

Universidad de las Ciencias Informáticas

FACULTAD 10.



**Análisis y diseño del Portal Nacional de
Programación Cultural**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

Autor: Yudisleydis Méndez Durán.

Tutor: Ing. Yusliel García Vázquez.

Ciudad Habana, Junio del 2008

“Año 50 de la Revolución”

***Quienquiera que espere ver
una obra perfecta, espera algo
que nunca fue, es ni será.***

Alexander Pope.

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 _____.

Yudisleydis Méndez Durán

Yusliel García Vázquez

Datos de Contacto.

Ing. Yusliel García Vázquez

Ingeniero en Ciencias Informáticas, graduado de la primera graduación de la UCI, actualmente con categoría docente de Instructor Recién Graduado, imparte la asignatura de Programación I y II, ha trabajado en proyectos productivos relacionados con el tema de programación Web y actualmente se desempeña como Jefe de Equipo de desarrollo del Portal Institucional del MENPET.

Agradecimientos.

Agradezco a mis padres, Magalis y Ernesto por haberme guiado hasta aquí, y convertirme en la persona que soy hoy.

A mi hermano Ernesto por inspirarme a ser un ejemplo para él.

A mi novio Orlando por brindarme apoyo y amor durante toda la carrera.

A Edilia, Regino, Yanet y Onell por haberme acogido como un miembro más de la familia.

A toda mi familia por su cariño y paciencia.

A mi tutor Yusliel por ser preocupado y ayudarme.

A mis amigos con los que he compartido todos estos años.

A todas las personas que de una forma u otra me ha ayudado o han contribuido a este trabajo.

Dedicatoria

A mi mamá, mi papá y mi hermano por su apoyo y cariño de toda la vida.

Resumen

El presente trabajo propone una solución desde el punto de vista de análisis y diseño de un Portal que automatice el proceso de promoción de las carteleras culturales a nivel nacional del Ministerio de Cultura. En la actualidad la información no está organizada, pues cuentan con diversas carteleras de distintas instituciones culturales que responden solo a los intereses de las mismas y son monotemáticas. Se pretende que todos los usuarios nacionales se informen sobre todas las actividades que se efectúan tanto en la capital y las cabeceras de provincia como en todos los municipios, utilizando la base de datos del Sistema Nacional de Programación Cultural que ya cuenta con la mayoría de dichos datos.

En este trabajo se realiza el análisis y diseño de una aplicación Web donde se identifican y describen los procesos fundamentales, que al ser desarrollada resolverá las demandas del MINCULT de promocionar las actividades culturales utilizando carteleras; gestionando información acerca de las actividades, potencial artístico y los espacios culturales además de enviar un boletín semanal.

Tabla de contenido

INTRODUCCIÓN	1
CAPÍTULO1: FUNDAMENTACIÓN TEÓRICA	4
1.1 INTRODUCCIÓN	4
1.2 GESTOR DE BASE DE DATOS MYSQL	4
1.3 LENGUAJES DE PROGRAMACIÓN	5
<i>Java</i>	5
<i>Practical Extracting and Reporting Language (Perl)</i>	5
<i>Phyton</i>	6
<i>Ruby on Rails</i>	6
<i>Hypertext Preprocessor (PHP)</i>	7
1.4 GESTORES DE CONTENIDO	7
<i>Joomla</i>	7
<i>Mambo</i>	8
<i>Drupal</i>	8
1.5 TECNOLOGÍAS DEL LADO DEL CLIENTE	9
<i>JavaScript</i>	9
<i>Cascade Style Sheets (CSS)</i>	9
<i>Hyper Text Markup Language (HTML)</i>	10
<i>Extensible Markup Language XML (XML)</i>	10
1.6 METODOLOGÍAS DE DESARROLLO DE SOFTWARE	10
<i>Extreme Programing (XP)</i>	10
<i>Microsoft Solution Framework (MSF)</i>	11
<i>Rational Unified Process (RUP)</i>	13
1.7 HERRAMIENTAS CASE.....	14
1.7.1 Umbrello	14
1.7.2 Visual Paradigm	14
CAPÍTULO2: CARACTERÍSTICAS DEL SISTEMA	16
2.1 INTRODUCCIÓN	16
2.2 DESCRIPCIÓN DE LOS PROCESOS DEL NEGOCIO PROPUESTOS	16
2.3 MODELO DEL DOMINIO.....	16
2.3.1 <i>Definición de los roles del dominio</i>	17
2.3.2 <i>Diagrama del modelo del dominio</i>	17
2.4 REQUISITOS	18
2.4.1 <i>Requisitos Funcionales</i>	18
2.4.2 <i>Requisitos no Funcionales</i>	23

2.5 DESCRIPCIÓN DEL SISTEMA.....	24
2.6 MODELO DE CASOS DE USO DEL SISTEMA	24
2.6.1 Descripción de los casos de uso del sistema	25
2.6.2 Diagrama de los casos de uso del sistema	28
2.6.3 Casos de uso expandidos	29
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	41
3.1 INTRODUCCIÓN.	41
3.2 MODELO DE ANÁLISIS.....	41
3.2.1 Diagramas de clases de análisis	41
3.2.2 Diagramas de interacción.....	44
3.3 MODELO DE DISEÑO	48
3.3.1 Diagrama de clases del diseño	49
3.3.1.1 Descripción de las clases de diseño.....	55
3.3.2 Diseño de la Base de Datos.....	68
3.3.2.1 Diagrama de clases persistentes.....	68
3.3.2.2 Modelo de datos	69
3.3.2.3 Descripción de las tablas.....	69
3.4 PATRÓN DE DISEÑO UTILIZADO.....	74
3.5 INTERFACES	75
CONCLUSIONES.....	77
RECOMENDACIONES	78
REFERENCIAS BIBLIOGRÁFICAS	79
BIBLIOGRAFÍA CONSULTADA:	80
GLOSARIO DE TÉRMINOS.....	83
ANEXOS.....	84
ANEXO 1. DIAGRAMAS DE CLASES DE ANÁLISIS.....	84
ANEXO 2. DIAGRAMAS DE COLABORACIÓN.....	87
ANEXO 3. DIAGRAMAS DE CLASES DEL DISEÑO	90

Introducción

Como resultado de disímiles dificultades con las que contaba el Ministerio de Cultura para gestionar la información relacionada con la programación de las actividades culturales que se realizan por todo el país como lo son las giras, los carnavales, las ferias, los conciertos, etc., se creó el Sistema Nacional de Programación Cultural pero aún no se cuenta con una aplicación que muestre o publique toda esta información almacenada en dicho sistema, relacionada con los quehaceres culturales de la nación. Además el sistema nacional de cultura ha tenido a lo largo de los años, entre otras tareas, la de brindar al público carteleras por meses de manifestaciones artísticas o culturales para promoverlas y con ello alcanzar un elevado índice de participantes. Por lo anterior expuesto surge la pregunta ¿Cómo se está dando a conocer al público la información referente a actividades culturales?

En la actualidad se publican varias carteleras en formato duro y otras más numerosas a través de la Web. Este último caso responde tanto a las de carácter nacional como la de CREART, otras generales como la Jiribilla, temáticas de Institutos y Consejos (Cubaescena, Cubacine), y las provinciales y municipales. Cada municipio cuenta con una página Web en la que publican toda la información referente a sus actividades culturales. Se pretende centralizar toda la información referente a dichas actividades como por ejemplo: el costo, la agrupación que va a tocar, etc. Siempre con el objetivo de que toda esta información que se manipula en el departamento de programación cultural de la institución sea vista desde un sitio a nivel central.

