos programadores por su sencillez, distribuido bajo la licencia GNU GPL es uno de los más usados, posee un amplio subconjunto del lenguaje SQL, se usa en aplicaciones tanto pequeñas como enormes, se conecta de forma segura. Proporciona sistemas de almacenamientos transaccionales y no transaccionales. Usa tablas B-tree muy rápidas con compresión de índice, tablas hash en memoria, que son usadas como tablas temporales; además es soportado por varios lenguajes de programación como: PHP, Java, Perl, Python, entre otros. (VAZQUEZ 2003)

PostgreSQL

Multiplataforma, con soporte par todas las características profesionales de una base de datos seria (objetos de uso común, orientación a objetos, eventos en los objetos de la base de datos, etc.), lenguajes mas populares del medio; PHP, C, C++, Java, Perl, Python, Ruby, soporta además protocolo de comunicación encriptado por SSL, su principal dificultad esta en la velocidad de conexión que no es la más rápida. (CALDERON 2003)

Microsoft SQL Server

Facilidad de gestión es lo primero que caracteriza a SQL Server, integra en la misma instalación el gestor del cliente y el gestor de consultas, permite compartir datos a través de múltiples plataformas, aplicaciones y dispositivos, desarrollar e implantar aplicaciones empresariales escalables, fiables y seguras se distribuye bajo las licencias privadas de Microsoft, esta licencia es difícil de obtener para un país como Cuba.

Oracle

Su mayor defecto es su enorme precio, que es de varios miles de euros. Otro aspecto que ha sido criticado por algunos especialistas es la seguridad de la plataforma, y las políticas de suministro de parches de seguridad, modificadas a comienzos de 2005 y que incrementan el nivel de exposición de los usuarios. En los parches de actualización provistos durante el primer semestre de 2005 fueron corregidas 22 vulnerabilidades públicamente conocidas, algunas de ellas con una antigüedad de más de 2 años. Por lo demás es un potente gestor con un rápido motor de búsqueda, es muy estable y multiplataforma.

1.3 Lenguajes de programación

Java Server Pages (JSP)

Esta tecnología nos permite crear páginas Web dinámicas programadas en Java, quien a su vez presenta la característica de ser multiplataforma. Las paginas creadas con JSP están compuestas de un código HTML/XML que se mezcla con las etiquetas que definen la programación de los scripts de java dentro de nuestra Web. Estas páginas presentan un motor que está basado en los servlets de Java que están destinados a ejecutarse en el servidor. Mediante el uso de JSP podemos reutilizar componentes de Java Beans, lo que hace que se optimice considerablemente la utilización de recursos en el servidor. También podemos embeber lenguajes de marca más complejos como XML para acceder a diferentes componentes o para levantar applets en los clientes.

Practical Extracting and Reporting Language (Perl)

Perl es utilizado para la construcción de aplicaciones CGI para la Web. Es multiplataforma y de libre uso. Perl es muy práctico en la extracción de información de archivos de textos y en la generación de informes a partir de la misma. Es un lenguaje interpretado al igual que muchos otros, por lo que el código no es compilado, sino que se lee cada vez que se quiera ejecutar poniendo en marcha lo que hay escrito.

Perl cuenta con un mecanismo de búsqueda de patrones y expresiones regulares sumamente potente, esto le permite ser eficaz en las tareas de administración de sistema. Existe gran cantidad de código en Internet en forma de módulos que nos permite reutilizarlos para resolver nuestros problemas con mayor facilidad.

Phyton

Es un lenguaje orientado a objetos que posee un alto nivel y nos permite realizar nuestros scripts en poco tiempo. Su sintaxis es muy clara a la hora del tratamiento de excepciones, tipos de datos y la inclusión de módulos. Posee una interfaz orientada a eventos con XML muy fácil y cómoda de programar. Al igual que los demás lenguajes de programación estudiados puede ser usado libremente y en varias plataformas. En definitiva, un lenguaje que, aunque poco conocido, cada día esta ganando usuarios debido, principalmente, a su potencia y a su sencillez de trabajo.

Hypertext Preprocessor (PHP)

Es un lenguaje de programación interpretado que se ejecuta en el servidor, además nos permite dar las características dinámicas a nuestra Web. Se puede embeber muy fácil dentro del HTML. Permite crear aplicaciones Web muy robustas con un código fácil y legible. Es totalmente gratis y su uso puede ser en varias plataformas. (VAZQUEZ 2003)

Mediante PHP también es posible crear aplicaciones con Interfaz Gráfica de Usuario (GUI), utilizando las extensiones PHP-GTK. Nos permite un potente trabajo con la generación de archivos como PDF, Flash, así como el trabajo con diferentes formatos de imágenes. Soporta diferentes gestores de bases de datos como son: MySQL, PostgreSQL, Oracle, SQL Server, entre otras. PHP ha transcurrido por varias generaciones, desde la 3.x hasta la 5.x, que es en esta última donde se incluyen las mayores ventajas y mejoras del mismo.

- Soporte sólido para Programación Orientada a Objetos.
- Mejoras de rendimiento.
- Mejor soporte para MySQL con extensión de rewrite completa.
- Mejor soporte a XML (XPath, DOM, entre otros).
- Soporte nativo para SQLite.

- Soporte integrado para SOAP.
- Iteradores de datos.
- Excepciones de errores.

PHP cuenta con una potente librería de acceso a datos como es ADOdb que nos permite implementar una capa de abstracción de acceso a datos que nos permite la conexión a servidores de bases de datos soportados por el lenguaje. También posee las librerías para el trabajo con plantillas, denominadas Smarty; las mismas nos dan la posibilidad de separar el negocio de la presentación o sea el código php del html, haciendo más legible la sintaxis de nuestro sitio.

1.4 Tecnologías del lado del cliente

JavaScript

JavaScript es un lenguaje interpretado que se ejecuta en los navegadores del cliente, posee una sintaxis parecida al C y Java. No podemos decir que es orientado a objetos de un todo ya que carece de herencia, pero si más bien es basado en prototipos ya que las nuevas clases son generadas clonando las clases bases y así extendiendo su funcionalidad.

Es muy usado a la hora de darle un dinamismo a nuestra Web, ya sea dinamismo visual (creación de menú, cambiar formato del documento, etc.) o en la validación de formularios, en este caso suele ser bastante efectivo del lado del cliente ahorrándole trabajo al servidor Web.

Cascade Style Sheets (CSS)

Las hojas de estilo en cascada describen como se va a mostrar un documento, permitiendo darle estilos a los documentos HTML y XML. Mediante CSS es posible separar el contenido de la presentación, lo que posibilita que se pueda simplificar y limpiar el HTML de los documento, haciendo éstos mas accesibles.

Hyper Text Markup Language (HTML)

HTML es el lenguaje que nos permite definir una página Web mediante la estructuración de textos y presentando los mismo en forma de hipertexto. Este lenguaje constituye en sí un conjunto de etiquetas utilizadas para estructurar dicho documento en forma de texto, imágenes, vínculos y otros elementos que componen la página. Unas de sus principales ventajas es su facilidad de uso porque puede ser editado desde cualquier editor, ya sea de textos o WYSIWYG.(BARCIA fecha desconocida)

Extensible Markup Language XML (XML)

XML es una tecnología en realidad muy sencilla que tiene a su alrededor otras tecnologías que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Desarrollado por el World Wide Web Consortium (W3C), tiene como objetivo principal desarrollar una Web más semántica.

XML no solo se usa en Internet, también se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas, pudiéndose usar en bases de datos, editores de texto, hojas de cálculo y mucho más. Está complementado por otras tecnologías que los hacen tener mayores posibilidades. También permite la compatibilidad entre sistemas para el intercambio de información de una forma fiable y segura.

1.5 ¿Qué es el software libre?

A grandes rasgos software libre es aquel que puede ser libremente usado, copiado, estudiado, modificado y redistribuido después de ser obtenido, más allá de los beneficios económicos que puede traer a cualquier país o entidad que lo use, el software libre está considerado un movimiento social liberador por todas las libertades que su uso propone. Cuba se ha hecho eco de este movimiento y está incursionando en el mismo con vistas a abrirse un espacio en el mercado pues por las políticas de guerra fría que EEUU ha implantado y sostenido durante muchos años no podemos utilizar las herramientas privadas que producen compañías norteamericanas ni pagar licencias a las mismas, cabe preguntarse entonces:

1.5.1 ¿Qué le depara a Cuba el camino del software libre?

Sin hacer un análisis profundo, ni pormenorizado, se puede decir que en la actualidad, los productos de software cubanos, son pagados a precios muy inferiores de su valor real ¿por qué? Porque, son productos desarrollados sobre plataforma Windows, productos que no pueden ser patentados como cubanos y mucho menos comercializados sin un respaldo o un tercer país. Sin embargo, si los productos fueran SWL, Cuba podría hacer uso de ellos, según la licencia bajo la que haya comprado las aplicaciones, sin complicaciones jurídicas y además de la remuneración económica, obtendría entonces reconocimiento por la calidad del producto y consiguientemente su propio mercado.

En resumen, el SWL le depara a Cuba ventajas económicas, legales, políticas y sociales y se ajusta más a la ideología que propone su sistema social.

Por esta razón hay un grupo de herramientas y aplicaciones que marcan la diferencia con el software “privativo” y que se pueden mencionar algunas de las que son orientadas a la Web, es el caso de Apache como aplicación servidora insignia del SWL, quanta, bluefish, gimp y otro grupo de herramientas que no deben ser pasadas por alto a la hora de elaborar la solución contenida en este trabajo las cuales poseen la mayor virtud de todas, ser libres y haber sido creadas para hacer libres a las personas.

1.6 Metodologías de desarrollo de software

En un proyecto de desarrollo de software la metodología es un proceso que se utiliza para definir quién debe hacer qué, cuándo y cómo debe hacerlo.

No existe una metodología de software universal. Las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable y por tanto que se utilice la más apropiada para su desarrollo. (MOLPECERES 15/12/2002)

XP (eXtreme Programming)

Es una metodología relativamente joven (alrededor de 5 años). Su filosofía es satisfacer al completo las necesidades del cliente, por eso, lo integra como una parte más del equipo de desarrollo. Está diseñada para el desarrollo de aplicaciones que requieren un grupo de programadores pequeño.

Sus principales características son:

- Comunicación.
- Simplicidad.
- Realimentación (Feedback).
- Coraje.

RUP (Rational Unified Process)

Es uno de los procesos más generales de los existentes actualmente, ya que en realidad esta pensado para adaptarse a cualquier proyecto. Por su dimensión el proceso de desarrollo se divide en ciclos de desarrollo los cuales dan como resultado una versión del producto final, estos ciclos son:

- Inicio: Se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos (puesta en marcha).
- Elaboración: Se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos (definición, análisis, diseño)
- Construcción: Se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario (implementación).
- Transición: Se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados (fin del proyecto y puesta en producción).

En cada una de estas fases del proyecto se realizan las siguientes actividades:

- Modelado del negocio
- Análisis de requisitos
- Análisis y diseño
- Implementación
- Prueba
- Distribución
- Gestión de configuración y cambios
- Gestión del proyecto
- Gestión del entorno

Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de caso de uso, el código fuente, etc.) y define una serie de roles que se distribuyen entre los miembros del proyecto y las tareas de cada uno a lo largo del proceso. ("RATIONAL_UNIFIED_PROCESS" 2003)

FDD (Feature Driven Development)

Esta pensado para proyectos con tiempo de desarrollo relativamente cortos (menos de un año). Se basa en un proceso iterativo con iteraciones cortas que producen un software funcional que el cliente y la dirección de la empresa pueden ver y monitorizar.

La metodología FDD esta dividida en fases:

- Desarrollo de un modelo general.
- Construcción de la lista de funcionalidades.
- Plan de lanzamientos en base a las funcionalidades a implementar.
- Diseñar en base a las funcionalidades.
- Implementar en base a las funcionalidades.

El trabajo se realiza en grupo, aunque siempre habrá un responsable último que puede ser el arquitecto jefe o jefe de programadores en dependencia de la fase en que se encuentre, con mayor experiencia, que tendrá la última palabra en caso de no llegar a un acuerdo. Al hacerlo en grupo se consigue que todos formen parte del proyecto y que los menos inexpertos aprendan de las discusiones de los mas experimentados, y al tener un responsable último, se asignan las responsabilidades que todas las empresas exigen.

1.7 Propuesta

Luego del análisis llevado a cabo, se plantea la propuesta de desarrollar una aplicación Web, que siga el modelo de un sistema de información confidencial y que incluya inicialmente, además de otras funcionalidades, un generador de reportes automáticos, utilizando como lenguaje de programación del lado del servidor el PHP dada su portabilidad y eficiencia, apoyándose en el uso del gestor de plantillas SMARTY dada la facilidad de trabajo que brinda, y el JavaScript para lograr una interactividad con el usuario en el navegador, además CSS para lograr el formato requerido documentos HTML en la interfaz de usuario . Se propone también la utilización de Mysql como SGBD, la librería ADOdb para el acceso a los datos y Apache como servidor Web. Como metodología de desarrollo se utilizará RUP por su capacidad de adaptarse a cualquier proyecto de software.

1.8 Conclusiones

En este capítulo se hizo un análisis de las tecnologías que se utilizan hoy en el desarrollo para escoger las tecnologías de la propuesta de solución, así como algunos conceptos y tendencias que esta debe adoptar para sí, como es el caso del comercio electrónico. Se fundamentó la elección del lenguaje de programación, el sistema gestor de bases de datos, el servidor Web, y la metodología de desarrollo de software; así como el uso de otras tecnologías. Finalmente se planteó la propuesta que incluye dichos aspectos.

Capítulo 2: Características del Sistema.

2.1 Introducción

El presente capítulo da paso a la descripción de la solución propuesta para el sistema. Se plantea el objeto de automatización describiendo los procesos que serán automatizados, los objetivos estratégicos que se persiguen por parte de la institución, además se especifica la información que se maneja, se hace un análisis crítico de la forma en que se ejecutan los procesos actuales. Se determinan actores, trabajadores y casos de uso del negocio planteando el modelo del negocio y por último los requisitos funcionales, no funcionales con los que debe cumplir el sistema y el modelo de casos de uso del sistema.

2.2 Fundamentación de la situación problemática

El MINCULT como entidad interesada en la ejecución de este trabajo persigue como **objetivos estratégicos**, centralizar la información referente a talento artístico, actividades programadas por parte de los municipios en las diferentes manifestaciones, las instalaciones culturales del país y el público que se interesa en participar, facilitar los procesos de planificación, información y control por medio de la implementación del sistema, monitorear la información referente al departamento de programación, los municipios y las instituciones relacionadas, hacer análisis cuantitativos y cualitativos basándose en la información que llega desde los municipios. Se identificaron para cumplir con las exigencias de la institución como procesos de negocio, la planificación de actividades culturales por parte del departamento de programación, la planificación de giras nacionales de diferentes agrupaciones de la danza y la música, los procesos de conformar la programación municipal, solicitar agrupaciones musicales e informar desde la base hasta el MINCULT los diferentes reportes de programación.

Durante el análisis de los procesos de negocio, en reuniones con el cliente, se pudo observar como se ejecutan, permitiendo identificar algunas dificultades; por ejemplo en los procesos de planificación se distribuyen el talento directamente a los municipios, sin embargo este documento llega a la provincia y no directamente al programador municipal, esto trae consigo que algunas veces las actividades no se conozcan en tiempo por problemas de comunicación entre los diferentes niveles. En el proceso de información desde la base hacia el MINCULT también se vieron irregularidades que entorpecían el buen funcionamiento del negocio pues actualmente la información viaja por dos canales diferentes, uno partiendo de las instituciones municipales, hacia las instituciones nacionales pasando por las provinciales

y finalmente informándose al MINCULT, el otro desde la dirección municipal de cultura, hacia el MINCULT, pasando a través de la dirección provincial, en ambos caso la información debería ser la misma pero producto de las dos diferentes vías por las que viaja al llegar al MINCULT no tienen los mismos datos lo que provoca imprecisiones cuando se hacen análisis sobre una determinada manifestación o territorio.

Los **documentos que se manejan** en el campo de acción comprendido son: modelo de programación, en el cual se envía la información desde la base hasta el MINCULT; distribución de las giras nacionales, contiene la información de las giras nacionales referentes a cada municipio; distribución fiestas populares, el mismo contiene la información de la programación de las fiestas populares de cada municipio.

2.3 Modelo del negocio

EL flujo de trabajo de modelación del negocio sirve para comprender los procesos de la organización y describe cómo desarrollar la nueva organización que se pretende alcanzar, en la empresa que se este analizando, y sobre esta base, definir los procesos, roles y responsabilidades de esa organización en el modelo de casos de uso del negocio y el modelo de objetos del negocio. 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.

2.3.1 Modelo de casos de uso

El modelo de Casos de Uso del Negocio es un modelo que describe los procesos de un negocio y su interacción con elementos externos (o sea los actores), tales como socios y clientes, es decir, describe las funciones que el negocio pretende realizar. Identificar los procesos en el negocio, definir las fronteras del negocio que van a modelarse, definir quién y qué interactúa con el negocio y crear diagramas del modelo de casos de uso del negocio son los **objetivos principales** de este modelo.

2.3.2 Actores del negocio

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados.

Actor	Descripción
Promotor cultural	Detona el CU programación cultural y utiliza los servicios de la programación pues es quien lleva a la cultura a la comunidad.
CNAE	Representa a la persona que dirige el departamento de programación del Centro Nacional de Artes Escénicas y es el que inicia el caso de uso programar giras artes escénicas.
Dirección municipal de cultura	Inicia el caso de uso Solicitud de agrupación, representa los intereses del pueblo que es el que se beneficia con la presentación de la agrupación que desean.
Artista Plástico	Es el interesado en utilizar los servicios de la galería y por tanto el que inicia el caso de uso programación de exposiciones de la plástica y el beneficiado con los servicios que presta la galería.
ICAIC	Representa a la persona que dirige el departamento del ICAIC referente a programación y es quien indica que se cree la programación cinematográfica.

ICM	Es quien inicia los casos de uso programación de fiestas populares y programación de giras de la música.
-----	--

2.3.3 Trabajadores del negocio

Son los que trabajan dentro de las fronteras del negocio y están en condiciones en convertirse en actores del sistema a desarrollar. (“RATIONAL_UNIFIED_PROCESS” 2003)

Trabajador	Descripción
Programador	Persona encargada plasmar en el modelo de programación una vez terminado el taller de programación.
Instituciones	Son las encargadas de las estadísticas en su área y nivel específico.
MINCULT	Responsable de las estadísticas nacionales de las instituciones y encargado de generar los reportes correspondientes para el departamento de estadísticas.
Dirección provincial de cultura	Encargada de las estadísticas de todas las instituciones a nivel municipal y provincial.
Programador ICAIC	Encargado de crear la programación de las películas que se exponen en los diferentes cines del país.
Programador ICM	Es el encargado de programar las giras de las agrupaciones musicales, las presentaciones de los grupos musicales en las fiestas populares de cada

	municipio y atender las solicitudes de agrupaciones musicales, verificando si están o no disponibles.
Programador CNAE	Encargado distribuir las presentaciones de las agrupaciones danzareas de primer nivel por municipios, creando un documento que envía a los mismos.
Encargado de galería	Es el que atiende a los artistas plásticos en el proceso de conveniar una exposición en la galería en que trabaja, además se encarga de informarle al programador municipal sobre el evento.
IPAIC	Recibe la programación cinematográfica proveniente del ICAIC y le distribuye a los cines provinciales y municipales.