Por lo anterior planteado se tomó la decisión de hacer un Portal Web que muestre al público la información que brinda el Sistema Nacional de Programación Cultural, además de otros datos de interés que no se encuentran almacenados en la base de datos de dicho sistema, permitiéndole al usuario interactuar con ella y buscar información sobre sus artistas favoritos, como pueden ser el lugar de presentación; así como un boletín informativo con las principales actividades de la semana.

Por lo tanto el **problema** a resolver debe responder a la pregunta ¿Cómo diseñar un portal Web que de solución a los problemas de acceso y promoción de las actividades culturales del sistema nacional de programación cultural y gestione nueva información?

El **objeto de estudio** lo constituyen los procesos de desarrollo de sistemas Web. Partiendo de esto la investigación se centra en el **campo de acción** que esta dado por el análisis y diseño de los portales Web.

Por consecuente el **objetivo general** es realizar el análisis y diseño para desarrollar un portal que permita publicar la información insertada en el Sistema Nacional de Programación Cultural y gestionar la información que no maneja dicho sistema.

En consecuencia surgen las siguientes **preguntas científicas**:

- ¿Qué tecnologías se pueden escoger para crear un sitio Web dinámico?
- ¿Qué tecnologías se usan para el envío de boletines de forma dinámica?
- ¿Qué tecnologías se pueden usar para dar solución al problema de gestionar la información que se encuentra de forma parcial en la base de datos del Sistema Nacional de Programación Cultural?

Del mismo se derivan los siguientes **objetivos específicos**:

- Recopilar las bases teóricas para el correcto desarrollo de un portal.
- Diseñar un módulo de búsqueda que le permita al usuario realizar búsquedas simples con solo insertar en un campo cualquier palabra que se asocie con una actividad cultural o búsquedas avanzadas que lo ayudarán a obtener mejores resultados a partir de una serie de parámetros que debe especificar.
- Diseñar un módulo que gestione la información que se publicará en la página principal. El Webmaster podrá adicionar, modificar o eliminar diferentes clasificaciones de la información como lo son los adelantos, los cambios, las actividades culturales y los banners.
- Diseñar de un módulo que envíe boletines semanales con la información que se publica en el home para que los usuarios estén actualizados sobre las principales actividades culturales que son de su interés sin tener que acceder al sitio.
- Diseñar de un módulo que permita adicionar, modificar y eliminar la información adicional referente a las actividades culturales, el potencial artístico y los espacios culturales, como lo son el nombre, la descripción y una imagen.

Para la realización del sistema se plantearon las siguientes **tareas**:

- Reuniones con el cliente.
- Estudio del Sistema Nacional de Programación Cultural.
- Análisis de las herramientas para el desarrollo de sistemas con estas características, fundamentando con investigación la elección.
- Diseño de un base de datos que satisfaga las condiciones del sistema.
- Diseño de una aplicación Web segura, con interfaz amigable y que este acorde con el fin por el cual fue concebido el portal.

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. Se hace un estudio de las tecnologías candidatas para el desarrollo del trabajo, como el gestor de base de datos, el lenguaje de programación, el gestor de contenido, y la metodología de desarrollo de software a utilizar.

Capítulo2: “Caracterización 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.

Capítulo3: “Análisis y diseño del sistema”, donde se especifican las clases que se utilizan en la solución, la relación entre ellas y se especifican los diagramas de interacción por cada caso de uso.

Capítulo1: Fundamentación teórica

1.1 Introducción

En el presente capítulo se hace un análisis de las tecnologías actuales que se pueden usar como vía de solución del problema. Se tienen en cuenta los lenguajes de programación, los gestores de contenido, la tecnología del lado del cliente así como el Gestor de Base de Datos MySql y los Web Services.

Finalmente se seleccionan los más convenientes que garanticen el cumplimiento de los intereses de los usuarios finales y de CUBARTE.

1.2 Gestor de Base de Datos MySql

Sistema multiplataforma que ha atraído a muchos programadores por su sencillez. Pertenece a Sun Microsystems a partir de enero del 2008 y esta bajo un número de licencias flexibles diseñadas para mantener la usabilidad y distribución del software para los diferentes tipos de usuarios. Es el servidor de BD relacionales más popular desarrollado por la MySql AB, por lo que 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. Es soportado por varios lenguajes de programación como: PHP, Java, Perl, Phython y Ruby on Rails de los que se tiene amplia documentación en el sitio oficial de MySql.

MySql tiene ventajas adicionales como la escalabilidad, mediante la cual es posible manipular bases de datos enormes, del orden de seis mil tablas y alrededor de cincuenta millones de registros, y hasta 32 índices por tabla. Está escrito en C y C++, probado con multitud de compiladores y dispone de APIs para muchas plataformas diferentes. También permite conexiones entre diferentes máquinas con distintos sistemas operativos. Es común que servidores Linux o Unix, usando MySql, sirvan datos para ordenadores con Windows, Linux, Solaris, etc. Para ello se usa TCP/IP, tuberías, o sockets Unix. Es multihilo, con lo que puede beneficiarse de sistemas multiprocesador. Además de que permite manejar multitud de tipos para columnas y registros de longitud fija o variable.

Se selecciona MySQL debido a todas las ventajas anteriormente expuestas y que el sistema del cual se nutrirá el Portal Nacional de Programación Nacional para ofrecer algunos servicios utiliza dicho gestor de Base de Datos.

1.3 Lenguajes de programación

Java

Java fue creado en el 1991 por la Sun Microsystems. El lenguaje de programación Java ha sido refinado, extendido, probado y provee una activa comunidad de sobre los 5 millones de desarrolladores de software.

Su robustez lo ha convertido en un recurso invaluable para el desarrollo, algunas de sus características más codiciadas son:

- Se puede escribir un software en una plataforma y correrlo en prácticamente cualquier otra plataforma.
- Crear programas para correr sin un “Web browser”, ni un “Web services”.
- Desarrollar un servidor para foros online, encuestas, almacenamiento y más.
- Escribir poderosas y eficientes aplicaciones para teléfonos móviles, procesadores remotos, y prácticamente cualquier dispositivo con heartbeat digital.

Java fue creado para abrir una nueva vía en la gestión de software complejo, y es por regla general aceptado que se ha comportado bien en ese aspecto. Sin embargo no puede decirse que Java no tenga grietas, ni que se adapta completamente a todos los estilos de programación, todos los entornos, o todas las necesidades. (Harold 1995)

Practical Extracting and Reporting Language (Perl)

Perl es un lenguaje de propósito general originalmente desarrollado para la manipulación de texto y que ahora es utilizado para un amplio rango de tareas incluyendo administración de sistemas, desarrollo Web, programación en red, desarrollo de GUI y más. 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. Otra característica de este lenguaje lo constituye el hecho de ser un lenguaje interpretado al igual que muchos otros, por lo que el código se lee cada vez que se quiera ejecutar.

Algunas de las desventajas de Perl son la libertad que se le otorga al programador, esto se puede traducir en obtener como resultado un programa ilegible. Si no se escribe con cuidado puede llegar a ser difícil de leer, de hecho hay un concurso de Perl ofuscado al cual se presentaron programadores de Perl durante 4 años consecutivos entre 1996 y el 2000, en este concurso ganaba el programador que lograra la salida de código más ilegible. Otra desventaja es que cada vez que corre un programa debe ser compilado, lo que lo hace más lento en tiempo de ejecución que otros lenguajes aunque se han implementado técnicas para mejorar esta situación.

Phyton

Phyton es comparado habitualmente con TCL, Perl, Scheme, Java y Ruby. En la actualidad se desarrolla como un proyecto de código abierto, administrado por la Python Software Foundation. Dicho lenguaje mantiene una oposición amistosa con Perl ya que los usuarios del primero lo consideran mucho más amplio y elegante para programar. Es un lenguaje multiparadigma (programación orientada a objetos, programación estructurada y programación funcional) 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. Otra característica importante es la resolución dinámica de nombres, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado ligadura dinámica de métodos). 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. Es un lenguaje que cada día está ganando mayor popularidad entre los usuarios, principalmente, por su potencia y su sencillez de trabajo.

Ruby on Rails

Ruby on Rails, también conocido como RoR o Rails es un framework de aplicaciones Web de código abierto escrito en el lenguaje de programación Ruby. Apareció en el año 1995, su principal problema había sido la falta de documentación en otro idioma que no sea japonés. Eso se ha ido solucionando y crece la popularidad del lenguaje. Su aplicación insignia, por decirlo de algún modo parece ser RoR.

Hypertext Preprocessor (PHP)

PHP es un lenguaje multiplataforma. Tiene capacidad de conexión con diferentes manejadores de BD entre los que se destaca su conectividad con MySQL. Presenta capacidad para aumentar su potencial debido a la cantidad de módulos que se le pueden insertar. Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda. Es libre, por lo que se presenta como una alternativa de fácil acceso para todos. Permite las técnicas de Programación Orientada a Objetos (en su última versión PHP.6 trae mejoras en cuanto a este aspecto). Cuenta con una biblioteca nativa de funciones sumamente amplia e incluida. No requiere definición de tipos de variables y tiene manejo de excepciones.

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 potenciando la conexión a servidores de bases de datos soportados por el lenguaje. También es utilizado en la programación de sistemas gestores de contenido muy populares como Joomla, Moodle, Mambo y Drupal.

PHP tiene abstracciones que Python no tiene, como por ejemplo:

- La sintaxis que utiliza, la toma de otros lenguajes muy extendidos como C y Perl,
- Las declaraciones de 'switch' y 'do... while'.
- Operadores de incremento, decremento y asignación (en python solo se usa el de asignación)

Se selecciona PHP por las ventajas anteriormente expuestas: el nivel de integración que tiene con MySQL, ajustarse a los CMS candidatos para la construcción del portal, poseer la capacidad de expandir su potencial utilizando las extensiones, ser multiplataforma y permitir técnicas de programación orientada a objetos.

1.4 Gestores de contenido

Joomla

Es de código abierto construido con PHP bajo la licencia GPL. Este administrador de contenidos se usa para publicar tanto en Internet como en intranets utilizando una base

de datos MySQL. En Joomla se incluyen características como: hacer caché de páginas para mejorar el rendimiento, indexar páginas Web, feed RSS, versiones imprimibles de páginas, flash con noticias, blogs, foros, encuestas, calendarios, búsqueda en el sitio Web, e internacionalización del lenguaje. Ofrece la posibilidad de instalar, desinstalar y administrar componentes y módulos que agregan servicios al sitio Web.

Mambo

Es una aplicación escrita en PHP de código abierto que utiliza como gestor de BD a MySQL, además de ser fácil de instalar y actualizar. Mambo permite la creación y el mantenimiento de sitios Web y portales de manera fácil y dinámica. Permite al administrador de una página Web, de manera sencilla, actualizar la página y hacer accesible a todo tipo de usuarios a través de una variedad de instrumentos. La simplicidad de Mambo radica en que no son necesarios conocimientos técnicos ni especializados para crear, mantener, actualizar o personalizar los contenidos de un sitio Web como HTML o PHP.

Entre sus principales características está el gestor de zonas y secciones, motor de búsqueda integrado, soporte para más de 20 lenguajes y administrador de banners.

Drupal

Dentro de las principales características de dicho CMS se puede decir que se distribuye bajo licencia GNU/GPL por lo que es software libre. Para organizaciones con instalaciones repartidas, Drupal posee un excelente apoyo para usuarios múltiples y un sistema de permisos infinitamente configurable. Presenta una larga lista de módulos dedicados al activismo y control de la mercadotecnia "viral". En cuanto a flexibilidad/configurabilidad a menudo es orientado hacia proyectos, y los proyectos pueden variar enormemente de uno a otro. Su comunidad de usuarios viene siendo adoptada por un considerable número de grandes instituciones y organizaciones globales. Dado que estas organizaciones perfeccionan Drupal para reflejar sus necesidades y valores, la herramienta se hace cada vez más pertinente y atractiva para otros, resultando de esto un bucle de retroalimentación positiva. En cuanto al lenguaje e idioma se ha visto una mejora significativa en su apoyo para el contenido multilingüe.

Una de las principales ventajas de Drupal y otra razón para seleccionarlo, es la numerosa documentación que podemos encontrar en Internet así como la facilidad de creación de módulos nuevos, además de utilizar PHP y MySQL. En el año 2007 Drupal fue ganador de una competencia de 3 meses de votos por el mejor CMS de código abierto.

1.5 Tecnologías del lado del cliente

JavaScript

Es un lenguaje interpretado basado en prototipos que puede ser utilizado por profesionales, no requiere de compilación pues se ejecuta en los navegadores del cliente. Muchos confunden el java script con el java aunque son lenguajes diferentes, con sus peculiaridades. Tiene grandes posibilidades ya que es utilizado para desarrollar programas pequeños que luego se insertan en las aplicaciones Web. JavaScript es soportado por la mayoría de los navegadores como IE, Mozilla Firefox, Opera, entre otros. A la hora de darle dinamismo a nuestra Web es muy usado, 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.

Existen dos tipos de JavaScript: por un lado está el que se ejecuta en el cliente, este es el Javascript propiamente dicho, aunque técnicamente se denomina Navigator JavaScript. Pero también existe un Javascript que se ejecuta en el servidor, se denomina LiveWire Javascript y es más reciente.

Cascade Style Sheets (CSS)

Las hojas de estilo en cascada describen como se va a mostrar un documento, se encuentran generalmente separados del código principal permitiendo darle estilos a los documentos HTML y XML por lo que facilita que en un equipo de trabajo el diseñador y el programador puedan hacer sus trabajos de forma independiente, sin correr el riesgo de que haya interferencia entre ambos. Mediante CSS es posible separar el contenido de la presentación, lo que posibilita que se pueda simplificar y limpiar el HTML de los documentos, haciendo éstos más accesibles. Gozan de gran flexibilidad pues se pueden cambiar en cualquier momento el diseño de la página sin que esto afecte el contenido.

Con su ayuda se reducen notablemente los tiempos de carga del sitio en el navegador, a esto le podemos unir la capacidad de éste para mantener nuestra hoja de estilo en caché.

Hyper Text Markup Language (HTML)

HTML es el lenguaje de marcado predominante en el desarrollo de páginas Web, describe la estructura y el contenido en forma de texto. 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, además nos permite la introducción de referencias a otras páginas por medio de los enlaces hipertexto.

Extensible Markup Language XML (XML)

XML es un metalenguaje. 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.

1.6 Metodologías de desarrollo de software

Para dar una idea de qué metodología se puede utilizar y cual se adapta más a nuestro medio, mencionaré tres de las más usadas, tal como: XP, MSF y RUP.

Extreme Programing (XP)

Es una de las metodologías ágiles de desarrollo de software más exitosa. En la actualidad es utilizada para proyectos cuyo plazo de entrega era ayer o con un escaso equipo. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

XP se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores, en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio. También se basa en la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma PC o puesto de trabajo.

Dicha metodología propone empezar con algo pequeño y añadir funcionalidades con retroalimentación continua; no introducir funcionalidades antes que sean necesarias; que el cliente o el usuario se convierten en miembro del equipo; decidir que se implementa; saber el estado real y el progreso del proyecto; añadir, cambiar o quitar requerimientos en cualquier momento; obtener lo máximo de cada semana de trabajo; obtener un sistema funcionando cada 3 o 4 meses; decidir como se implementan los procesos; crear el sistema con la mejor calidad posible; pedir al cliente en cualquier momento aclaraciones de los requerimientos; cambiar los requerimientos en base a nuevos descubrimientos.