2.3.4 Casos de uso extendidos

Un caso de uso representa un proceso completo identificado en el negocio, se pueden extender o incluir entre ellos en dependencia de la complejidad de los mismos. Su descripción extendida representa cada paso del proceso que tiene lugar dentro de su frontera. ("RATIONAL_UNIFIED_PROCESS" 2003)

2.3.4.1 CU Crear programación

Caso de Uso:	Crear programación
Actores:	Promotor cultural (inicia)
Trabajadores:	
Resumen:	El caso de uso se inicia cuando el promotor cultural se presenta ante el programador cultural con las propuestas de programación para el mes

	indicado.
Precondiciones:	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. Llega con sus propuestas al programador	<p>1.1 El programador cultural conforma la programación, escribiéndola en el modelo de programación.</p> <p>1.2 El programador envía la programación a la dirección provincial de cultura.</p> <p>1.3 La dirección provincial de cultura recibe la programación.</p> <p>1.4 La dirección provincial de cultura le da a conocer a las instituciones provinciales sobre la programación del municipio en cada una de las áreas (música, artes plásticas, artes escénicas, literatura, etc.).</p> <p>1.5 Las instituciones provinciales ven la programación y se la envían a las instituciones nacionales.</p> <p>1.6 Las instituciones naciones reciben la programación y se la dan a conocer al MINCULT.</p>
Flujos Alternos	

Acción del Actor		Respuesta del Negocio
Poscondiciones		
Mejoras	Este caso de uso se automatizará completamente.	
Prioridad	Crítico.	

2.3.4.2 CU Solicitar agrupación

Caso de Uso:	Solicitar agrupación.	
Actores:	Dirección municipal de cultura (inicia)	
Trabajadores:		
Resumen:	El caso de uso se inicia cuando la dirección municipal de cultura hace la solicitud de una agrupación ya sea comercial o subvencionada para que su pueblo disfrute de la misma.	
Precondiciones:		
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Negocio
1. Envía la solicitud.		1.1 La dirección provincial de cultura recibe la solicitud. 1.2 La dirección provincial de cultura asigna la

	agrupación al municipio.
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
Acción 1.1	La agrupación solicitada por el municipio no es provincial, en este caso la provincia tiene que pedirla al ICM, quien tendría que valorar si puede o no asignar la agrupación con las agencias musicales, en caso de que la empresa a la que pertenece el grupo este de acuerdo, se le asigna la agrupación al municipio que la solicito, si no se le oferta otra.
Acción 1.2	Puede que la agrupación solicitada no este disponible y se le oferte otra, en este caso el municipio puede o no recibirla.
Poscondiciones	
Mejoras	
Prioridad	Secundario.

2.3.4.3 CU Programar fiestas populares

Caso de Uso:	Programar fiestas populares.
Actores:	ICM
Trabajadores:	

Resumen:	El caso de uso se inicia cuando el ICM representando a la persona o departamento encargado pide que se programe la distribución de agrupaciones para las fiestas populares, la cual una vez creada es enviada al municipio, pasando por la provincia.	
Precondiciones:		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. Pide se cree la distribución.	<p>1.1 El programador de la música hace la distribución.</p> <p>1.2 El programador informa la distribución a la dirección provincial de cultura.</p> <p>1.3 La dirección provincial de cultura recibe la distribución.</p> <p>1.4 La dirección provincial de cultura envía la distribución al municipio.</p> <p>1.5 El programador municipal recibe la distribución.</p>	
Flujos Alternos		
Acción del Actor	Respuesta del Negocio	
Poscondiciones		

Mejoras	
Prioridad	Primario

2.3.4.4 CU Programar Exposiciones.

El contenido de este caso de uso es muy similar al del anterior (Programar fiestas populares) por tanto se omite.

2.3.4.5 CU Programar giras nacionales (Música)

El contenido de este caso de uso es muy similar al del anterior (Programar fiestas populares) por tanto se omite.

2.3.4.6 CU Programar giras nacionales (Artes escénicas)

El contenido de este caso de uso es muy similar al del anterior (Programar fiestas populares) por tanto se omite.

2.3.4.7 CU Crear programación cinematográfica

El contenido de este caso de uso es muy similar al del anterior (Programar fiestas populares) por tanto se omite.

2.4 Requisitos

2.4.1 Requisitos funcionales

Los Requerimientos funcionales especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física. (SCHMULLER 2000)

R1 Autenticar usuarios

R1.1 Verificar que un usuario exista en el sistema y que su contraseña es correcta.

R1.2 Guardar el nivel y el rol del usuario autenticado.

R2 Gestionar unidades artísticas

R2.1 Listar unidades artísticas existentes en un municipio.

R2.2 Adicionar una nueva unidad artística.

R2.3 Modificar los datos de una unidad artística existente.

R2.4 Eliminar una unidad artística.

R2.5 Inhabilitar una unidad artística por un tiempo determinado.

R3 Gestionar espacios culturales

R3.1 Listar espacios culturales existentes en un municipio.

R3.2 Adicionar un nuevo espacio cultural.

R3.3 Modificar los datos de un espacio cultural existente.

R3.4 Eliminar un espacio cultural.

R3.5 Inhabilitar temporalmente un espacio cultural.

R4 Gestionar actividades culturales

R4.1 Listar actividades culturales planificadas en el municipio.

R4.2 Planificar actividades culturales a partir del talento local y de las asignaciones de instituciones superiores.

R4.3 Modificar una actividad cultural existente.

R4.4 Eliminar una actividad cultural existente.

R4.5 Completar los datos (asistencia, cobertura de televisión, cobertura de prensa, cobertura de radio, recaudación en moneda nacional, recaudación en divisa) de una actividad cultural posteriormente a su realización.

R4.6 Especificar las causas de la actividad cultural que no se realizó.

R5 Planificar fiestas populares

R5.1 Crear la distribución de fiestas populares para cada municipio.

R6 Planificar giras de agrupaciones musicales

R6.1 Crear la distribución de la participación de las agrupaciones musicales en cada municipio.

R7 Planificar proyecciones cinematográficas

R7.1 Asignar a cada provincia la distribución de películas correspondiente.

R8 Planificar giras de artes escénicas.

R8.1 Crear la distribución de las agrupaciones de artes escénicas.

R9 Mostrar vistas

R9.1 Mostrar a las instituciones una vista de su área en los diferentes niveles.

R10 Mostrar estadísticas

R10.1 Crear reportes personalizados en dependencia de las necesidades del cliente.

R11 Administrar usuarios.

R11.1 Listar los usuarios existentes.

R11.2 Adicionar un nuevo usuario.

R11.3 Editar información de un usuario dado.

R11.4 Eliminar un usuario dado.

R12 Administrar roles.

R12.1 Listar los roles existentes en el sistema.

R12.2 Adicionar un nuevo rol.

R12.3 Editar la información de un rol dado.

R12.4 Eliminar un rol dado.

R13 Administrar permisos

R13.1 Listar los roles existentes y los módulos a los que tiene acceso.

R13.2 Editar los permisos.

2.4.2 Requisitos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener.

Apariencia o interfaz externa

- Diseño sencillo, legible con colores adecuados y agradables.
- Diseño creado específicamente para resolución de 800x600, pero que se vea en otras resoluciones con la misma claridad.

Rendimiento

- Garantizar la menor cantidad de imágenes posibles en los documentos.

Soporte

- Garantía de instalación y pruebas al sistema.
- Preparación de los usuarios finales del sistema.

Portabilidad

- Debe ser multiplataforma.

Seguridad

- Control de acceso a través de diferentes roles con permisos a determinados módulos.
- Envío de las contraseñas de los usuarios al servidor de forma encriptada.
- Confirmar todas las acciones que no se puedan deshacer (Ejemplo: Eliminar un elemento).

Confiabilidad

- Recibir datos introducidos por los usuarios correctamente validados.
- Garantizar un tratamiento de excepciones cuyos mensajes no comprometan ninguna información del servidor y sean lo más concreta posible.

2.5 Definición de los actores y casos de uso del sistema

2.5.1 Listado de actores del sistema

Actor	Descripción
Programador	Interactúa con el sistema para llenar los formularios de programación y enviar los datos a los niveles superiores.
Administrador del sistema	Interactúa con el sistema para la gestión de los roles y usuarios del sistema, así como el nivel de acceso de cada rol.
MINCULT	Es el que inicia el caso de uso mostrar estadísticas y clasificadores, es el beneficiado en este proceso pues es quien se interesa por las estadísticas referentes a programación cultural.
Programador ICAIC	Interactúa con el sistema en el momento de especificar la planificación cinematográfica para los diferentes cines de la capital y las demás provincias.
Programador ICM	Interactúa con el sistema cuando especifica la planificación de agrupaciones musicales para las fiestas populares de los municipios y las giras nacionales de las mismas.
Programador CNAE	Especifica la información referente a las giras nacionales de las agrupaciones de artes escénicas, interactúa con el sistema en el caso de uso gestionar planificaciones.
Usuario	Es una generalización de todos los usuarios que se autentican en el sistema.

Instituciones	Inicia el caso de uso mostrar vistas mediante el cual puede conocer el comportamiento de las actividades programadas en los municipios.
---------------	---

2.5.2 Listado de casos de uso del sistema

Los casos de uso del sistema tienen como objetivo satisfacer los requisitos funcionales, en un caso de uso se pueden satisfacer varios requisitos en dependencia de su complejidad.

CU-1	Autenticar usuarios
Actor	Usuario
Descripción	Este caso de uso tiene lugar cada vez que un usuario entra al sistema pues tiene que autenticarse, el actor se autentica ante el sistema para identificarse y recibir de este los derechos correspondientes a su nivel.
Referencia	R1, R1.1, R1.2

CU-2	Gestionar unidades artísticas.
Actor	Programador.
Descripción	El caso de uso se inicia cuando el programador municipal accede al sistema para hacer alguna operación con las unidades artísticas que tiene a su disposición. Mediante este caso de uso se puede agregar, eliminar, editar un recurso artístico en el sistema.

Referencia	R2, R2.1, R2.2, R2.3, R2.4, R2.5
------------	----------------------------------

CU-3	Gestionar espacios culturales.
Actor	Programador.
Descripción	Este caso de uso se origina en la necesidad que tiene el MINCULT de llevar el control de los espacios culturales, se inicia cuando el programador municipal accede dentro del módulo programación a la opción de espacios culturales y comienza a ejecutar las acciones necesarias (listar, adicionar, editar y eliminar) para la gestión de los mismos.
Referencia	R3, R3.1, R3.2, R3.3, R3.4, R3.5

CU-4	Gestionar actividades culturales.
Actor	Programador.
Descripción	Permite al programador listar las actividades culturales, planificarlas y especificar las causas por las que una actividad cultural planificada no se ejecute correctamente.
Referencia	R4, R4.1, R4.2, R4.3, R4.4

CU-5	Planificar fiestas populares.
Actor	Programador ICM.

Descripción	El actor se beneficia de este caso de uso, pues mediante él puede planificar las presentaciones de las agrupaciones musicales en las fiestas populares de cada municipio e informar de esta programación a los municipios de forma simple.
Referencia	R5, R5.1

CU-6	Programar giras musicales.
Actor	Programador ICM.
Descripción	El actor se beneficia de este caso de uso, pues mediante él puede planificar las presentaciones de las agrupaciones musicales del ICM y hacer llegar esta programación a los municipios de forma simple.
Referencia	R6, R6.1

CU-7	Planificar proyección cinematográfica.
Actor	Programador ICAIC.
Descripción	El caso de uso permite al programador del ICAIC hacer llegar a las provincias y municipios de forma simple la programación de la exhibición de películas en los respectivos cines.
Referencia	R7, R7.1

CU-8	Planificar giras CNAE.
Actor	Programador CNAE.
Descripción	El actor mediante este caso de uso, puede planificar las presentaciones de las agrupaciones de danza e informar de esta programación a los municipios de forma simple.
Referencia	R8, R8.1

CU-9	Mostrar vistas.
Actor	Instituciones.
Descripción	El actor instituciones inicia el caso de uso y mediante él puede conocer según su nivel, las estadísticas de su área específica.
Referencia	R9, R9.1

CU-10	Mostrar estadísticas.
Actor	MINCULT.
Descripción	El MINCULT como actor interesado en conocer las estadísticas de los municipios inicia el caso de uso, en el cual puede conocer datos de la programación de cada municipio o manifestación que le permiten hacer análisis

	cualitativos y cuantitativos.
Referencia	R10, R10.1

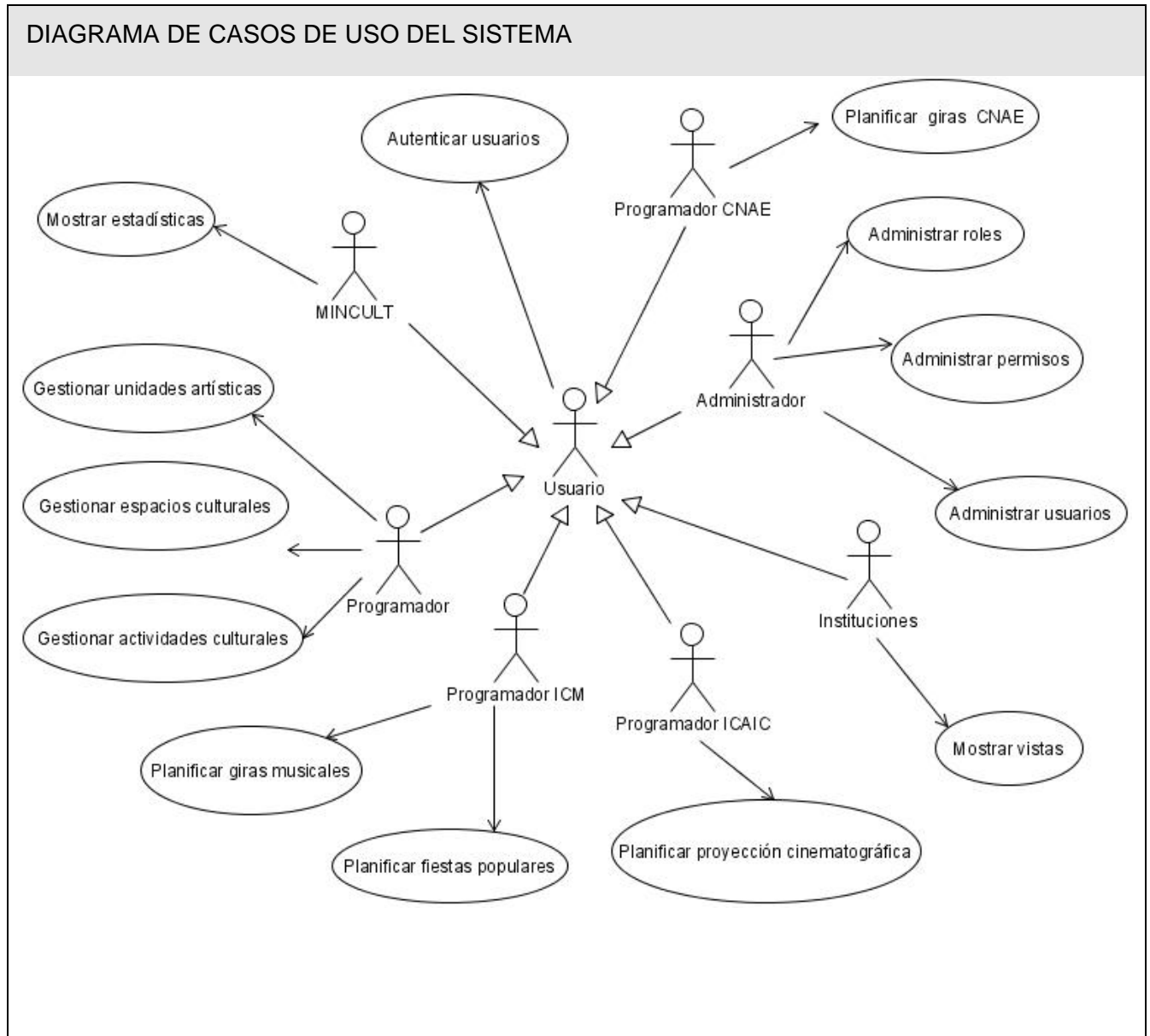
CU-11	Administrar usuarios
Actor	Administrador.
Descripción	El administrador por medio de este caso de uso puede adicionar, eliminar y editar un usuario al sistema.
Referencia	R11, R11.1, R11.2, R11.3, R11.4

CU-12	Administrar roles.
Actor	Administrador.
Descripción	Por medio de este caso de uso el administrador del sistema puede adicionar, eliminar y editar los diferentes roles que pueden jugar los usuarios, también puede dar o quitar permisos a cada rol.
Referencia	R12, R12.1, R12.2, R12.3, R12.4

CU-13	Administrar permisos.
Actor	Administrador.
Descripción	El administrador puede administrar listar los roles de

	existentes en el sistema y cambiar los permisos de usuarios en dependencia de los niveles que estos tengan.
Referencia	R13, R13.1, R13.2

2.6 Diagrama de casos de uso del sistema



2.7 Casos de uso expandido

2.7.1 CU Autenticar usuarios

Caso de uso

CU-1	Autenticar usuarios	
Propósito	Permitir a los usuarios identificarse para acceder al sistema	
Actores	Usuario	
Resumen: El caso de uso se inicia cuando el usuario solicita el acceso al sistema. Permite obtener la información de los usuarios que acceden al sistema.		
Referencias	R1, R1.1, R1.2	
Acción del actor		Respuesta del sistema
1. Solicita el acceso al sistema		1.1 Muestra un formulario para que el usuario teclee sus credenciales (usuario y contraseña).
2. Introduce las credenciales (usuario y contraseña).		2.1 Verifica que las credenciales (usuario y contraseña) entradas existan en la base de datos. 2.2 Verifica si el usuario está activo. 2.3 Obtiene de la base de datos el nivel y el rol del usuario y lo guarda en la sesión. 2.4 Redirecciona a la página principal del sistema.
Flujo alternativo		
Acción del actor		Respuesta del sistema
Acción 2.1		En caso que las credenciales no existan en la base datos, muestra un mensaje de error.
Acción 2.2		Si el usuario no está activo muestra un mensaje de error.