A modo de conclusión se podría decir que lo fundamental en XP es la comunicación entre los usuarios y los desarrolladores; la simplicidad, al desarrollar y codificar los módulos del sistema y por último la retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

Microsoft Solution Framework (MSF)

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. Es propiedad de Microsoft por lo que es un software privativo y comercial.

MSF es adaptable pues se compara con un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar. Escalable, ya que puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más. Flexible porque es utilizada en el ambiente de desarrollo de cualquier cliente. Es de tecnología agnóstica porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

Esta metodología de desarrollo se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.

Modelo de Arquitectura del Proyecto: Diseñado para acortar la planificación del ciclo de vida. Este modelo define las pautas para construir proyectos empresariales a través del lanzamiento de versiones.

Modelo de Equipo: Este modelo ha sido diseñado para mejorar el rendimiento del equipo de desarrollo. Proporciona una estructura flexible para organizar los equipos de un proyecto. Puede ser escalado dependiendo del tamaño del proyecto y del equipo de personas disponibles.

Modelo de Proceso: Diseñado para mejorar el control del proyecto, minimizando el riesgo, y aumentar la calidad acortando el tiempo de entrega. Proporciona una estructura de pautas a seguir en el ciclo de vida del proyecto, describiendo las fases, las actividades, la liberación de versiones y explicando su relación con el Modelo de equipo.

Modelo de Gestión del Riesgo: Diseñado para ayudar al equipo a identificar las prioridades, tomar las decisiones estratégicas correctas y controlar las emergencias que puedan surgir. Este modelo proporciona un entorno estructurado para la toma de decisiones y acciones valorando los riesgos que puedan provocar.

Modelo de Diseño del Proceso: Diseñado para distinguir entre los objetivos empresariales y las necesidades del usuario. Proporciona un modelo centrado en el usuario para obtener un diseño eficiente y flexible a través de un enfoque iterativo. Las fases de diseño conceptual, lógico y físico proveen tres perspectivas diferentes para los tres tipos de roles: los usuarios, el equipo y los desarrolladores.

Modelo de Aplicación: Diseñado para mejorar el desarrollo, el mantenimiento y el soporte, proporciona un modelo de tres niveles para diseñar y desarrollar aplicaciones software. Los servicios utilizados en este modelo son escalables, y pueden ser usados en un solo ordenador o incluso en varios servidores.

Rational Unified Process (RUP)

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en 4 fases el desarrollo del software:

Inicio: el objetivo en esta etapa es determinar la visión del proyecto.

Elaboración: en esta etapa el objetivo es determinar la arquitectura óptima.

Construcción: en esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.

Transición: el objetivo es llegar a obtener el realce del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, el cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. El ciclo de vida que se desarrolla por cada iteración, es llevado bajo dos disciplinas:

Disciplina de Desarrollo: En esta disciplina se toca la Ingeniería de Negocios (entendiendo las necesidades del negocio) los requerimientos en donde se trasladan las necesidades del negocio a un sistema automatizado, el análisis y diseño (se trasladan los requerimientos dentro de la arquitectura de software), la implementación donde se crea un software que se ajuste a la arquitectura y que tenga el comportamiento deseado y por último las pruebas asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado esta presente.

Disciplina de Soporte: aquí se toca la configuración y administración del cambio, guardando todas las versiones del proyecto, se administran las horas y recursos del proyecto, el ambiente de desarrollo y la distribución, haciendo todo lo necesario para la salida del proyecto.

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierte luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

Por lo anterior expuesto se propone utilizar RUP como metodología ya que en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las

metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. Además de que asegura la calidad del software, permite rastrear los cambios y se adapta a todo tipo de proyectos. La elaboración de distintos diagramas y artefactos siguiendo esta metodología proveen una fácil ejecución del proceso de elaboración de un Sistema de Software, ya que describen cómo está estructurado el sistema desde diferentes perspectivas orientadas a los diferentes involucrados en un proyecto. Nos proporciona una guía de cómo utilizar de forma efectiva el UML. Es un proceso configurable ya que no estamos obligados a hacer uso de todas las actividades y entregables definidos, sino que podemos configurar nuestro proceso para adaptarnos únicamente a aquellas partes que se consideran necesarias. Craig Larman es un fiel defensor de utilizar RUP como una metodología ágil y para ello a escrito libros como “*Applying UML and Patterns*” que contiene un proceso que esta muy basado en su pensamiento ligero de RUP.

1.7 Herramientas CASE

1.7.1 Umbrello

Umbrello UML Modeller es una herramienta libre de diagramas que ayuda en el proceso del desarrollo de software. Facilita la creación de un producto, especialmente durante las fases de análisis y diseño del proyecto. Puede usarse para documentar los diseños de software. Soporta diferentes tipos de diagramas como son: diagrama de clase, diagrama de secuencia, diagrama de colaboración, diagrama de caso de uso, diagrama de estado, diagrama de actividad, diagrama de colaboración y diagrama de secuencia. Permite la generación de código fuente en diferentes lenguajes de programación como ActionScript, Ada, C++, CORBA IDL, Java™, JavaScript, PHP, Perl, Python, SQL y XMLSchema.

1.7.2 Visual Paradigm

Visual Paradigm es una herramienta que sirve para el lenguaje de modelado UML y que soporta el ciclo de vida completo de desarrollo de RUP (análisis y diseño orientados a objetos, construcción, pruebas y despliegue). Tiene varias funcionalidades, permite dibujar todos los tipos de diagramas de clases, código inverso, generar código PHP desde diagramas y generar documentación. Actualmente se puede instalar tanto en el SO Linux como Windows. Esta herramienta tiene disponible diferentes versiones

comerciales como Enterprise, Professional, Standard, Modeler, Personal y una gratuita llamada Visual Paradigm UML Community. Facilita licencias especiales para fines académicos. Importa al Racional Rose, esta característica lo hace muy útil por la diversidad de proyectos que son modelados en el Racional.

Se selecciona esta herramienta por su creciente popularidad y debido a que cubre casi todo el ciclo de vida de RUP. Permite modelar bases de datos, presenta gran interoperabilidad, puede generar código fuente en más de 10 lenguajes de programación y documentación en diferentes formatos como PDF, HTML y MS Word.

1.8 ¿Qué es un portal?

Consolidan una gran cantidad de información obtenida de diversos lugares en una sola aplicación, facilitando el acceso a una serie de recursos y servicios que están dirigidos principalmente a resolver necesidades específicas de un grupo de personas o de acceso a la información y servicios de una institución pública. Los portales tienen varias denominaciones, se pueden clasificar en las modalidades de horizontales si son masivos o de propósito general, o verticales si solo ofrecen contenido sobre un tema en específico así como los hay de información o de gestión de contenido aunque generalmente se combinan ambos para dar como resultado un producto que cumpla con todas las necesidades.

Luego del análisis llevado a cabo se propone el desarrollo de un Portal Web utilizando a Drupal como CMS debido al crecimiento de su potencial, como lenguaje de programación del lado del servidor PHP, dada su portabilidad, eficiencia, ser fácil de aprender, sus extensiones y funciones MySQL para la conexión a la BD. De el lado del cliente JavaScript para lograr una interactividad con el usuario en el navegador, además CSS y HTML para lograr el formato requerido en la interfaz de usuario. Se propone también la utilización de MySQL como SGBD por ser potente y gozar de gran compatibilidad con PHP y finalmente como metodología a utilizar se propone RUP pues se adapta a cualquier proyecto de software ya que es configurable.

Capítulo2: Características del Sistema

2.1 Introducción

El presente capítulo da paso a la descripción de la solución propuesta para el sistema. Se describen los procesos del negocio y se realiza, debido a las características del sitio, un modelo del dominio, enunciando los conceptos del mismo. Se enumeran los requisitos funcionales y no funcionales que debe tener el sistema permitiendo dar una concepción general del sitio, e identificar mediante un diagrama de casos de usos los actores que interactúan con el portal.

2.2 Descripción de los Procesos del negocio propuestos

Dentro del modelado del negocio el primer paso es la identificación de los diferentes procesos del negocio de la organización que en este caso es el Ministerio de Cultura (MINCULT). Para describir los procesos del negocio es necesario centrarnos en la publicación de carteleras culturales y en los servicios que brindará el Portal Nacional de Programación Cultural llamado “La Papeleta”.

El proceso de publicación de carteleras culturales

El proceso de publicación de carteleras culturales comienza cuando se programan las actividades por parte de los programadores culturales y estas son enviadas a las diferentes instituciones de Cuba para ser difundidas a través de diferentes vías (TV, radio, Web, carteles, etc.) y garantizar así la asistencia del público.

El proceso de los diferentes servicios que brinda el portal

El proceso comienza cuando un usuario desea asistir a alguna actividad cultural y para ello debe informarse de cuales están programadas. El portal brinda varios servicios entre ellos se destacan la búsqueda simple o avanzada de actividades culturales además de una búsqueda por fecha. Otro servicio es el de un boletín informativo al cual el usuario accede mediante la suscripción al mismo.

2.3 Modelo del dominio

Después de haber estudiado los procesos de negocio descritos anteriormente y de arribar a la conclusión de que el negocio posee un bajo nivel de estructuración, se utiliza el modelo de dominio con el objetivo de representar las clases conceptuales o

entidades del mundo real. Esto ayuda a los clientes y desarrolladores a utilizar un lenguaje común para poder entender el contexto en que se emplaza el sistema.

Para un mayor entendimiento del diagrama del dominio se exponen los siguientes conceptos:

MINCULT: Ministerio de Cultura de Cuba, es el organismo que atiende todo lo que tiene que ver con el quehacer cultural de nuestro país.

Sistema nacional de programación cultural: Sistema creado para gestionar a nivel nacional la programación cultural.

Programadores culturales: Son los encargados de insertar la programación de las actividades culturales en el sistema nacional de programación cultural.

Portal “La papeleta”: Sitio Web que contiene información referente a la programación cultural, extraída en gran parte del Sistema Nacional de Programación Cultural.

Webmaster: Es el administrador del Portal.

Usuario: Es el internauta que navega por el portal.

Potencial artístico: Es el artista que participa en una actividad cultural.

Espacio cultural: Es el lugar donde se va a desarrollar la actividad.

2.3.1 Definición de los roles del dominio

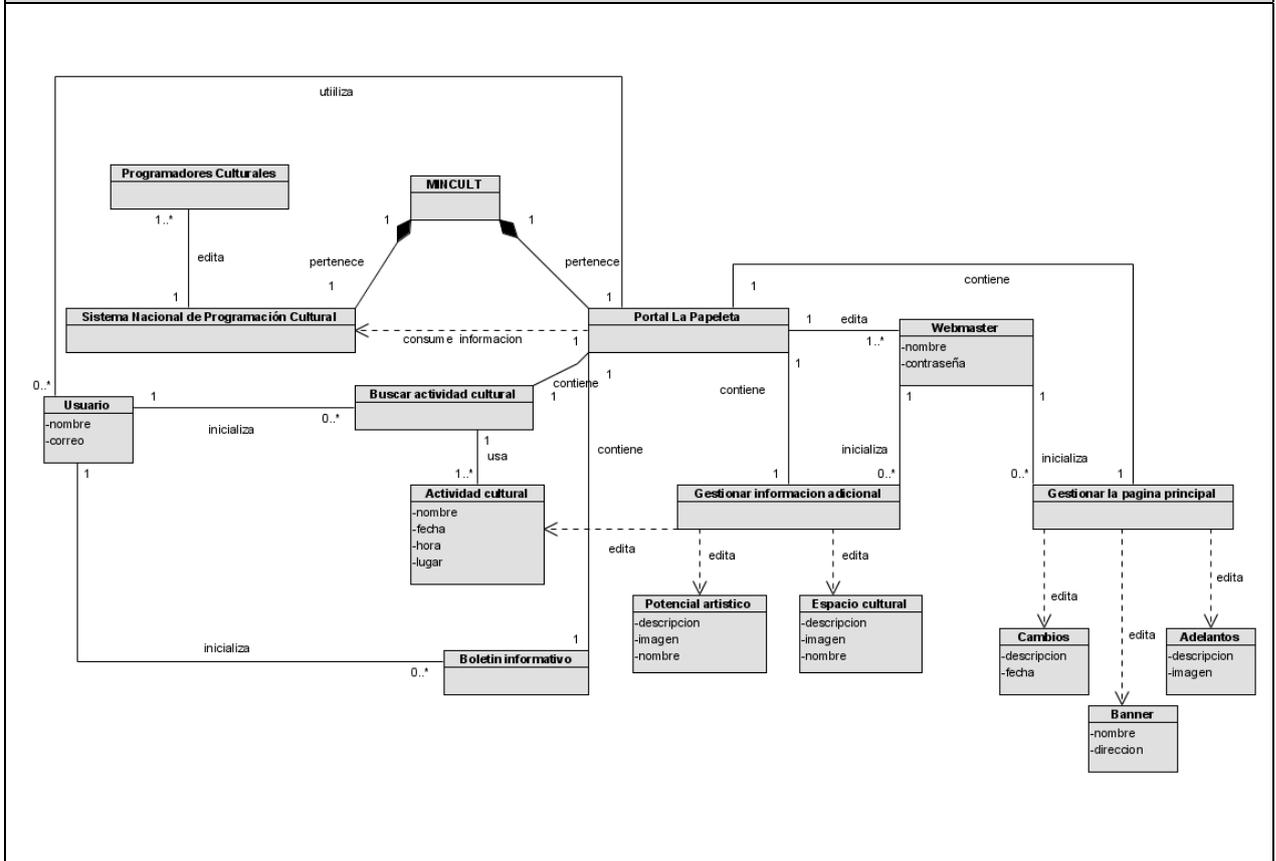
Webmaster: El Webmaster puede desempeñar el rol del editor aunque también es el encargado de adicionar, eliminar, o editar la información que se va a mostrar en el portal. Su principal función es mantener el sitio actualizado.

Usuario: Representa al usuario común que navega por el portal y solo tiene permisos para consultar la información, así como para registrarse al boletín informativo.

2.3.2 Diagrama del modelo del dominio

El modelo del dominio se describe mediante un diagrama UML, específicamente con un diagrama de clases conceptuales significativas en el dominio del problema.

Diagrama del modelo de dominio



2.4 Requisitos

2.4.1 Requisitos Funcionales

R1 Mostrar actividades culturales en cartelera.

Se refiere a:

- Nombre de la actividad.
- Nombre del artista o agrupación.
- Breve descripción.
- Imagen.

R2 Escoger las actividades que se van a publicar.

Se muestran las actividades programadas para la semana y el administrador escoge las que se van a promocionar, no deben exceder de 20.

R3 Mostrar las actividades menos relevantes en cartelera.

La diferencia entre estas y las primeras es que no incluye descripción ni imagen. Se refiere a:

- Nombre de la actividad.
- Nombre del artista o agrupación.

R4 Escoger las actividades menos relevantes a promocionar en el home.

Se muestran las actividades programadas para la semana y el administrador escoge las que se van a promocionar, no deben exceder de 10.