Acción 2.4	En caso que el usuario haya tratado de entrar directamente una página del sistema, redirecciona hacia esa página.
------------	---

2.7.2 CU Gestionar unidades artísticas

Caso de uso	
CU-2	Gestionar unidades artísticas
Propósito	Permitir listar, adicionar, eliminar y modificar los datos de las unidades artísticas existentes, así como inhabilitar una unidad artística por un tiempo determinado.
Actores	Programador
Resumen: El caso de uso se inicia cuando el programador accede al sistema y selecciona la opción de gestionar unidades artísticas	
Referencias	R2, R2.1, R2.2, R2.3, R2.4, R2.5
Acción del actor	Respuesta del sistema
1 Selecciona la opción gestionar unidad artística	<p>1.1 El sistema muestra por defecto la interfaz con el listado de unidades artísticas existentes (ir a la sección “Listar unidades artísticas”), que pertenezcan al municipio del programador.</p> <p>2.1 Ejecuta de las siguientes acciones, la que el programador escoja.</p> <p>a) Si decide adicionar una unidad artística, ir a la sección “Adicionar unidad artística”.</p> <p>b) Si decide eliminar una unidad artística, ir a la</p>

	<p>sección “Eliminar unidad artística”.</p> <p>c) Si decide modificar una unidad artística, ir a la sección “Modificar unidad artística”.</p> <p>d) Si decide inhabilitar una unidad artística, ir a la sección “Inhabilitar unidad artística”.</p>
Sección “Listar unidades artísticas”	
Acción del actor	Respuesta del sistema
1 Selecciona la opción listar unidades artísticas	<p>1.1 Muestra un listado con todas las unidades artísticas existentes en el sistema.</p> <p>1.2 Muestra un formulario de búsqueda para filtrar el listado de unidades artísticas.</p>
2 Introduce el criterio de búsqueda.	2.1 Muestra un listado de las unidades artísticas que coincidan con el criterio de búsqueda introducido.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Acción 2.1	En caso que no existan unidades artísticas que coincidan con el criterio de búsqueda, muestra un mensaje de error.
Sección “Adicionar unidad artística”	
Acción del actor	Respuesta del sistema
1 Selecciona la opción de adicionar unidad	1.1 Muestra un formulario para introducir los

artística.	datos de la unidad artística que se desea adicionar.
2 Introduce los datos en el formulario.	<p>2.1 Verifica que los datos introducidos sean correctos.</p> <p>2.2 Verifica si existe esa unidad artística en la base de datos.</p> <p>2.3 Inserta los datos de la unidad artística en la base de datos.</p> <p>2.4 Muestra un mensaje de confirmación</p>
Flujo alternativo	
Acción del actor	Respuesta del sistema
Acción 2.1	En caso que los datos introducidos no sean correctos, muestra un mensaje de error.
Acción 2.2	Si la unidad artística ya existe en la base de datos, muestra en mensaje de error.
Sección "Eliminar unidad artística"	
Acción del actor	Respuesta del sistema
1 Selecciona la unidad artística que desea eliminar.	1.1 Muestra una interfaz pidiéndole al programador que confirme si desea eliminar la unidad artística.
2 Confirma que desea eliminar la unidad artística.	2.1 Elimina de la base de datos la información referente a la unidad artística seleccionada.

	2.2 Muestra un mensaje de confirmación.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Acción 2.1	En caso que el programador no confirme eliminar la unidad artística, redirecciona hacia el listado de unidades artísticas
Sección "Modificar unidad artística"	
Acción del actor	Respuesta del sistema
1 Selecciona la unidad artística que desea modificar.	1.1 Muestra un formulario con los datos de la unidad artística que seleccionó el programador.
2 Cambia los datos que desea modificar	2.1 Verifica que los datos introducidos sean correctos. 2.2 Modifica los datos de la unidad artística en la base de datos. 2.3 Muestra un mensaje de confirmación.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Acción 2.1	En caso que los datos introducidos por el programador no sean correctos muestra un mensaje de error.
Sección "Inhabilitar unidad artística"	

Acción del actor	Respuesta del sistema
1 Selecciona la unidad artística que desea inhabilitar.	1.1 Muestra un formulario para que el programador introduzca el tiempo que debe permanecer la unidad artística inhabilitada.
2 Introduce el tiempo que debe permanecer la unidad artística inhabilitada.	2.1 Verifica que el tiempo introducido sea correcto. 2.2 Modifica en la base de datos la información de la unidad artística, introduciendo el tiempo que va a esta inhabilitada. 2.3 Muestra un mensaje de confirmación.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Acción 2.1	En caso que el tiempo introducido no sea correcto, muestra un mensaje de error.

2.7.3 CU Gestionar espacios culturales

Omitido por ser similar al caso de uso, Gestionar unidades artísticas.

2.7.4 CU Gestionar actividades culturales

Omitido por ser similar al caso de uso, Gestionar unidades artísticas.

2.7.5 CU Planificar fiestas populares

Caso de uso	
CU-5	Planificar fiestas populares.

Propósito	Permitir al ICM planificar la distribución de las fiestas populares.	
Actores	Programador ICM.	
Resumen: El caso de uso se inicia cuando el Programador ICM accede al sistema y solicita la opción de planificar fiestas populares.		
Referencias	R5, R5.1	
Acción del actor		Respuesta del sistema
1 Selecciona la opción de planificar fiestas populares.		<p>1.1 Muestra por defecto el listado de las planificaciones de fiestas populares que se han realizado, ver sección “Listar planificaciones”.</p> <p>1.2 Ejecuta de las siguientes acciones la que el Programador ICM escoja:</p> <p>a) Si decide crear una planificación, ir a la sección “Crear planificación”.</p> <p>b) Si decide eliminar una planificación, ir a la sección “Eliminar planificación”.</p>
Sección “Listar planificaciones”		
Acción del actor		Respuesta del sistema
1 Selecciona la opción listar planificaciones.		1.1 Muestra un listado con todas las planificaciones realizadas.
Sección “Adicionar planificación”		
Acción del actor		Respuesta del sistema

1 Selecciona la opción adicionar planificación.	1.1 Muestra un formulario para introducir los datos de la planificación.
2 Introduce los datos de la planificación.	2.1 Verifica que los datos introducidos sean correctos. 2.2 Inserta en la base de datos los datos de la planificación. 2.3 Muestra un mensaje de confirmación
Flujo alternativo	
Acción del actor	Respuesta del sistema
Acción 2.1	En caso que los datos introducidos no sean correctos, muestra un mensaje de error.
Sección "Eliminar planificación"	
Acción del actor	Respuesta del sistema
1 Selecciona la planificación que desea eliminar.	1.1 Muestra un mensaje pidiéndole al programador ICM que confirme si desea eliminar la planificación.
2 Confirma que desea eliminar la planificación.	2.1 Elimina de la base de datos la información de la planificación seleccionada. 2.2 Muestra un mensaje de confirmación.
Flujo alternativo	
Acción del actor	Respuesta del sistema

Acción 2.1	En caso que el programador ICM no confirme que desea eliminar la planificación, redirecciona hacia el listado de planificaciones existentes.
------------	--

2.7.6 CU Planificar giras musicales

Omitido por ser similar al caso de uso, Planificar fiestas populares.

2.7.7 CU Planificar proyección cinematográfica

Omitido por ser similar al caso de uso, Planificar fiestas populares.

2.7.8 CU Planificar giras CNAE

Omitido por ser similar al caso de uso, Planificar fiestas populares.

2.7.9 CU Mostrar vistas

Caso de uso					
CU-9	Mostrar vistas				
Propósito	Permitir que las instituciones puedan ver lo que está sucediendo respecto a su manifestación en su nivel.				
Actores	Instituciones.				
Resumen: El caso de uso se inicia cuando las instituciones acceden al sistema y solicitan la opción de mostrar vistas.					
Referencias	R9, R9.1				
<table border="1" style="width: 100%;"> <tr> <td data-bbox="196 1583 836 1667">Acción del actor</td> <td data-bbox="836 1583 1518 1667">Respuesta del sistema</td> </tr> <tr> <td data-bbox="196 1667 836 1795">1 Selecciona la opción de mostrar vistas.</td> <td data-bbox="836 1667 1518 1795">1.1 Muestra un reporte de lo que está sucediendo a nivel nacional respecto a la institución.</td> </tr> </table>		Acción del actor	Respuesta del sistema	1 Selecciona la opción de mostrar vistas.	1.1 Muestra un reporte de lo que está sucediendo a nivel nacional respecto a la institución.
Acción del actor	Respuesta del sistema				
1 Selecciona la opción de mostrar vistas.	1.1 Muestra un reporte de lo que está sucediendo a nivel nacional respecto a la institución.				

	1.2 Muestra un formulario para seleccionar una provincia específica.
2 Selecciona la provincia que le quiere hacer la vista	2.1 Muestra un reporte de lo que está sucediendo respecto a la institución en esa provincia.
Flujo alternativo.	
Acción del actor	Respuesta del sistema
Acción 1.1	Si la institución es provincial, muestra un reporte de lo que está sucediendo a nivel provincial respecto a esa institución.
Acción 1.2	Si la institución es provincial, muestra un formulario para seleccionar un municipio específico.
Acción 2.1	Si la institución es provincial, muestra un reporte de lo que está sucediendo respecto a la institución en ese municipio.

2.7.10 CU Mostrar estadísticas

Caso de uso	
CU-10	Mostrar estadísticas.
Propósito	Permitir al MINCULT que pueda obtener reportes personalizados a partir de la información suministrada por los municipios.
Actores	Instituciones.

Resumen: El caso de uso se inicia cuando el MINCULT accede al sistema y solicita la opción de mostrar estadísticas.	
Referencias	R10, R10.1
Acción del actor	Respuesta del sistema
1 Selecciona la opción de mostrar estadísticas.	1.1 Muestra un listado con los reportes que el MINCULT ha creado. 1.2 Muestra la opción de crear nuevos reportes. (Ir a la sección “Crear reportes”).
2 Selecciona el reporte que desea ver.	2.1 Muestra un listado con los datos que están en el reporte.
Sección “Crear reportes”	
Acción del actor	Respuesta del sistema
1 Selecciona la opción crear reporte.	1.1 Muestra un formulario para que seleccione los datos que el actor desea que tener en su reporte.
2 Selecciona los datos que desea que tenga el reporte.	1.1 Inserta en la base de datos la información del nuevo reporte. 1.2 Muestra un mensaje de confirmación.
Flujo alternativo	
Acción del Actor	Respuesta del sistema
Acción 2.1	En caso que el actor no seleccione ningún dato,

	muestra un mensaje de error.
--	------------------------------

2.7.11 CU Administrar usuarios

Caso de uso	
CU-11	Administrar usuarios
Propósito	Permitir listar, adicionar, eliminar y modificar los datos de los usuarios que tienen acceso al sistema.
Actores	Administrador
Resumen: El caso de uso se inicia cuando el administrador accede al sistema y selecciona la opción de administrar usuarios.	
Referencias	R11, R11.1, R11.2, R11.3, R11.4
Acción del actor	Respuesta del sistema
1 Selecciona la opción administrar usuarios.	<p>1.1 El sistema muestra por defecto la interfaz con el listado de usuarios existentes (ir a la sección "Listar usuarios").</p> <p>2.1 Ejecuta de las siguientes acciones, la que el administrador escoja.</p> <p>a) Si decide adicionar un usuario, ir a la sección "Adicionar usuario".</p> <p>b) Si decide eliminar un usuario, ir a la sección "Eliminar usuario".</p> <p>c) Si decide modificar un usuario, ir a la sección</p>

	"Modificar usuario".
Sección "Listar usuarios"	
Acción del actor	Respuesta del sistema
1 Selecciona la opción listar usuarios.	1.1 Muestra el listado de usuarios existentes en el sistema. 1.2 Muestra un formulario de búsqueda para filtrar los usuarios existentes.
2 Introduce el criterio de búsqueda	2.1 Muestra el listado de usuarios que coincida con el criterio de búsqueda insertado.
Flujo alternativo	
Acción del actor	Respuesta de sistema
Acción 2.1	En caso que no existan usuarios que coincidan con el criterio de búsqueda, muestra un mensaje de error.
Sección "Adicionar usuario"	
Acción del actor	Respuesta del sistema
1 Selecciona la opción de adicionar usuario.	1.1 Muestra un formulario para introducir los datos del usuario que se desea adicionar.
2 Introduce los datos en el formulario.	2.1 Verifica que los datos introducidos sean correctos. 2.2 Verifica si existe ese usuario en la base de datos.

	<p>2.3 Inserta los datos del usuario en la base de datos.</p> <p>2.4 Muestra un mensaje de confirmación.</p>
Flujo alternativo	
Acción del actor	Respuesta de sistema
Acción 2.1	En caso que los datos introducidos no sean correctos, muestra un mensaje de error.
Acción 2.2	Si el usuario ya existe en la base de datos, muestra en mensaje de error.
Sección "Eliminar usuario"	
Acción del actor	Respuesta del sistema
1 Selecciona el usuario que desea eliminar.	1.1 Muestra una interfaz pidiéndole al administrador que confirme si desea eliminar el usuario.
2 Confirma que desea eliminar el usuario.	<p>2.1 Elimina de la base de datos la información referente al usuario seleccionado.</p> <p>2.2 Muestra un mensaje de confirmación.</p>
Flujo alternativo	
Acción del actor	Respuesta de sistema
Acción 2.1	En caso que el administrador no confirme eliminar el usuario, redirecciona hacia el listado de usuarios.

Sección "Modificar usuario"	
Acción del actor	Respuesta del sistema
1 Selecciona el usuario que desea modificar.	1.1 Muestra un formulario con los datos del usuario que seleccionó el administrador.
2 Cambia los datos que desea modificar	2.1 Verifica que los datos introducidos sean correctos. 2.2 Verifica que el usuario que se está modificando sea único. 2.3 Modifica los datos del usuario en la base de datos. 2.4 Muestra un mensaje de confirmación.
Flujo alternativo	
Acción del actor	Respuesta de sistema
Acción 2.1	En caso que los datos introducidos por el administrador no sean correctos, muestra un mensaje de error.
Acción 2.2	Si el usuario no es único, muestra un mensaje de error.

2.7.12 CU Administrar roles

Se omite por ser similar al caso de uso, Administrar usuarios.

2.7.13 CU Administrar permisos

Se omite por ser similar al caso de uso, Administrar usuarios.

2.8 Conclusiones

En este capítulo se hizo una descripción de la propuesta de solución a través de la modelación del negocio, la identificación de los actores, trabajadores y los casos de uso correspondientes; el planteamiento de los requisitos funcionales y no funcionales del sistema a desarrollar. Por último se plantearon los casos de uso del sistema de modo que se puedan satisfacer los requisitos funcionales.

Capítulo 3: Análisis y diseño del sistema

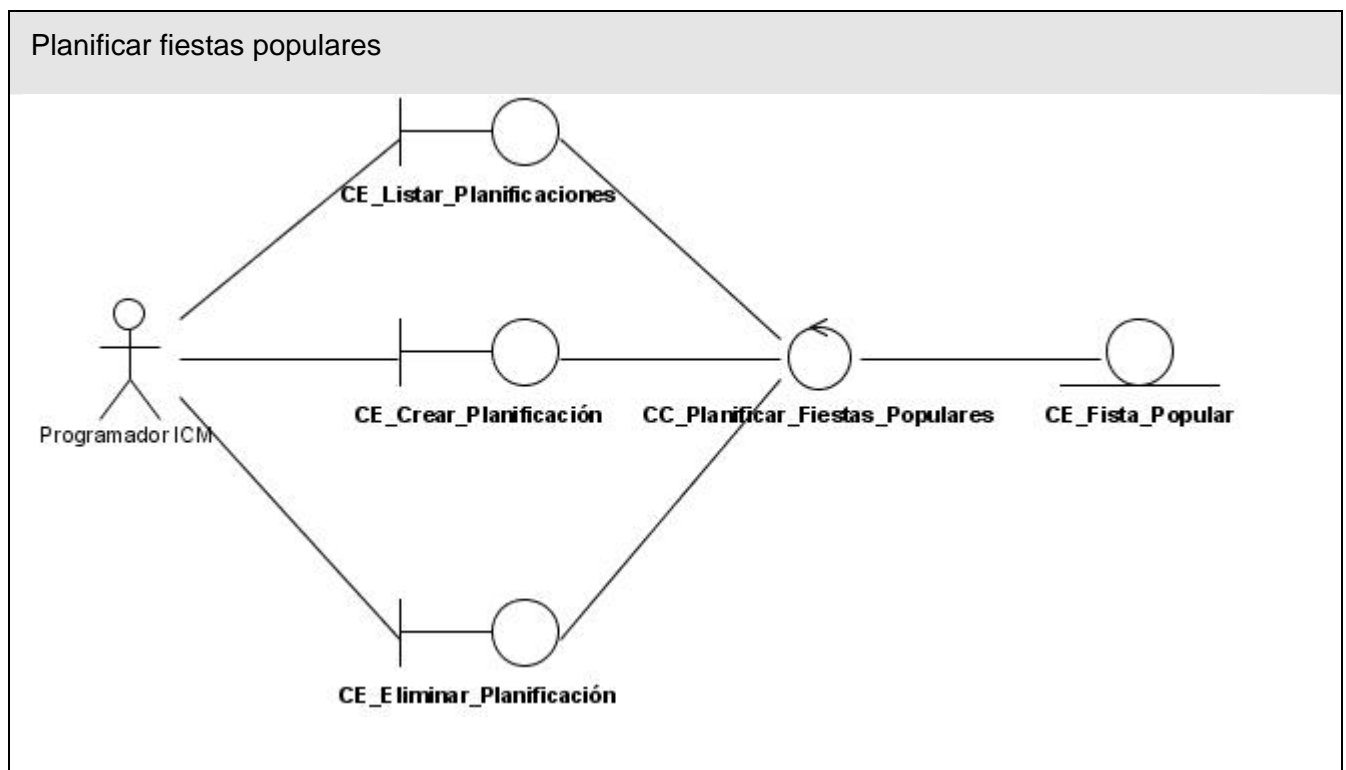
3.1 Introducción

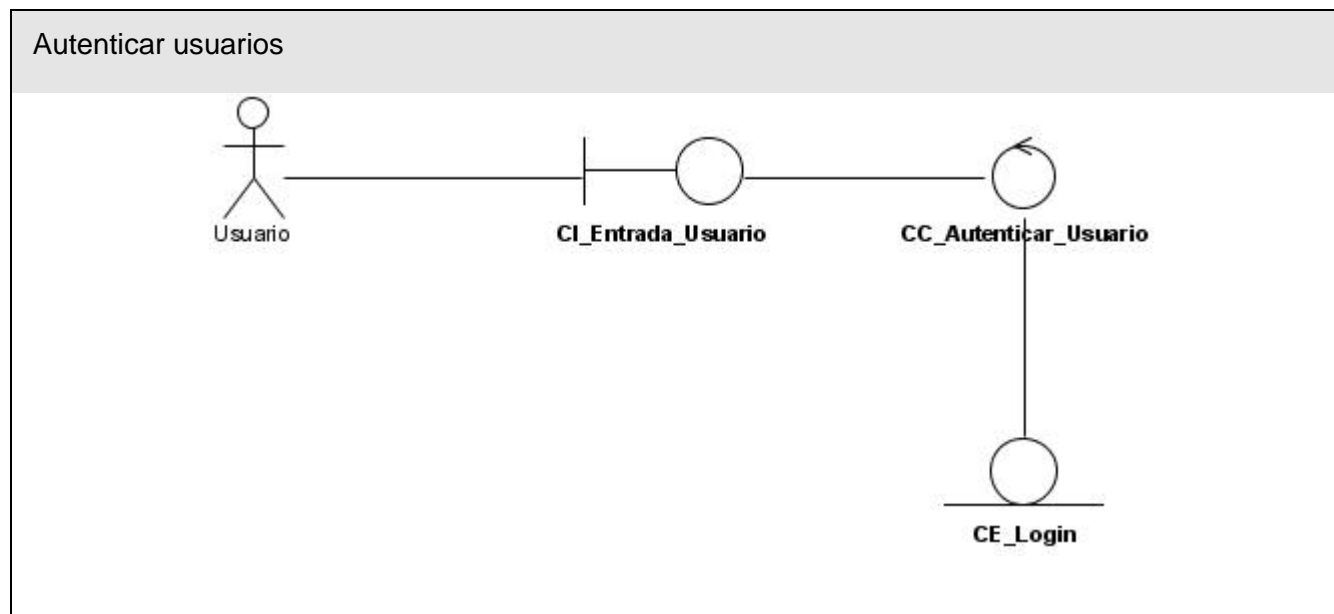
En este capítulo se comienza a construir la solución que ha sido descrita en el anterior, se plantean los modelos de análisis y diseño, el diagrama de clases persistentes y el modelo de datos. Por medio de los diagramas de clases de análisis, diagramas de clases de diseño, los de interacción y el diagrama entidad relación.