R5 Gestionar información adicional de actividades culturales.

R5.1 Adicionar la información de actividades culturales.

El parámetro "Nombre" se debe seleccionar de un menú desplegable. Se obtienen dichos nombres, de las actividades programadas en la semana en curso.

Los parámetros que se van a adicionar son:

- Descripción de la actividad.
- Imagen.
- Nombre.

R5.2 Modificar la información de actividades culturales.

R5.3 Eliminar la información de actividades culturales.

R6 Gestionar los adelantos:

R6.1 Adicionar adelantos.

Los parámetros requeridos son:

- Descripción.
- Imagen.

R6.2 Modificar los adelantos.

R6.3 Eliminar los adelantos

R7 Gestionar los cambios de la programación cultural

R7.1 Adicionar cambios de la programación cultural.

Los parámetros requeridos son:

- Descripción del cambio.
- Imagen

R7.2 Modificar los cambios de la programación cultural.

R7.3 Eliminar los cambios de la programación cultural.

R8: Realizar búsquedas.

Los tipos de búsqueda deben ser:

R8.1 Búsquedas simples.

R8.2 Búsquedas avanzadas.

Los parámetros para este tipo de búsqueda son:

Datos básicos:

- Artista o Agrupación.
- Lugar o Institución.

Fecha:

- Fecha de inicio de búsqueda.
- Fecha de fin de búsqueda.
- Por defecto buscar desde fecha actual.

Manifestaciones:

- Nombre de manifestación.
- Provincia / Municipio.

Buscar solamente por:

- Adultos.
- Infantiles.

R9 Mostrar resultados de búsqueda.

Se refiere a:

- Nombre de la actividad.
- Marco en el que se realiza.
- Manifestación artística o cultural a la que pertenece.
- Lugar.
- Fecha.
- Hora.
- Nombre del artista.
- Estatus laboral del artista.
- Provincia.
- Municipio.

R9.1 Imprimir resultados de la búsqueda.

Consiste en:

- Opción de imprimir.
- Imprimir solo las tablas resultantes.

R11 Inscribir a boletín:

EL usuario recibirá semanalmente una cartelera de las principales actividades culturales que están programadas para la semana en curso.

R11.1 Dar alta:

Se deben insertar los siguientes parámetros:

- Nombre.
- Correo electrónico.

- Manifestación
- Provincia y municipio.

R11.2 Dar baja.

Se debe insertar el correo electrónico.

R12 Mostrar vínculos con sitios de carteleras para el turismo internacional.

R13 Gestionar banners.

Se deben actualizar automáticamente cada cierto tiempo.

R13.1 Adicionar banner.

R13.2 Eliminar banner.

R14 Gestionar información adicional de potencial artístico.

Se gestiona la información que no se encontraba en la BD del sistema de programación y que es necesaria para el portal.

R14.1 Adicionar; los parámetros a tener en cuenta son:

- Nombre.
- Descripción.
- Imagen.

R14.2 Modificar.

R14.3 Eliminar.

R15 Gestionar información adicional de espacios culturales.

Se gestiona la información que no se encontraba en la BD del sistema de programación y que es necesaria para el portal.

R15.1 Adicionar, los parámetros a tener en cuenta son:

- Nombre.
- Descripción.
- Imagen.

R15.2 Modificar.

R15.3 Eliminar.

R16 Mostrar descripción

Se refiere a tener vínculos que ayuden al usuario a obtener la descripción de un artista o agrupación así como la de un espacio cultural.

2.4.2 Requisitos no Funcionales

Los requisitos no funcionales son propiedades o cualidades que el sistema debe tener.

Apariencia o interfaz externa

- Diseño sencillo, con colores agradables y que atraigan al usuario permitiéndole así navegar sin grandes complicaciones y que encuentre fácilmente las principales opciones.
- Diseño creado para que pueda verse en monitores de cualquier resolución de pantalla, dígase resoluciones que abarquen desde 800x600 hasta 1024x768.

Rendimiento

- Procesamiento de datos y tiempo de respuesta rápidos. El sistema debe ser eficiente para lograr los resultados de las búsquedas en el menor tiempo posible.

Soporte

- Garantía de instalación y prueba del sistema.
- Adiestrar al cliente en la administración del portal.

Portabilidad

- Debe ser multiplataforma.
- Debe poderse acceder al sitio desde los navegadores más comunes.

Seguridad

- Garantizar que las funcionalidades del sistema se muestren de acuerdo al usuario que esté activo.
- Confirmar las acciones que no se puedan deshacer (ejemplo: eliminar).

Confiabilidad

- Recibir datos introducidos por los usuarios correctamente validados.
- Garantizar un tratamiento de excepciones cuyos mensajes no comprometan información del servidor y sean lo más concretos posible.
- Garantizar que se actualice o adicione información al portal sin la necesidad de detener el servicio que se le brinda a la comunidad.

2.5 Descripción del Sistema

El Portal será desarrollado como un módulo más del CMS Drupal. Teniendo en cuenta los requisitos el mismo estará compuesto por 2 secciones o bloques fundamentales, uno para el usuario común que navega por el sitio y otro para el administrador; el primero estará compuesto por la zona del buscador, mediante la cual podrá hacer tanto búsquedas simples como avanzadas, esta última incluye las opciones de buscar por fechas, por manifestación o provincia, además podrá acceder a las noticias culturales, el boletín e información adicional, referente a las actividades culturales y el potencial artístico. La sesión de administrador estará compuesta por un módulo para gestionar los adelantos, los banners, la información adicional de las actividades culturales, los espacios culturales y el potencial artístico además del boletín informativo.

2.6 Modelo de casos de uso del sistema

Se representan los requisitos funcionales del sistema, mediante un diagrama de casos de uso con la ayuda de UML. De acuerdo con lo anterior planteado se definen los siguientes actores que interactúan con el portal y los casos de usos.

Actores	Descripción
Webmaster	Representa a la persona que va a administrar el sitio para que el mismo y sus servicios se mantengan siempre actualizados.
Usuario	Representa a una persona que va a acceder al sitio y utilizar sus servicios

2.6.1 Descripción de los casos de uso del sistema

CU-1	Buscar actividades culturales
Actor	Usuario
Descripción	Mediante este CU el usuario puede solicitar el tipo de búsqueda que desea realizar sobre cualquier actividad cultural.
Referencia	R8, R9

CU-2	Inscribir en boletín informativo
Actor	Usuario
Descripción	El usuario puede solicitar que se le envíe semanalmente un boletín sobre las actividades culturales planificadas para esos días.
Referencia	R11

CU-3	Mostrar descripción
Actor	Usuario
Descripción	Mediante este CU el usuario puede conocer datos adicionales de sus artistas favoritos así como espacios culturales de interés.
Referencia	R16

Capítulo 2: Características del sistema

CU-4	Gestionar actividades culturales promovidas al home.
Actor	Webmaster
Descripción	Mediante este CU el administrador del portal puede adicionar o eliminar las actividades culturales que van en la página principal del sitio.
Referencia	R1,R2,R3,R4,R12

CU-5	Gestionar la información adicional de las actividades culturales
Actor	Webmaster
Descripción	Consiste en que al Administrador se le permita editar la información adicional que va a mostrar el portal, referente a las actividades culturales
Referencia	R5

CU-6	Gestionar la información adicional de los espacios culturales
Actor	Webmaster
Descripción	El Administrador puede editar la información adicional que va a mostrar el portal referente a los espacios culturales
Referencia	R15