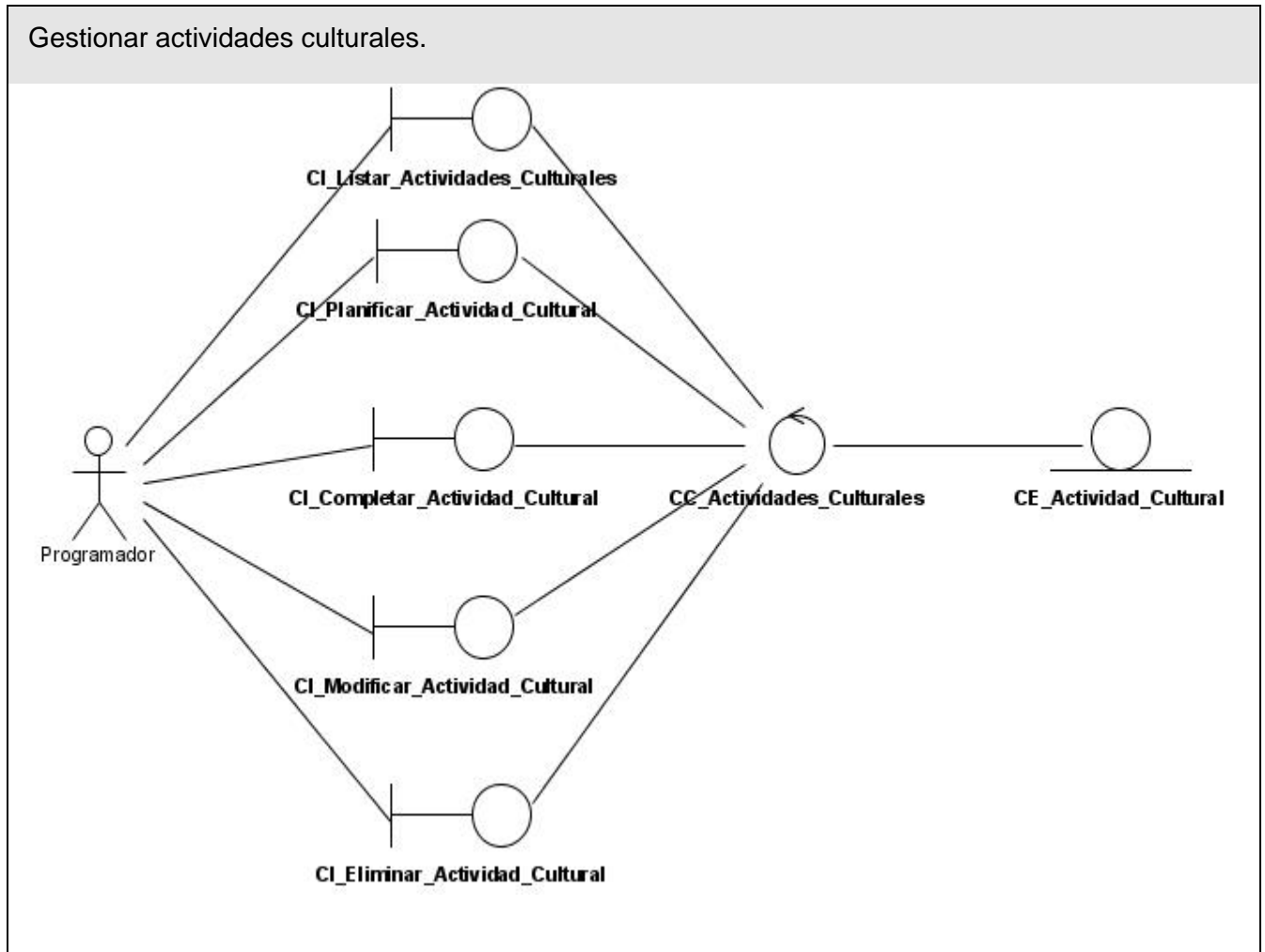
3.2 Modelo de análisis

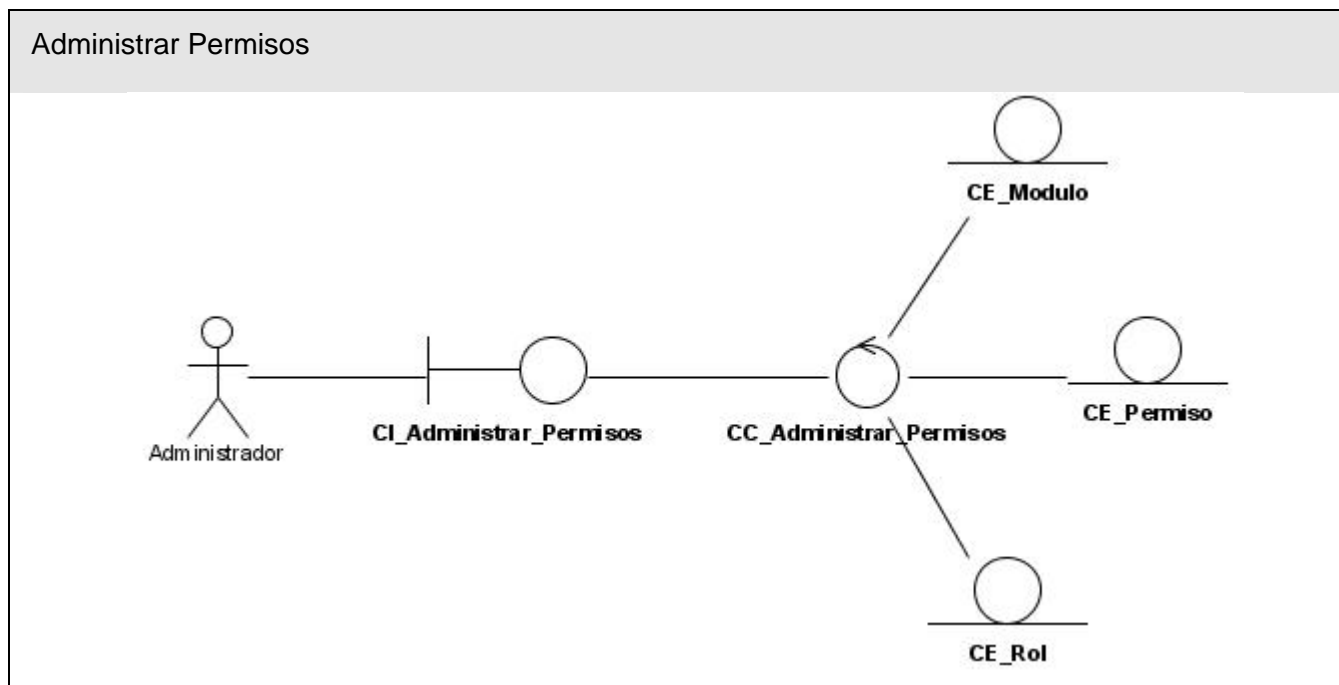
El Modelo de Análisis tiene el propósito de refinar los casos de uso más detalladamente, y realizar una asignación inicial del comportamiento del sistema; a un conjunto de objetos que proporcionen el funcionamiento esperado. Es un Modelo de Objetos que describe la realización de casos de uso, y sirve como abstracción para el Modelo del Diseño. (“RATIONAL_UNIFIED_PROCESS” 2003)

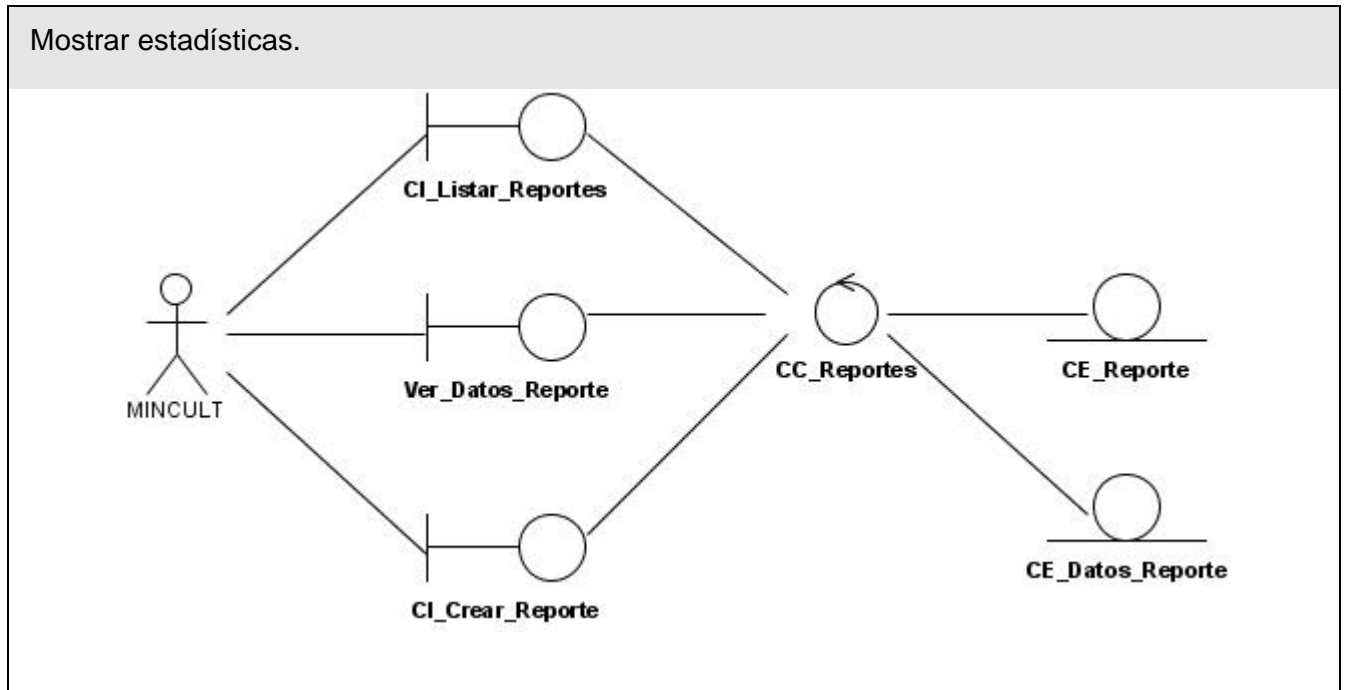
3.2.1 Diagramas de clases de análisis

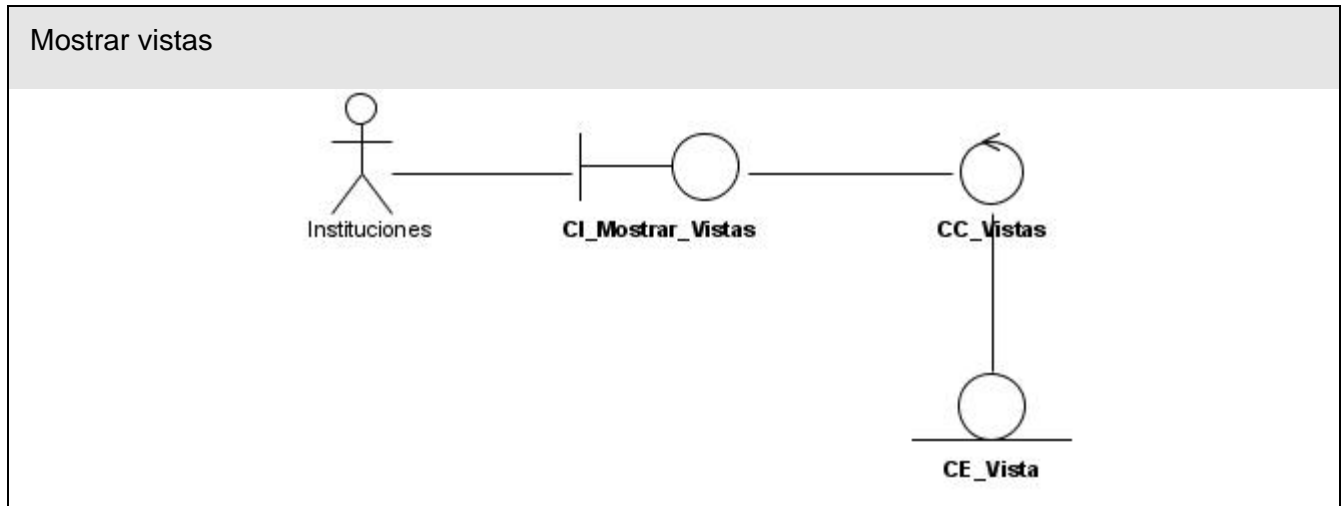




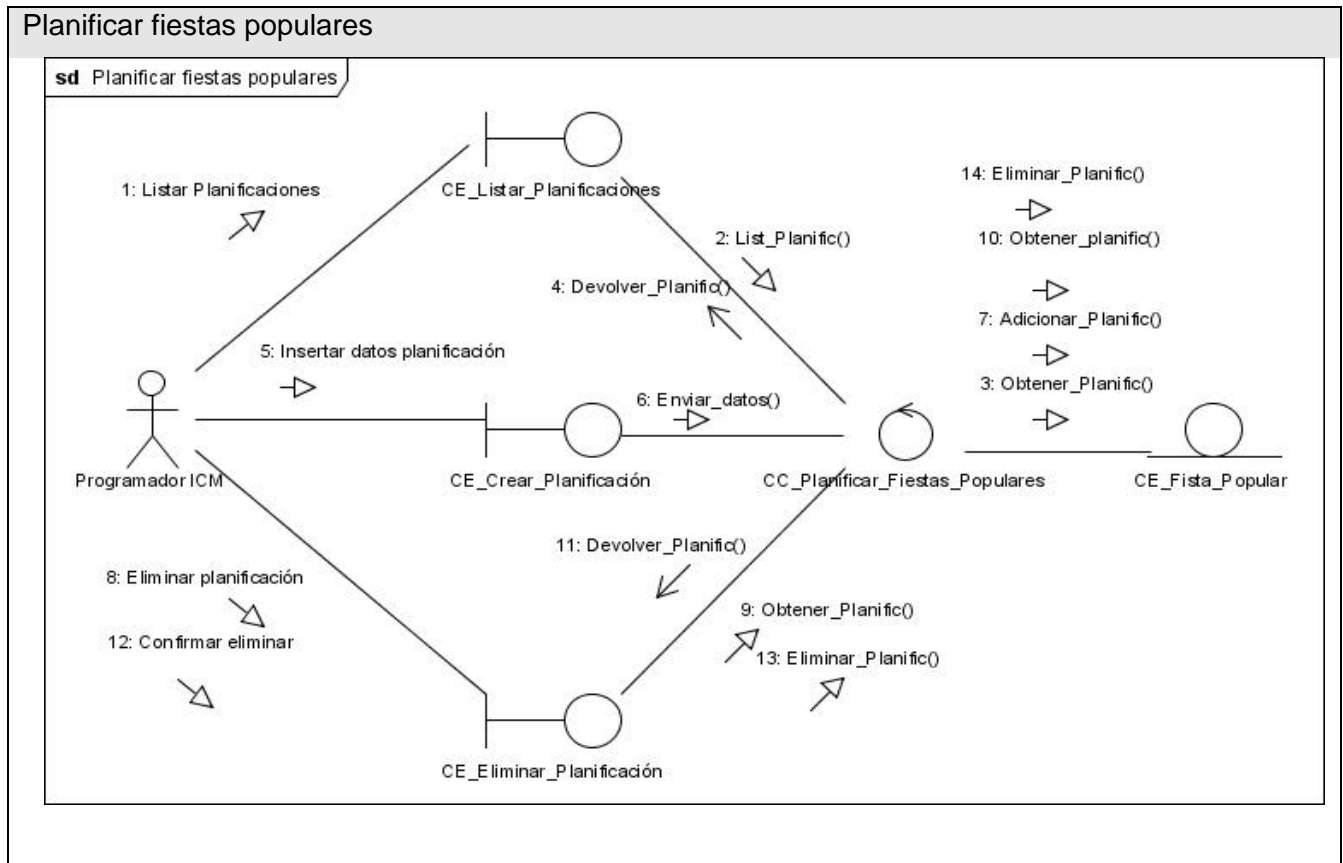


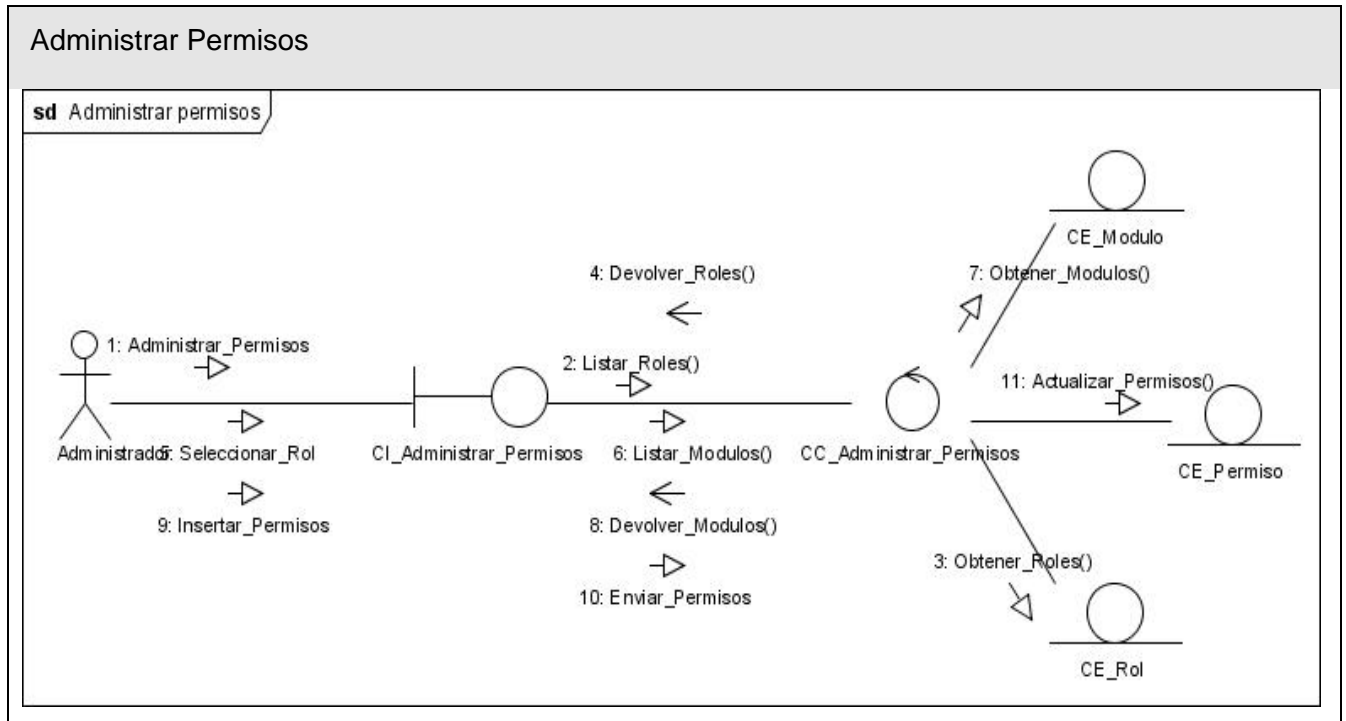


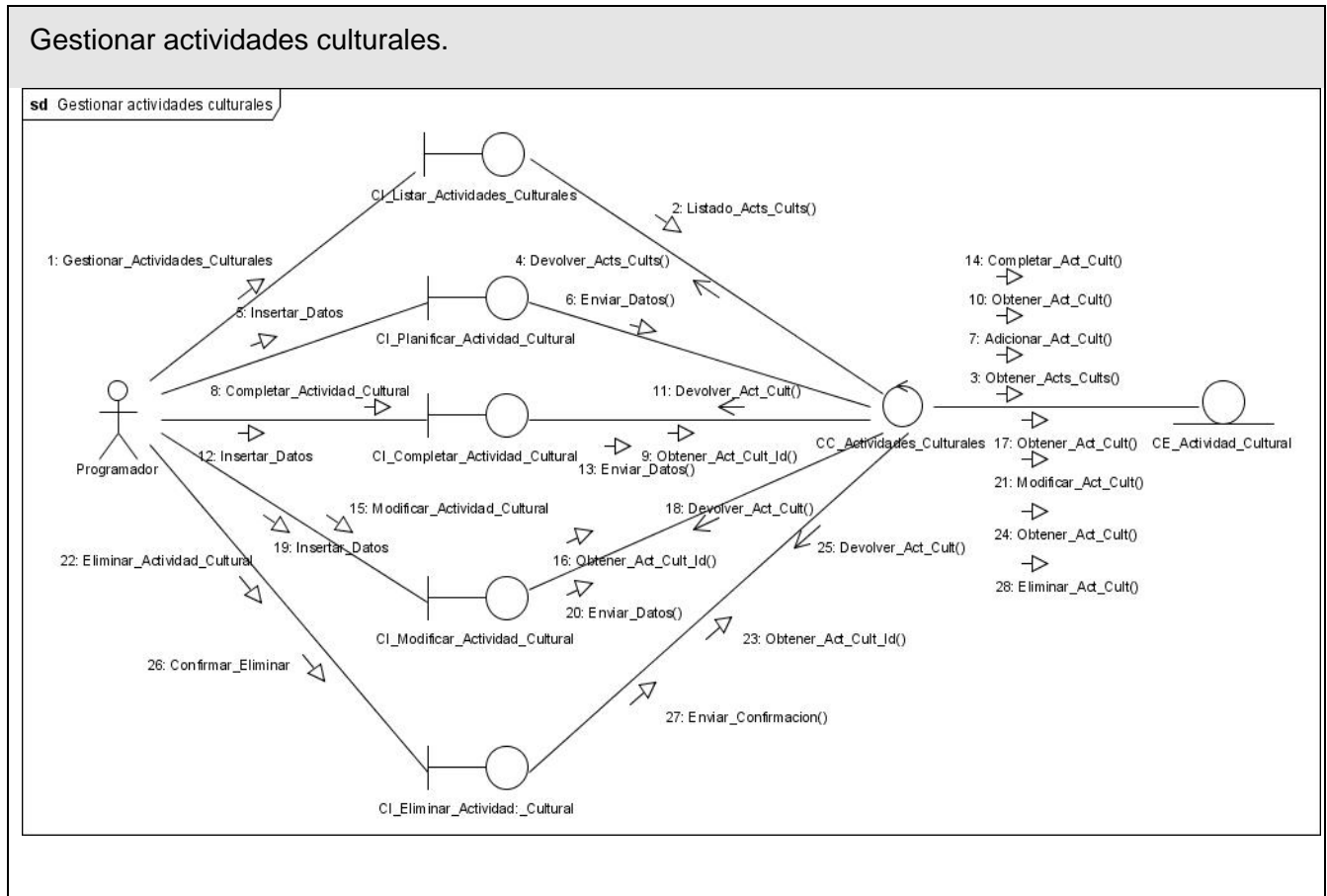


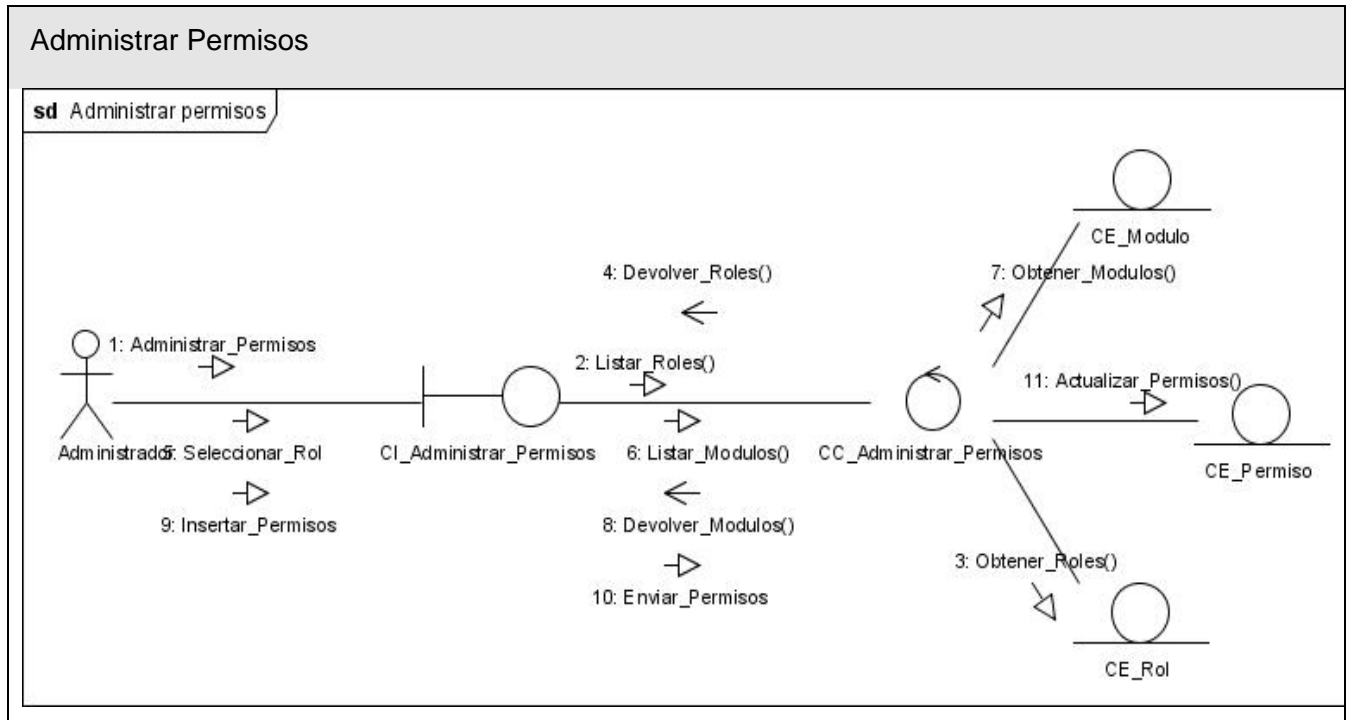


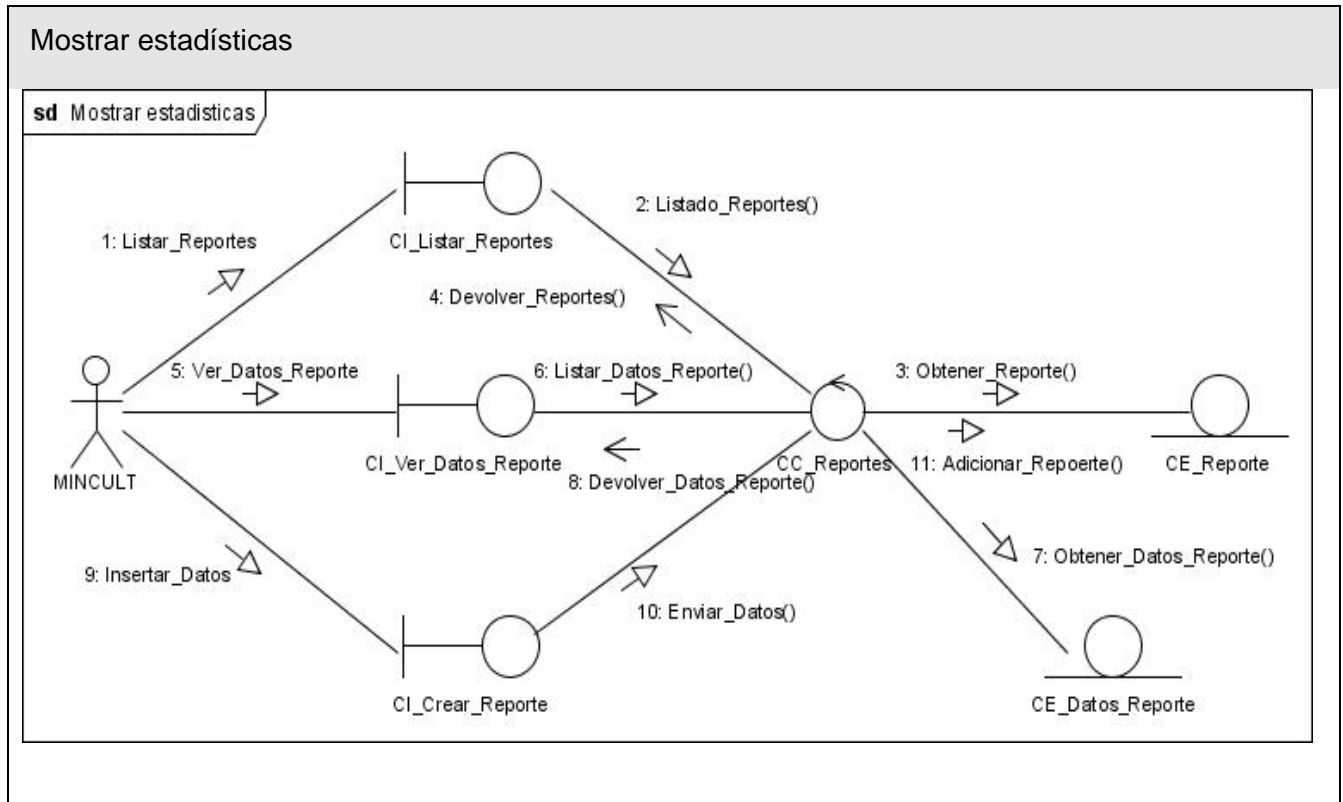
3.2.1 Diagramas de interacción

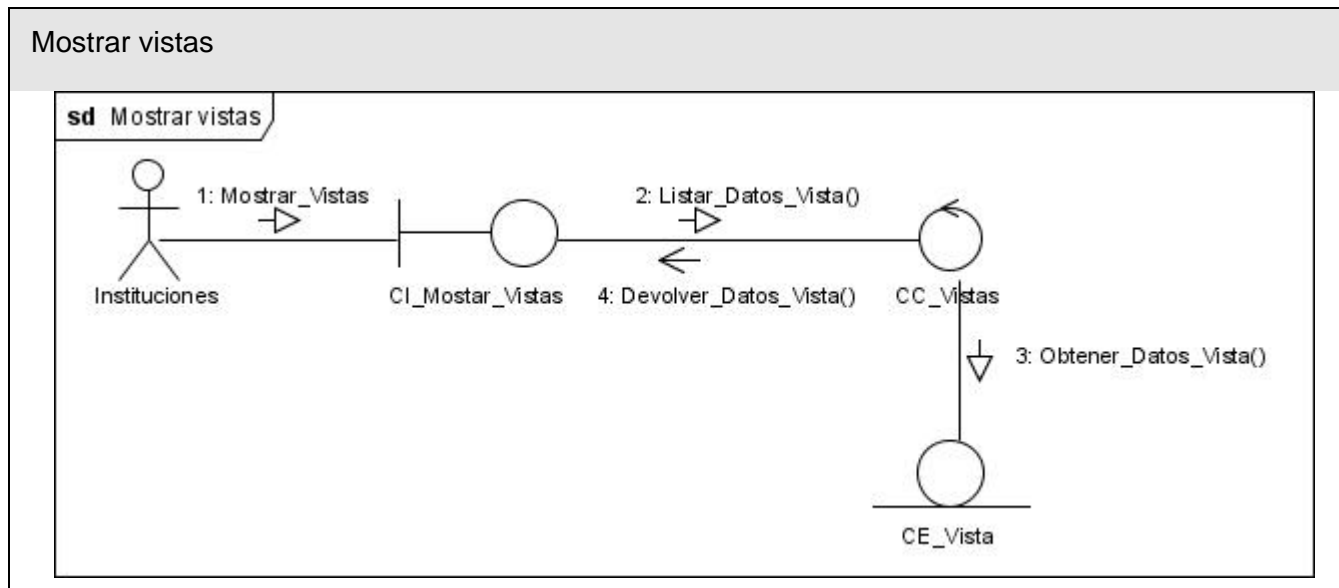










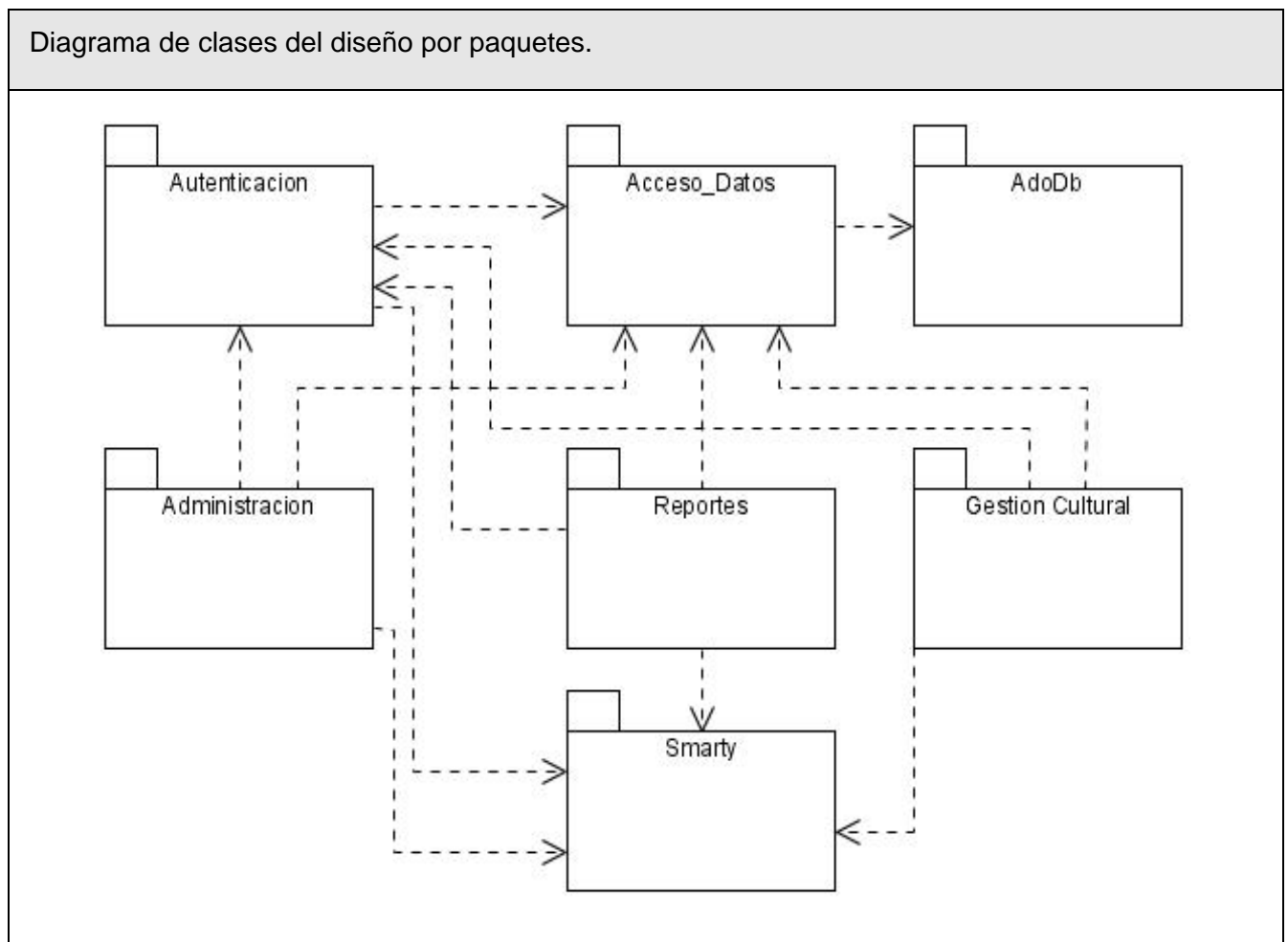


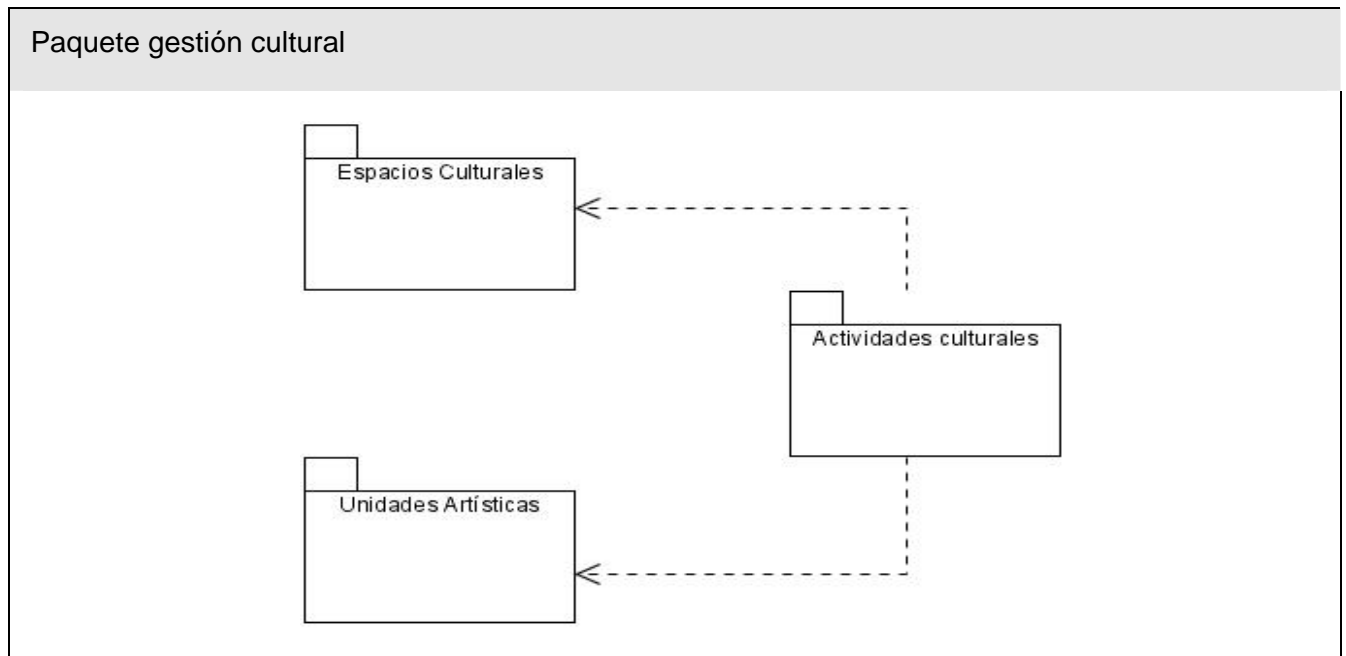
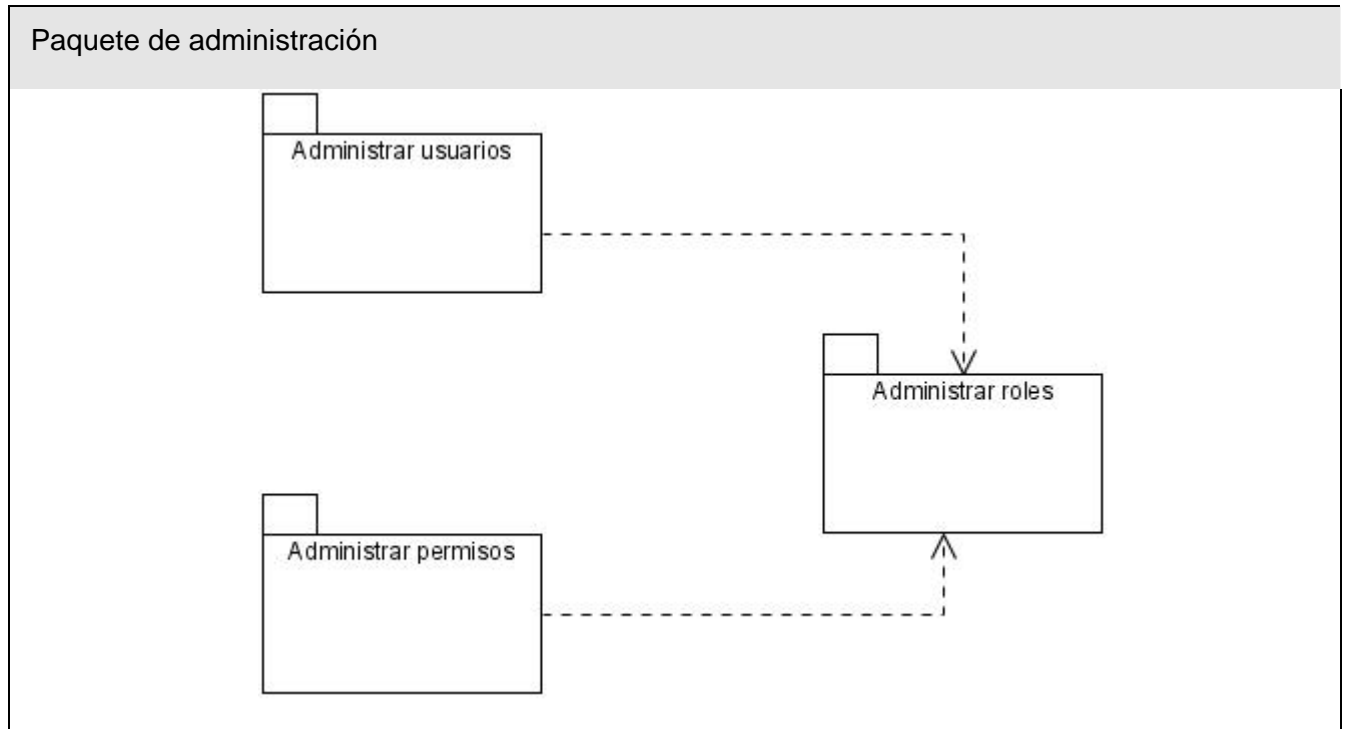
En este modelo fueron omitidos varios diagramas de análisis por ser muy similares a los que ya están representados. Los diagramas de los casos de uso, planificar proyección cinematográfica, planificar giras musicales y planificar giras CNAE son similares al de planificar fiestas populares. Los diagramas de gestionar espacios culturales