Capítulo 2: Características del sistema

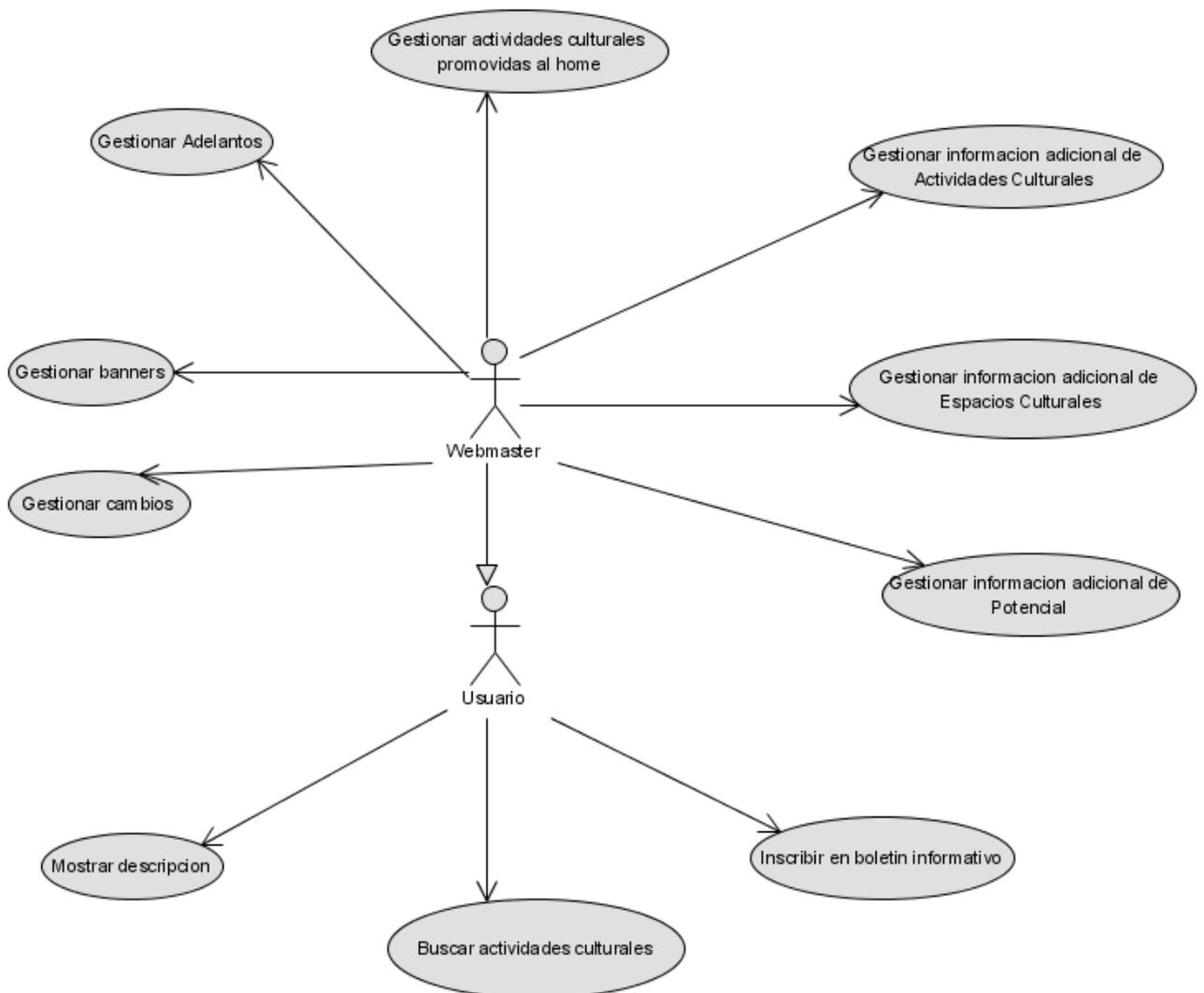
CU-7	Gestionar la información adicional del potencial artístico.
Actor	Webmaster
Descripción	Al Administrador se le da la opción de editar la información adicional que va a mostrar el portal referente al potencial artístico.
Referencia	R14

CU-8	Gestionar adelantos
Actor	Webmaster
Descripción	Al Administrador se le da la opción de adicionar o eliminar los adelantos que se van a publicar en el home del sitio
Referencia	R6

CU-9	Gestionar cambios de la programación cultural.
Actor	Webmaster
Descripción	Al Administrador del portal se le da la opción de editar los cambios de la programación cultural que van a ser publicados en la página principal.
Referencia	R7

CU-10	Gestionar banners
Actor	Webmaster
Descripción	Al Administrador se le da la opción de adicionar o eliminar los banners que van a ser publicados en el home del portal.
Referencia	R13

2.6.2 Diagrama de los casos de uso del sistema



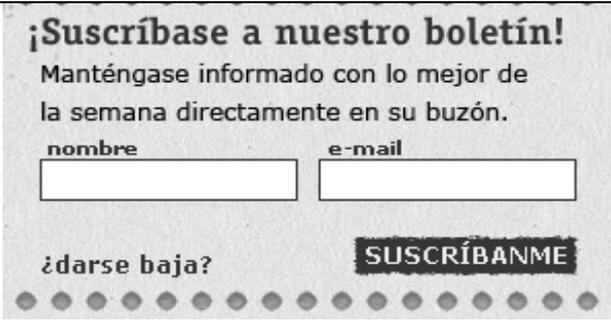
2.6.3 Casos de uso expandidos

Caso de Uso:	Buscar Actividades Culturales	
Propósito:	Buscar las actividades culturales registradas en la BD	
Actor:	Usuario	
Resumen:	El caso de uso se inicia cuando el usuario inserta los datos relacionados con la actividad cultural que desea buscar, la misma puede ser simple o avanzada, esta última se realizará bajo determinados parámetros.	
Referencias	R8, R9	
Flujo Normal de Eventos		
Sección “Búsqueda Avanzada”		
Acción del Actor	Respuesta del Sistema	
1. El usuario introduce los datos que desea buscar en correspondencia con los parámetros del formulario.	2. El sistema busca en la BD según los parámetros solicitados.	
	3. Muestra los resultados encontrados en una nueva página.	
Prototipo de Interfaz		

Sección “Búsqueda Simple”	
Acción del Actor	Respuesta del Sistema
1. El usuario llena el campo de búsqueda simple.	2. El sistema busca en la BD.
	3. Muestra los resultados encontrados en una nueva página.
Prototipo de Interfaz	
	
Poscondiciones	La búsqueda de actividades culturales se ha realizado de manera exitosa
Sección “Vista de impresión”	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción de imprimir	2. Muestra ventana de impresión
3. Configura el tipo de impresión que desea.	5. Imprime
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4. El sistema comprueba la existencia de impresora.
Poscondiciones	Se imprimió la búsqueda satisfactoriamente.