y gestionar unidades artísticas son muy similares al de gestionar actividades culturales. Los diagramas de administrar roles y administrar usuarios son similares al de administrar permisos.

3.3 Modelo de diseño

El Modelo de Diseño define la estructura estática del sistema, tales como: subsistemas, clases e interfaces, y la realización de los casos de uso como colaboraciones entre los subsistemas, clases e interfaces. Es un Modelo de Objetos que describe la realización del caso de uso, y sirve como una abstracción del Modelo de Implementación y sus códigos fuentes. (“RATIONAL_UNIFIED_PROCESS” 2003)

3.3.1 Diagrama de clases de diseño





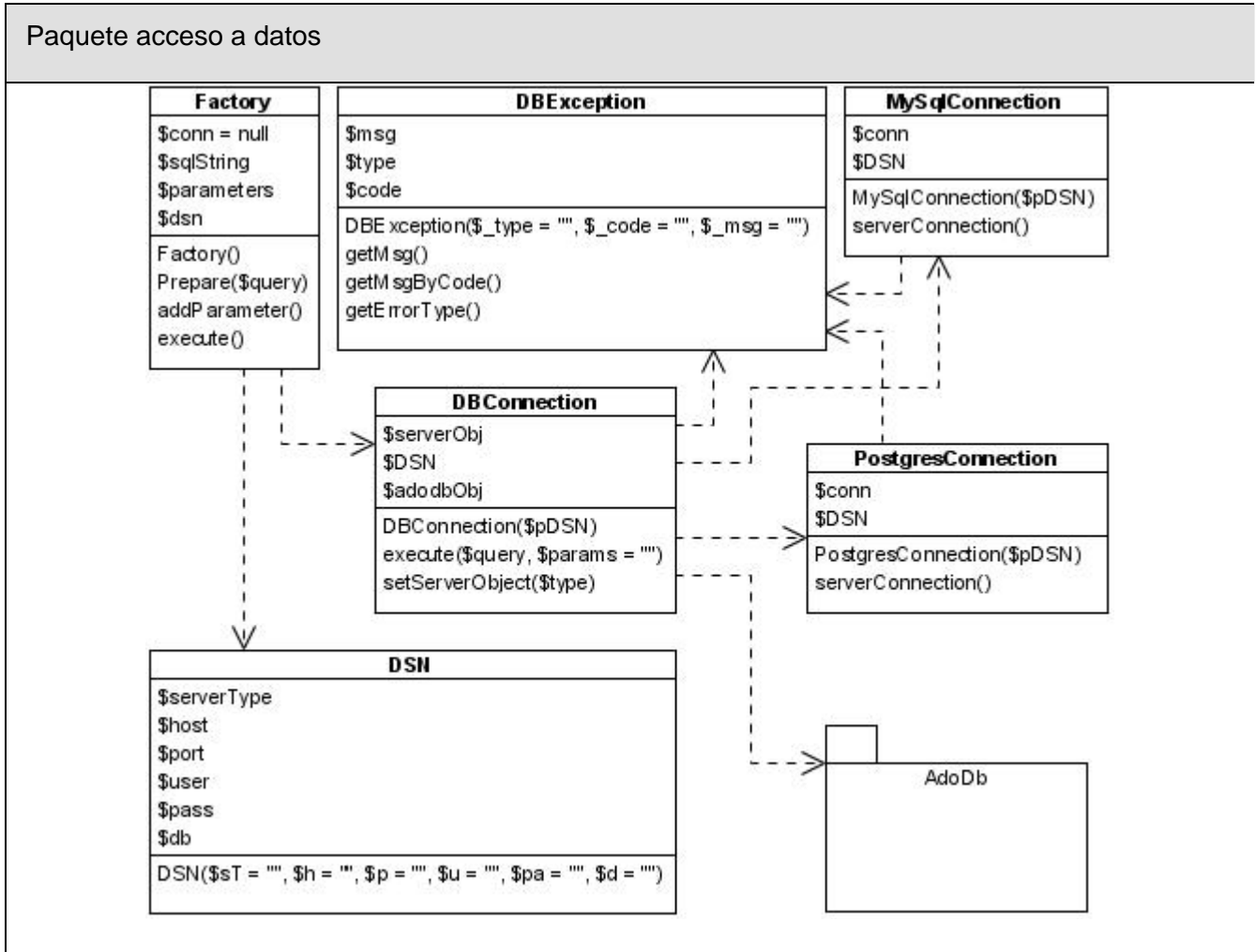


Diagrama de clases del diseño. Paquete de autenticación.

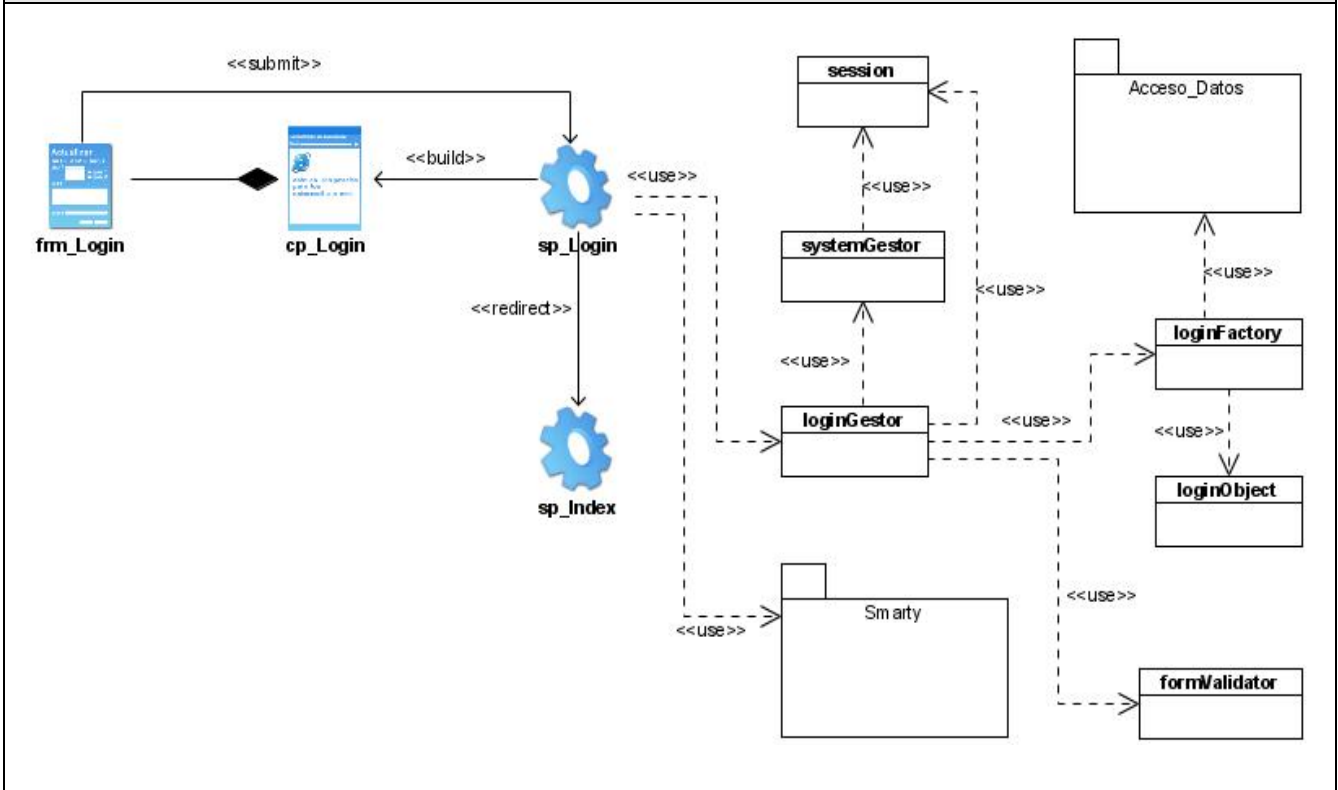
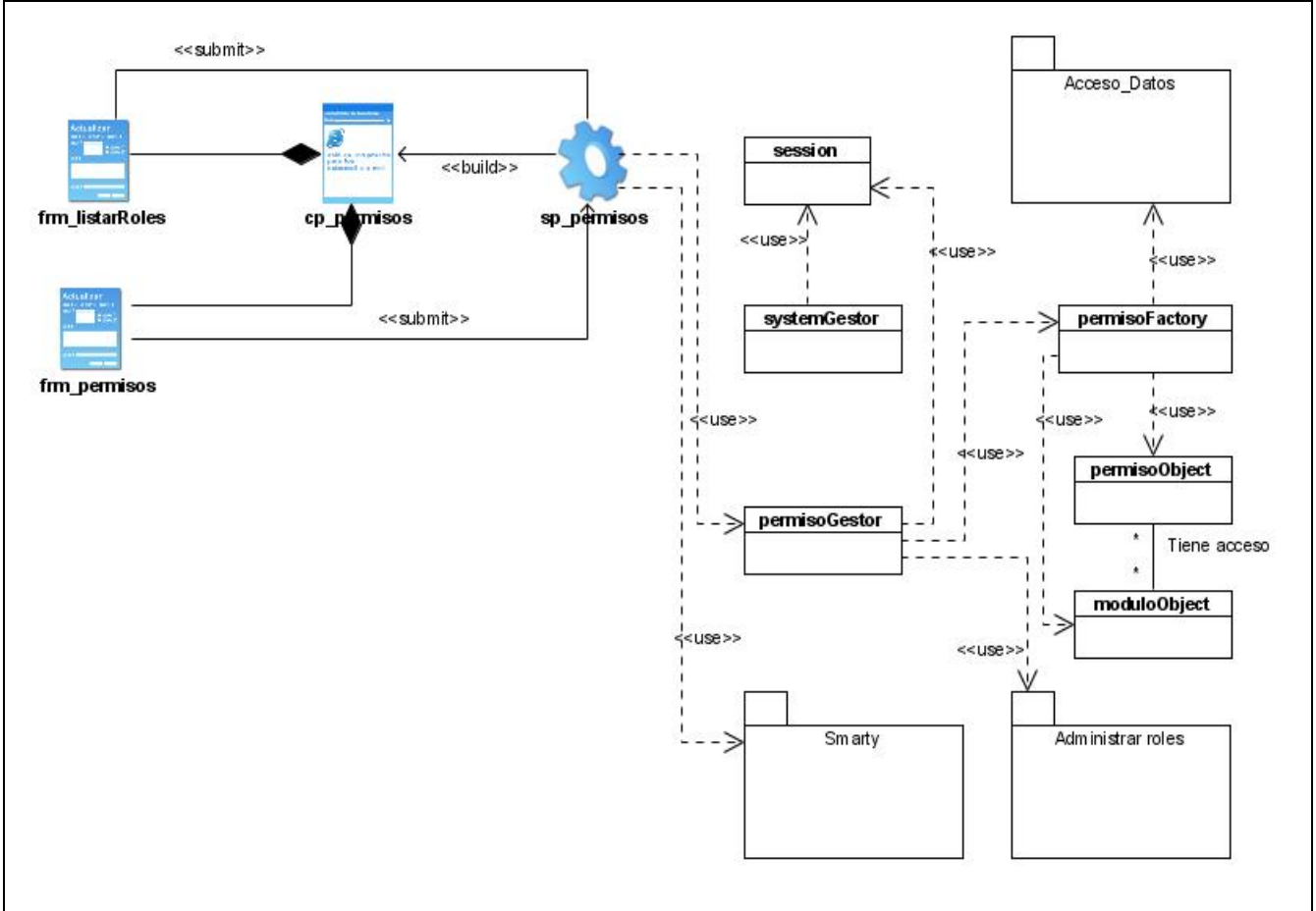
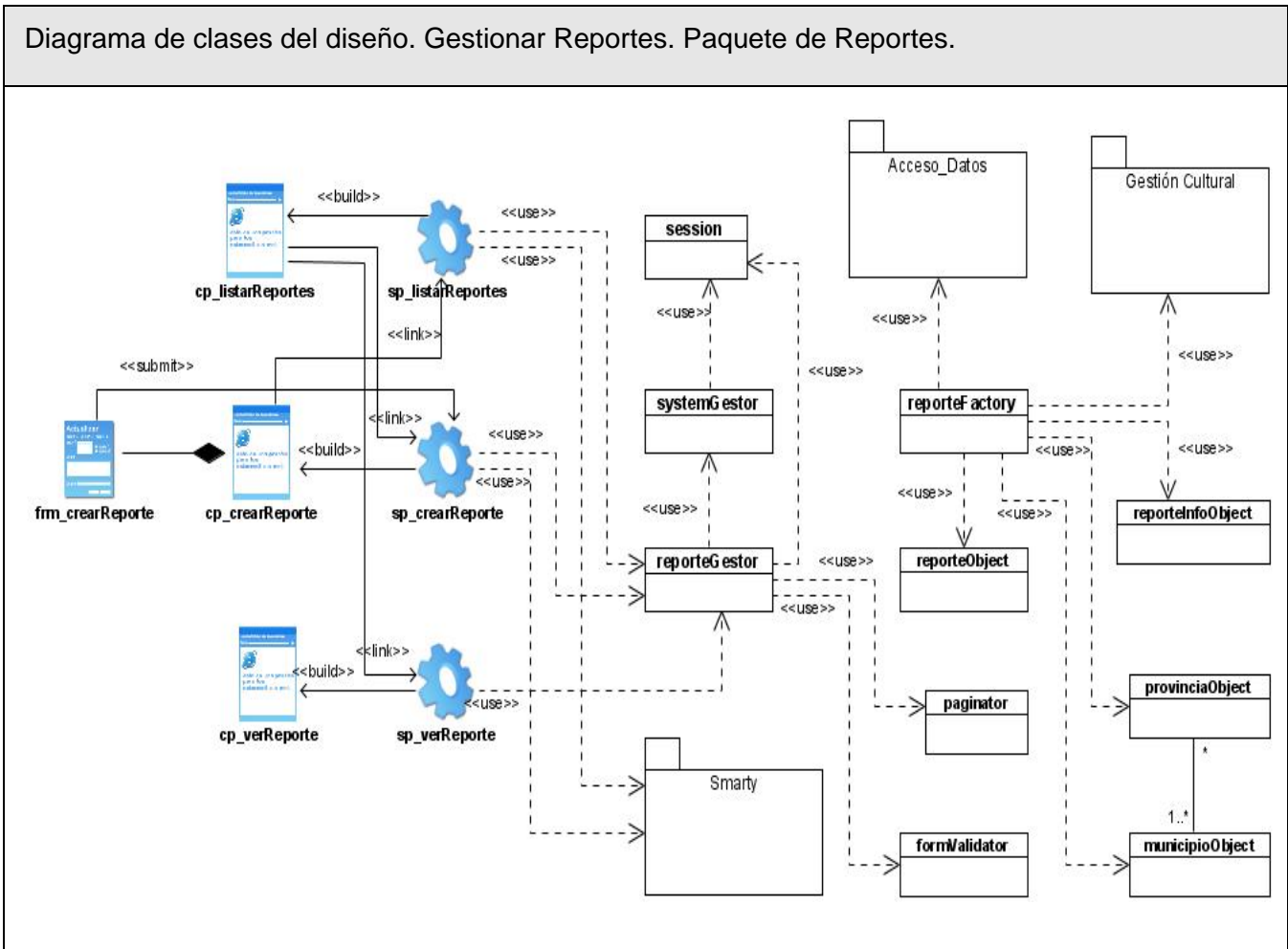


Diagrama de clases del diseño. Administrar permisos. Paquete de administración.





Todos los diagramas de clases de diseño pueden ser vistos en el anexo # 3, de este documento.

3.3.1.1 Descripción de las clases de diseño

Producto de la complejidad del diseño y la gran cantidad de clases que lo componen, se decidió describir solo las clases más importantes del sistema o sea las que son utilizadas por todas las clases controladoras del sistema.

Nombre: SystemGestor.	
Tipo de clase: controladora.	
Atributo	Tipo
errorMsg	Array
InformationMsg	Array
sessionMsg	Array
userObj	Object
moduleID	Int
Métodos:	
Nombre:	SystemGestor().
Descripción:	Constructor de la clase.
Nombre:	doAll().
Descripción:	Encargado de recibir los datos que el usuario envía por GET o POST y los asigna a los atributos de las clases gestoras de cada caso cuyos nombres coincidan.
Nombre:	VerificarAcceso().
Descripción:	Verifica si el usuario que esta intentando hacer determinada acción tiene permisos para ello.

Nombre:	addErrorMsg()
Descripción:	Adiciona al arreglo de mensajes de error un nuevo mensaje.
Nombre:	addinformacionMsg()
Descripción:	Adiciona al arreglo de información del mensaje de error un nuevo elemento.
Nombre:	addsessionMsg()
Descripción:	Adiciona un mensaje en la sesión del usuario para leerlo en la página a la que se redirecciona una vez realizada una acción.
Nombre:	resetValues().
Descripción:	Pone en vació a todos los arreglos que tienen que ver con la gestión del sistema.

Nombre: Session	
Tipo de clase: controladora.	
Atributo	Tipo
Métodos:	
Nombre:	initSession()
Descripción:	Hace una llamada el inicio de sesión de la variable global session de PHP.
Nombre:	issetSession(\$sessionVar)

Descripción:	Retorna verdadero si la sesión ya fue iniciada y fase si no.
Nombre:	createSession(\$sessionVar, \$sessionValue)
Descripción:	Crea una sesión para el usuario que se halla autenticado con éxito.
Nombre:	getSessionValue(\$sessionVar)
Descripción:	Devuelve el nombre de la sesión que esta activa, identificando el usuario.
Nombre:	destroySession()
Descripción:	Destruye la sesión, esta función es llamada una vez que el usuario sale del sistema.
Nombre:	unsetSession(\$sessionVar="")
Descripción:	Elimina todas la variables de la sesión.

Nombre: Factory.	
Tipo de clase: controladora.	
Atributo	Tipo
conn	int
sqlString	String
parameters	array
dsn	Object

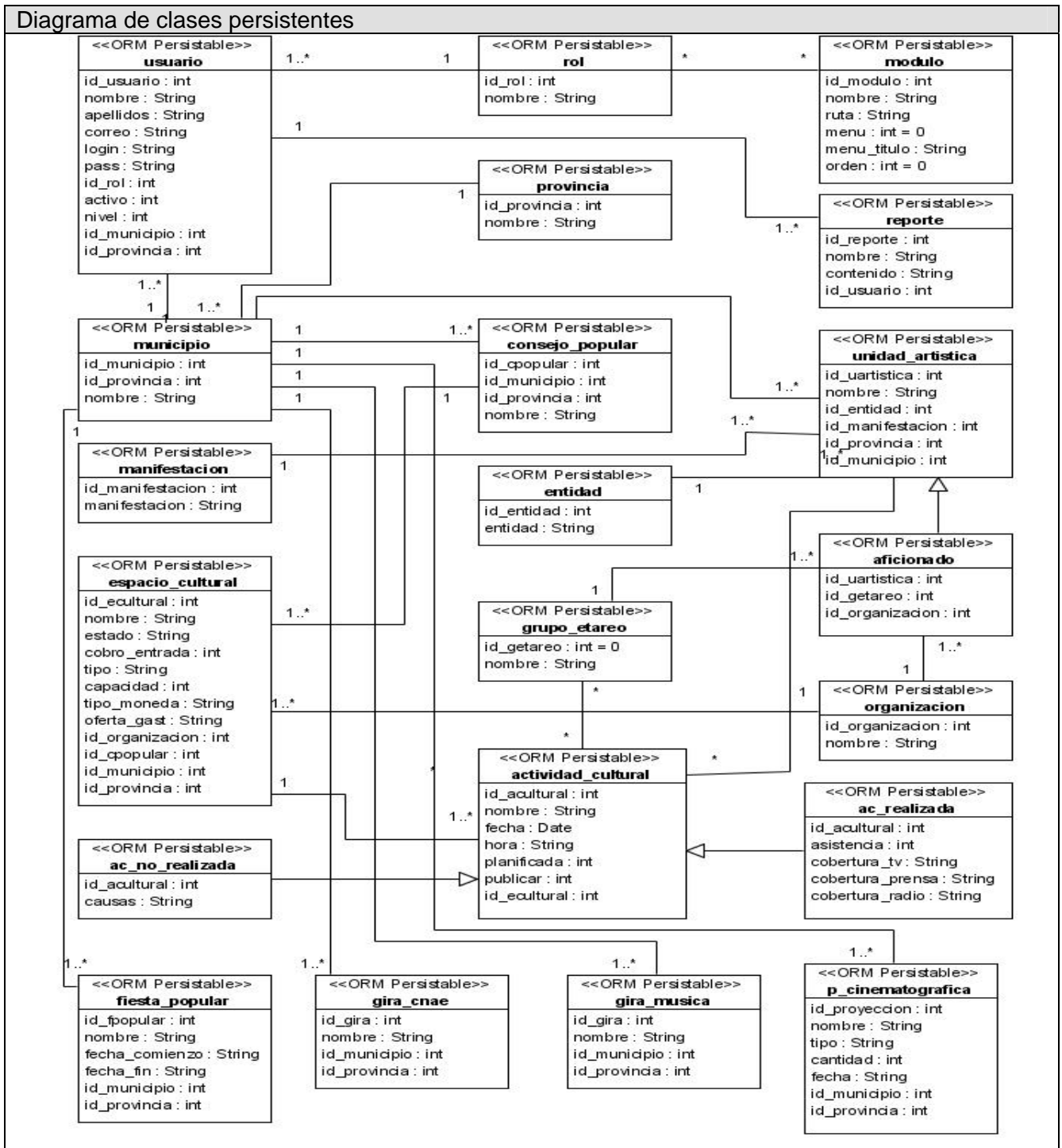
Métodos:	
Nombre:	Factory()
Descripción:	Constructor de la clase.
Nombre:	Prepare(\$query)
Descripción:	Le dice a la clase que consulta se va a ejecutar, aunque no la ejecuta, solo la asigna a la variable sqlString.
Nombre:	addParameter()
Descripción:	Adiciona a la consulta un parámetro de forma tal que sean valida la consulta para las clases ADOdb.
Nombre:	execute()
Descripción:	Ejecuta la consulta que esta en prepare().

Nombre: formValidator	
Tipo de clase: controladora.	
Atributo	Tipo
values	array
types	array
msg	array

msgError	array
count	int
Métodos:	
Nombre:	formValidator()
Descripción:	Constructor de la clase.
Nombre:	addValidation(\$_value, \$_type, \$_msg)
Descripción:	Adiciona al arreglo values el valor \$_value como un nuevo valor que debe ser validado y al arreglo msg el valor \$_msg como un nuevo mensaje que contiene el estado en que se encuentra el valor validado,
Nombre:	validateForm(&\$obj)
Descripción:	Valida el formulario y coloca los mensajes de error encontrados en el arreglo de mensajes del objeto.
Nombre:	checkValidation(\$_value, \$_type)
Descripción:	Verifica el tipo de validación con que se esté tratando.

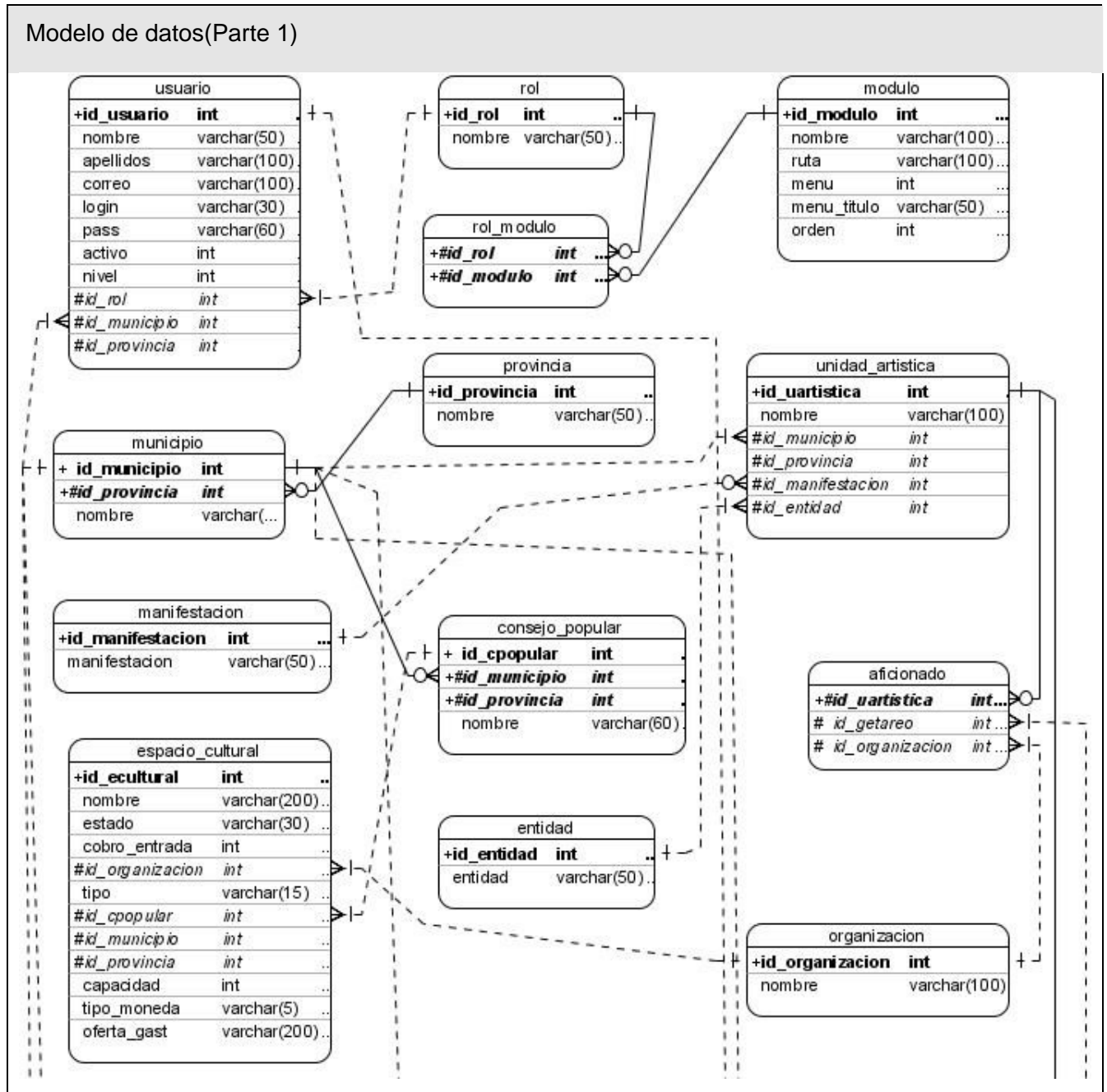
3.3.2 Diseño de la Base de Datos

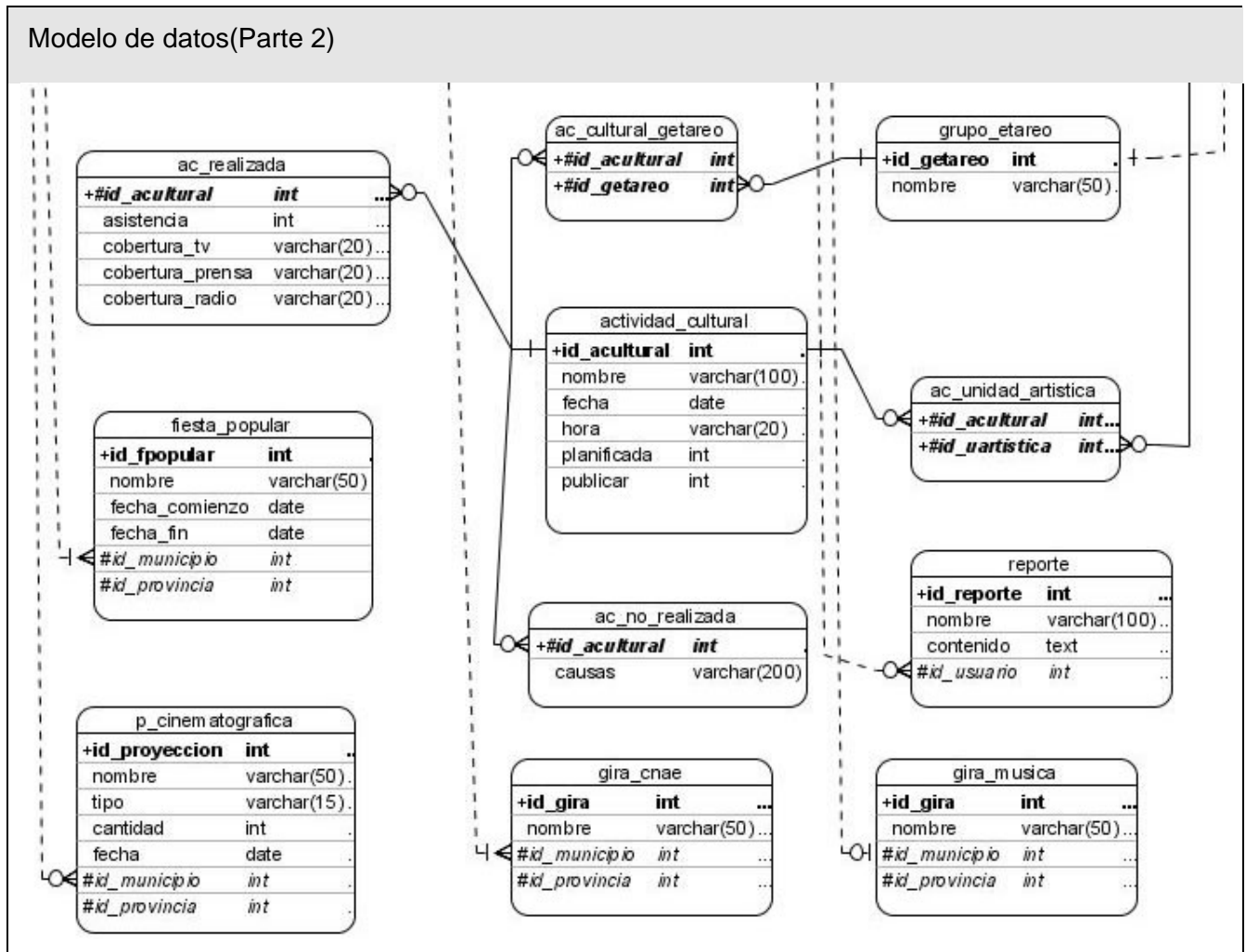
3.3.2.1 Diagrama de clases persistentes



3.3.2.2 Modelo de datos

Modelo de datos(Parte 1)





3.3.2.3 Descripción de las tablas

A continuación se describen cada una de de las tablas de la base de datos.

Nombre: usuario		
Descripción: Tabla que permite almacenar la información de los usuarios que van a interactuar con el sistema, además tiene relación con municipio.		
Atributo	Tipo	Descripción
id_usuario	integer(11)	Llave primaria de la tabla que asegura que aparezca la información de un usuario solo una vez.
nombre	varchar(50)	Nombre del usuario.
apellidos	varchar(100)	Apellidos del usuario.
correo	varchar(100)	Correo del usuario.
login	varchar(30)	Nombre con el que se identificará el usuario ante el sistema.
pass	varchar(60)	Contraseña para autenticarse ante el sistema.
Activo	integer(11)	Este campo solo toma dos valores posibles, 0 o 1 que indican si la cuenta de usuario esta o no activa.
nivel	integer(11)	Nivel de usuario, que le da en dependencia del valor de este campo privilegios al usuario.
id_rol	integer(11)	Llave foránea que vincula al usuario con el

		rol que desempeña en el sistema.
id_municipio	integer(11)	Llave foránea que vincula al usuario con el municipio al que pertenece.
id_provincia	integer(11)	Llave foránea que vincula al usuario con la provincia a la que pertenece.

Nombre: rol		
Descripción: Tabla que almacena la información de los roles que puede desempeñar un usuario.		
Atributo	Tipo	Descripción
id_rol	integer(11)	Llave primaria que identifica cada uno de los roles.
Nombre	varchar(50)	Nombre del rol.

Nombre: provincia		
Descripción: Tabla que almacena la información de las provincias.		
Atributo	Tipo	Descripción
id_provincia	integer(11)	Identificador de la provincia (PK)
nombre	varchar(50)	Nombre de la provincia.

Nombre: unidad_artistica		
Descripción: Almacena la información de las unidades artísticas de diferentes manifestaciones y géneros.		
Atributo	Tipo	Descripción
id_uartistica	integer(11)	Identificador de la unidad artística, es un valor único y la PK de la tabla.
nombre	varchar(100)	Nombre de la unidad artística.
id_municipio	integer(11)	Identificador del municipio al que pertenece la unidad artística FK.
id_provincia	integer(11)	Identificador de la provincia a la que pertenece la unidad artística FK.
id_manifestacion	integer(11)	Identificador la manifestación a la que pertenece la unidad artística FK.
id_entidad	integer(11)	Identificador la entidad a la que pertenece la unidad artística FK.

Nombre: aficionado		
Descripción: Almacena la información de las unidades artísticas aficionadas.		
Atributo	Tipo	Descripción
id_uartistica	integer(11)	Identificador de la unidad artística a la que pertenece.

id_getareo	integer(11)	Identificador del grupo etéreo al que pertenece el artista aficionado.
id_organizacion	integer(11)	Identificador de la entidad a la que pertenece el artista aficionado.

Nombre: organizacion		
Descripción: Almacena la información de las diferentes organizaciones a las que pertenecen las unidades artísticas.		
Atributo	Tipo	Descripción
id_organizacion	integer(11)	Identificador de la organización PK.
nombre	varchar(100)	Nombre de la organización.

Nombre: municipio		
Descripción: La información de los municipios del país es almacenada en esta tabla.		
Atributo	Tipo	Descripción
id_municipio	integer(11)	Identificador del municipio ante el sistema.
id_provincia	integer(11)	PK que viene de la tabla provincia e indica a que provincia pertenece el municipio.
nombre	varchar(50)	Nombre del municipio.

Nombre: modulo		
<p>Descripción: Es la tabla que almacena la información de los diferentes módulos con que cuenta el sistema, para poder acceder a ellos fácilmente pues almacena entre otras cosas la ruta física en el servidor.</p>		
Atributo	Tipo	Descripción
id_modulo	integer(11)	Identificador (PK) del módulo.
Nombre	varchar(100)	Nombre del módulo.
Ruta	varchar(100)	Camino físico hasta el módulo.
menu	integer(11)	Una variable que indica si el módulo va a aparecer o no en el menú de usuario.
menu_titulo	varchar(50)	El título que va a tener el módulo en el menú.
orden	integer(11)	Indica el orden en que va a aparecer el módulo en el menú de usuario.

Nombre: manifestación		
<p>Descripción: Contiene la información de cada una de las posibles manifestaciones a manejar en el sistema.</p>		
Atributo	Tipo	Descripción
id_manifestacion	integer(11)	Identificador de la manifestación.

manifestacion	varchar(50)	Nombre de la manifestación.
---------------	-------------	-----------------------------

Nombre: grupo_etareo		
Descripción: Grupos que maneja el sistema en cuanto a publico, dígase niños, jóvenes, adultos y tercera edad.		
Atributo	Tipo	Descripción
id_getareo	integer(11)	Identificador del grupo.
nombre	varchar(50)	Nombre del grupo.

Nombre: espacio_cultural		
Descripción: Almacena la información referente a los espacios dedicados por cada consejo popular a la cultura.		
Atributo	Tipo	Descripción
id_ecultural	integer(11)	Identificador (PK) de la tabla de espacios culturales.
Nombre	varchar(200)	Nombre del local.
Estado	varchar(30)	Estado en que se encuentra, puede ser en reparación, explotación o cerrado.
cobro_entrada	integer(11)	Si se cobra o no la entrada, en caso que se cobre se pone el precio.

id_organizacion	integer(11)	Identificador de la organización a la que pertenece (FK).
Tipo	varchar(15)	Tipo de espacio cultural.
id_cpopular	integer(11)	Identificador del consejo (FK) popular el que pertenece el espacio.
id_municipio	integer(11)	Identificador del municipio (FK) al que pertenece el espacio.
id_provincia	integer(11)	Identificador de la provincia (FK) a la que pertenece el espacio.
Capacidad	integer(11)	Capacidad de personas que tiene el espacio.
tipo_moneda	varchar(5)	Moneda en que se cobran la entrada y las ofertas gastronómicas del espacio cultural.
oferta_gast	varchar(200)	La oferta gastronómica del espacio cultural en caso de que no tenga no se especifica.

Nombre: actividad_cultural		
Descripción: Almacena la información referente a las actividades culturales planificadas.		
Atributo	Tipo	Descripción
id_acultural	integer(11)	Identificador de la actividad cultural.
Nombre	varchar(100)	Nombre de la actividad.

Fecha	date(0)	Fecha en que se debe realizar.
Hora	varchar(20)	Hora en que se debe realizar.
Planificada	integer(11)	Indica si esta planificada o si ya se realizo.
Publicar	integer(11)	Indica si el evento puede ser publicado en la cartela cultural del territorio.

Nombre: entidad		
Descripción: Información de las diferentes entidades que interactúan con el sistema.		
Atributo	Tipo	Descripción
id_entidad	integer(11)	Identificador de la entidad.
entidad	varchar(50)	Nombre de la entidad.

Nombre: consejo_popular		
Descripción: Información de los consejos populares en los que se realizan las actividades culturales.		
Atributo	Tipo	Descripción
id_cpopular	integer(11)	Llave primaria (PK) de la tabla.
id_municipio	integer(11)	Llave foránea (FK) que indica el municipio al que pertenece el consejo popular.

id_provincia	integer(11)	Llave foránea (FK) que indica la provincia a la que pertenece el consejo popular.
nombre	varchar(60)	Nombre del consejo popular.

Nombre: ac_realizada		
Descripción: Tabla para almacenar la información de las actividades realizadas.		
Atributo	Tipo	Descripción
id_acultural	integer(11)	Identificador de la tabla.
Asistencia	integer(11)	Cantidad de personas que asistieron.
cobertura_tv	varchar(20)	Si tuvo cobertura o no de la televisión cubana.
cobertura_prensa	varchar(20)	Si tuvo o no cobertura de prensa.
cobertura_radio	varchar(20)	Si tuvo o no cobertura radial.

Nombre: rol_modulo		
Descripción: Tabla que aparece producto de la relación de muchos a muchos entre las tablas rol y modulo e indica el acceso de los roles sobre los módulos.		
Atributo	Tipo	Descripción
id_rol	integer(11)	Identificador del rol.

id_modulo	integer(11)	Identificador del modulo.
-----------	-------------	---------------------------

Nombre: ac_cultural_getareo		
Descripción: Tabla que aparece producto de la relación de muchos a muchos entre las tablas act_cultural y getareo e indica que a que grupo esta destinado cada actividad.		
Atributo	Tipo	Descripción
id_acultural	integer(11)	Identificador de la actividad cultural.
id_getareo	integer(11)	Identificador del grupo.

Nombre: ac_no_realizada		
Descripción: Tabla para almacenar la información de las actividades culturales que no se pueden realizar.		
Atributo	Tipo	Descripción
id_acultural	integer(11)	Identificador de la actividad.
causas	varchar(200)	Causas por las que no se realizó.

Nombre: ac_unidad_artistica		
Descripción: Tabla que aparece producto de la relación de muchos a muchos entre las tablas act_cultural y unidad_artistica e indica las unidades artísticas que participan en una actividad		

cultural.		
Atributo	Tipo	Descripción
id_acultural	integer(11)	Identificador de la actividad cultural
id_uartistica.	integer(11)	Identificador de la unidad artística que participa.

Nombre: fiesta_popular		
Descripción: Tabla para almacenar al información de las fiestas populares de cada municipio.		
Atributo	Tipo	Descripción
id_fpopular	int(11)	PK de la tabla.
Nombre	varchar(50)	Nombre de la fiesta popular tendrá lugar.
fecha_comienzo	date(0)	Fecha en que inicia.
fecha_fin	date(0)	Fecha en que termina.
id_municipio	int(11)	Municipio donde se celebran las fiestas FK.
id_provincia	int(11)	Provincia a la que pertenece el municipio FK.

Nombre: gira_cnae		
Descripción: En esta tabla tiene lugar el almacenamiento de la información de las giras de las		

agrupaciones de artes escenicas.		
Atributo	Tipo	Descripción
id_gira	int(11)	Identificador de la gira PK
nombre	varchar(50)	Nombre de la gira en caso que lo tenga.
id_municipio	int(11)	Municipio en el que se encuentra girando la agrupación FK.
id_provincia	int(11)	Provincia a la que pertenece el municipio FK.

Nombre: gira_musica		
Descripción: Almacena la información de las giras de las agrupaciones musicales.		
Atributo	Tipo	Descripción
id_gira	int(11)	Identificador de la gira PK
nombre	varchar(50)	Nombre de la gira en caso que lo tenga.
id_municipio	int(11)	Municipio en el que se encuentra girando la agrupación FK.
id_provincia	int(11)	Provincia a la que pertenece el municipio FK.

Nombre: p_cinematografica

Descripción: Tabla para almacenar al información de las proyecciones cinematográficas en los cines del país.		
Atributo	Tipo	Descripción
id_proyeccion	Int(11)	Llave primaria de la tabla.
Nombre	varchar(50)	Nombre de la película a proyectar.
Tipo	int(11)	Tipo de proyección.
cantidad	int(11)	Cantidad de personas que asisten a ver la película.
Fecha	date(0)	Fecha en que se proyecta.
id_municipio	int(11)	Municipio en el que se encuentra girando la agrupación FK.
id_provincia	int(11)	Provincia a la que pertenece el municipio FK.

3.4 Patrón de diseño utilizado.

“Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software.” En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Debemos tener presente los siguientes elementos de un patrón: su nombre, el problema, la solución y las consecuencias.(TEDESCHI fecha desconocida)

El patrón de diseño que se utilizó para dar solución al problema planteado en este trabajo fue modelo – vista – controlador, en los diagramas de clases de diseño se pueden observar a simple vista las clases que pertenecen a cada parte del patrón. La vista representada por las interfaces, el modelo por las

entidades y el controlador por el grupo de clases controladoras que se encuentran mediando funciones entre las dos anteriores.

3.5 Interfaces.

Interfaz propuesta para la implementación del sistema.

The screenshot displays the 'Gestcult' web interface. At the top, the logo 'Gestcult' is shown with the subtitle 'Sistema Nacional de Programación Cultural'. The breadcrumb trail reads 'Inicio » Gestión Cultural » Potencial artístico'. A 'Módulo:' dropdown menu is set to 'Gestión Cultural'. Below this is a 'Formulario de búsqueda' with fields for 'Nombre', 'Tipo', 'Provincia', 'Municipio', 'Estado', 'Manifestación', and 'Centro de pertenencia', along with a 'Buscar' button. A table lists three cultural entities, all of which are 'Profesional' and have a 'Estado' of '✓'. A right-hand sidebar contains 'Opciones' for user profile management, including 'Ver perfil' and 'Cerrar Sesión', and a navigation menu for 'Gestión Cultural' with sub-items like 'Potencial artístico', 'Espacios culturales', and 'Actividades culturales'.

#	Nombre	Tipo	Estado
1	<input type="radio"/> Charanga Habanera	Profesional	✓
2	<input type="radio"/> Grupo Sierra Maestra	Profesional	✓
3	<input type="radio"/> Los Van Van	Profesional	✓

Conclusiones

En este trabajo se demostró la necesidad de diseñar e implementar un sistema que facilite los procesos del departamento de programación del MINCULT, entre los que se destacan los procesos de información, solicitudes y control por parte de los responsables de las diferentes esferas del departamento.

Luego de un análisis de las tecnologías más usadas en la actualidad para la construcción de sistemas informáticos similares, se elaboró la propuesta de utilizar el lenguaje PHP, con su motor de plantillas: Smarty y su capa de acceso a datos: ADOdb, el gestor de bases de datos MySQL y el servidor Web Apache para la construcción de la solución propuesta, junto a otras tecnologías como son javascript y las hojas de estilo CSS. Se decidió también que la metodología idónea para llevar a cabo el proceso de desarrollo es RUP (Proceso Unificado de Rational). Se analizó el negocio actual y se modeló la propuesta de uno, con mejoras sustanciales en cuanto a organización de la información y los canales por los que viaja la misma, se identificaron los actores y trabajadores del negocio y se plantearon los requerimientos funcionales y no funcionales del sistema, describiéndose cada caso de uso para una mejor comprensión de la funcionalidad que brindan. Se diseñó el sistema, a través de diagramas de análisis, de clases Web, diagrama de clases persistentes, entre otros. Se estructuró el modelo de datos, que es la representación física de la base de datos. Quedó listo el sistema para pasar a las fases siguientes según establece RUP que basado en el diseño propuesto será implementado con la calidad requerida para ser entregado a la institución interesada.

Recomendaciones

Se recomienda el desarrollo iterativo del trabajo para que sea implementado con éxito y se pueda satisfacer las expectativas del cliente. Que se analicen los restantes procesos del departamento de programación y las instituciones que no se contemplaron en este trabajo. La extensión del sistema a otros departamentos del MINCULT y el uso de este trabajo en otros centros con características semejantes

Referencias Bibliográficas

“RATIONAL_UNIFIED_PROCESS”. *Ayuda del RUP. Suite del Rational* 2003.

BARCIA, D. <http://www.maestrosdelweb.com/editorial/introcss/>, fecha desconocida.

CALDERON, P. M. C. A. D. P. M., 2003.

CONFERENCIAS Tomado de las conferencias de bases de datos de la UCI, Conferencia # 1: Introduccion a la asignatura, 2005.

MOLPECERES, A. Procesos de desarrollo: RUP, XP y FDD, 15/12/2002.

SCHMULLER, J. *Aprendiendo UML en 24 hora*. 2000. p.

TEDESCHI, N. *¿Que es un patron de diseño?*, fecha desconocida.

VAZQUEZ, J. A. G. *Desarrollo web con PHP y Mysql*, 2003 p.

Bibliografía

"RATIONAL_UNIFIED_PROCESS". *Ayuda del RUP. Suite del Rational* 2003.

ALVAREZ, M. A. <http://www.desarrolloweb.com/articulos/497.php>, fecha desconocida

ANTONIO, A. D. *La Gestión de la Configuración del Software*. p.

BARCIA, D. <http://www.maestrosdelweb.com/editorial/introcss/>, fecha desconocida.

CALDERÍN., E. E. H. *¿Cómo elaborar un diseño de investigación en Gráfica de Ingeniería?*, 2000.

CALDERON, P. M. C. A. D. P. M., 2003.

CONFERENCIAS Tomado de las conferencias de bases de datos de la UCI , Conferencia # 1: Introducción a la asignatura, 2005.

GUIA_DE_USUARIO_VP. *Guia_de_usuario_VP*. 2006. p.

JAMES RUMBAUGH, I. J., GRADY BLOOCH. *El lenguaje unificado de modelado*. fecha desconocida. p.

MCCN, C. *Accesibilidad y XML*, fecha desconocida.

MICROSOFT.COM. <http://www.microsoft.com/sql/prodinfo/default.msp>, fecha desconocida.

MOLPECERES, A. *Procesos de desarrollo: RUP, XP y FDD*, 15/12/2002.

PAGINA_OFICIAL. <http://www.python.org/about/>, fecha desconocida.

PAGINA_OFICIAL_DE_PHP. <http://www.php.net/>, fecha desconocida.

PRESSMAN, R. S. *Ingeniería de Software un enfoque practico*, 2002.

SANCHEZ, M. A. M. *Metodologías de desarrollo de software*, 7/7/2004.

SCHMULLER, J. *Aprendiendo UML en 24 hora*. 2000. p.

STALLMAN, R. M. *Software Libre para una sociedad libre*, 2004.

TEDESCHI, N. *¿Que es un patron de diseño?*, fecha desconocida.

VAZQUEZ, J. A. G. *Desarrollo web con PHP y Mysql*, 2003 p.

XML, S. *Extensible Markup Language*, fecha desconocida.

Glosario de Términos

SGDB: Sistema gestor de base de datos.

MINCULT: Ministerio de cultura del estado cubano.

ICL: Instituto Cubano del Libro.

CNAE: Consejo Nacional de Artes Escénicas.

WYSIWYG: What You See Is What You Get (en español: “lo que ves es lo que obtienes”). Para describir un editor de texto que permite escribir un documento viendo directamente el resultado final, frecuentemente el resultado impreso.

UML: Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modelling Language) es el lenguaje de modelado de sistemas de software más conocido en la actualidad.

JSP: Java Server Pages.

SWL: Software libre.

Smarty: Clases escritas en PHP para gestionar plantillas html.

ADODB: Clases escritas en PHP para acceso a distintos gestores de bases de datos.

PK: Término usado para identificar la llave primaria de una tabla.

FK: Término usado para identificar la llave foránea de una tabla.

XML: Metalenguaje capaz de definir sus propias etiquetas.

CSS: Hojas de estilos para darle al HTML el aspecto deseado.