Capítulo 2: Características del sistema

Caso de Uso:	Inscribir en boletín informativo	
Propósito:	Mantener al usuario informado vía e-mail sobre las principales actividades culturales planificadas para la semana.	
Actor:	Usuario	
Resumen:	El CU se inicia cuando el usuario se suscribe al boletín y selecciona la manifestación que prefiere junto con la provincia a la que pertenece.	
Referencias	R11	
Flujo Normal de Eventos		
Sección “Dar alta”		
Acción del Actor	Respuesta del Sistema	
1. Introduce los datos en los campos nombre y e-mail. 2. Selecciona la opción de registrarse.	3. El sistema lo redirecciona a la página de suscripción.	
4. EL usuario introduce los parámetros necesarios para recibir el boletín como la provincia y la manifestación	5. El sistema registra al usuario en la BD.	
	6. El sistema le envía un boletín semanal.	
Prototipo de Interfaz		

	
Poscondiciones	Se le da de alta al usuario satisfactoriamente
Sección "Dar baja"	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción de darse baja del boletín.	2. El sistema lo redirecciona a un formulario.
3. El usuario introduce su correo electrónico.	4. El sistema lo elimina de la lista de contactos.
Poscondiciones	Se le da de baja al usuario satisfactoriamente.
Sección "Modificar"	
Acción del Actor	Respuesta del sistema
1. El usuario selecciona la opción de modificar.	2. El sistema lo envía a un formulario donde podrá introducir su correo electrónico.
3. El usuario introduce la información requerida	4. El sistema genera un nuevo formulario donde podrá cambiar los datos.
5. El usuario cambia la información y oprime el botón actualizar	6. El sistema actualiza la información en la BD.
Poscondiciones	Se modifica la información de manera satisfactoria

Capítulo 2: Características del sistema

Caso de Uso:	Mostrar descripción	
Propósito:	Facilitar al usuario el proceso de obtener la descripción de un espacio cultural especificado o sobre algún artista.	
Actor:	Usuario	
Resumen:	El CU se inicia cuando el usuario solicita el servicio dándole click a cualquier nombre de espacio cultural, artista o agrupación.	
Referencias	R16	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El usuario da clic sobre algún vínculo.	2. El sistema selecciona el tipo de búsqueda y se conecta a la BD. 3. Genera una nueva página en la que el actor obtiene la información solicitada	
Poscondiciones	Se mostró correctamente la descripción.	

Caso de Uso:	Gestionar actividades culturales promovidas al home.	
Propósito:	Administrar las actividades culturales que se van a promover a la página principal.	
Actor:	Administrador	
Resumen:	El CU se inicia cuando el administrador selecciona la opción de administrar la página principal. Para mostrar las actividades en el	

Capítulo 2: Características del sistema

	home debe seleccionarlas o deseccionarlas en caso contrario.
Referencias	R1,R2,R3,R4,R12
Flujo Normal de Eventos	
Sección “”	
Acción del Actor	Respuesta del Sistema
1. El administrador selecciona la opción de actualizar la página inicio	2. El sistema actualiza el listado de actividades culturales para la semana. 3. Muestra el listado de actividades culturales
4. Selecciona las actividades que desea promocionar. 5. Oprime el botón actualizar	
Poscondiciones	Mostrar información en la página principal.

Caso de Uso:	Gestionar información adicional de las actividades culturales
Propósito:	Adicionar información adicional que no se encuentra en la BD del sistema nacional de programación cultural como lo son las imágenes y la descripción
Actor:	Administrador
Resumen:	El CU se inicia cuando el actor solicita adicionar, modificar o eliminar la información adicional referente a las actividades culturales
Referencias	R5

Capítulo 2: Características del sistema

Flujo Normal de Eventos	
Sección “Adicionar”	
Acción del Actor	Respuesta del Sistema
1. El actor escoge la opción de adicionar nueva información.	2. El sistema lo envía a una nueva página donde podrá llenar los datos deseados a través de un formulario
3. El actor llena el formulario y oprime el botón de adicionar.	4 El sistema adiciona los nuevos datos a la BD.
Poscondiciones	Mostrar mensaje de confirmación.
Sección “Modificar”	
Acción del Actor	Respuesta del Sistema
1. El actor escoge la opción de modificar la información almacenada de las actividades culturales.	2. El sistema lo envía a una nueva página donde se encuentran las actividades que existen en la BD.
3.El actor selecciona la que desea modificar	4. El sistema lo envía a un formulario donde podrá cambiar la información.
5. El actor modifica la información que desee y oprime el botón actualizar.	6. El sistema almacena los cambios en la BD reescribiendo los datos antiguos.
Poscondiciones	Mostrar mensaje de confirmación.
Sección “Eliminar”	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona la opción de eliminar la información almacenada de las	2. El sistema lo envía a una nueva página donde se encuentran las actividades que

Capítulo 2: Características del sistema

actividades culturales.	existen en la BD
3. El actor selecciona la que desea eliminar y oprime el botón eliminar.	4. El sistema elimina la actividad cultural de la BD.
Poscondiciones	Mostrar mensaje de confirmación.

CU Gestionar información adicional de los espacios culturales: El contenido de este caso de uso es muy similar al anterior (Gestionar información adicional de las actividades culturales) por tanto se omite.

CU Gestionar información adicional del potencial artístico: El contenido de este caso de uso es muy similar al anterior (Gestionar información adicional de las actividades culturales) por tanto se omite.

Caso de Uso:	Gestionar adelantos
Propósito:	Gestionar el contenido que se sujete a esta denominación como lo son, por ejemplo, los estrenos.
Actor:	Administrador
Resumen:	El CU se inicia cuando el administrador solicita la gestión de un adelanto (adicionar, modificar, eliminar).
Referencias	R6
Flujo Normal de Eventos	
Sección “Adicionar”	
Acción del Actor	Respuesta del Sistema
1. El actor escoge la opción de adicionar nueva información.	2. El sistema lo envía a una nueva página donde podrá llenar los datos

Capítulo 2: Características del sistema

	deseados a través de un formulario
3. El actor llena el formulario y oprime el botón de adicionar.	4 El sistema adiciona los nuevos datos a la BD. 5 Los promueve a la página principal.
Poscondiciones	Mostrar mensaje de confirmación.
Sección “Modificar”	
Acción del Actor	Respuesta del Sistema
1. El actor escoge la opción de modificar la información almacenada de los adelantos	2. El sistema lo envía a una nueva página donde se encuentran las actividades que existen en la BD.
3.El actor selecciona la que desea modificar	4. El sistema lo envía a un formulario donde podrá cambiar la información.
5. El actor modifica la información que desee y oprime el botón actualizar.	6. El sistema almacena los cambios en la BD reescribiendo los datos antiguos.
Poscondiciones	Mostrar mensaje de confirmación.
Sección “Eliminar”	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona la opción de eliminar la información almacenada de las actividades culturales.	2. El sistema lo envía a una nueva página donde se encuentran las actividades que existen en la BD
3. El actor selecciona la que desea eliminar y oprime el botón eliminar.	4 El sistema elimina la actividad cultural de la BD.
Poscondiciones	Mostrar mensaje de confirmación.

Capítulo 2: Características del sistema

CU Gestionar Cambios: El contenido de este caso de uso es muy similar al anterior (Gestionar adelantos) por tanto se omite.

Caso de Uso:	Gestionar banners	
Propósito:	Adicionar o eliminar los banners que se van a publicar en el home del portal.	
Actor:	Administrador	
Resumen:	El CU se inicia cuando el administrador solicita la gestión de un banner.	
Referencias	R13	
Flujo Normal de Eventos		
Sección "Adicionar"		
Acción del Actor	Respuesta del Sistema	
1. El actor escoge la opción de adicionar un nuevo banner	2. El sistema lo envía a una nueva página donde podrá subir el banner.	
3. El actor selecciona la opción de subir. 5. Selecciona la dirección donde se encuentra el banner	4. Se muestra un explorador. 6. El sistema adiciona el nuevo banner. 7. Los promueve a la página principal.	
Poscondiciones	Mostrar en la página principal.	

Capítulo 2: Características del sistema

Sección "Eliminar"	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona la opción de eliminar algún banner	2. El sistema lo envía a una nueva página donde se encuentran los banners adicionados.
3. El actor selecciona el que desea eliminar y oprime el botón eliminar.	4. El sistema elimina el banner.
Poscondiciones	Mostrar mensaje de confirmación.

En este capítulo se creó la propuesta de la solución que se obtuvo a partir del análisis de los procesos del negocio, determinando los roles y el diagrama de dominio. Se obtuvo el modelo de CU del sistema de forma que satisficieran los requisitos, se definieron los actores y los requisitos funcionales y no funcionales del Portal que se pretende construir.

Capítulo 3: Análisis y diseño del sistema

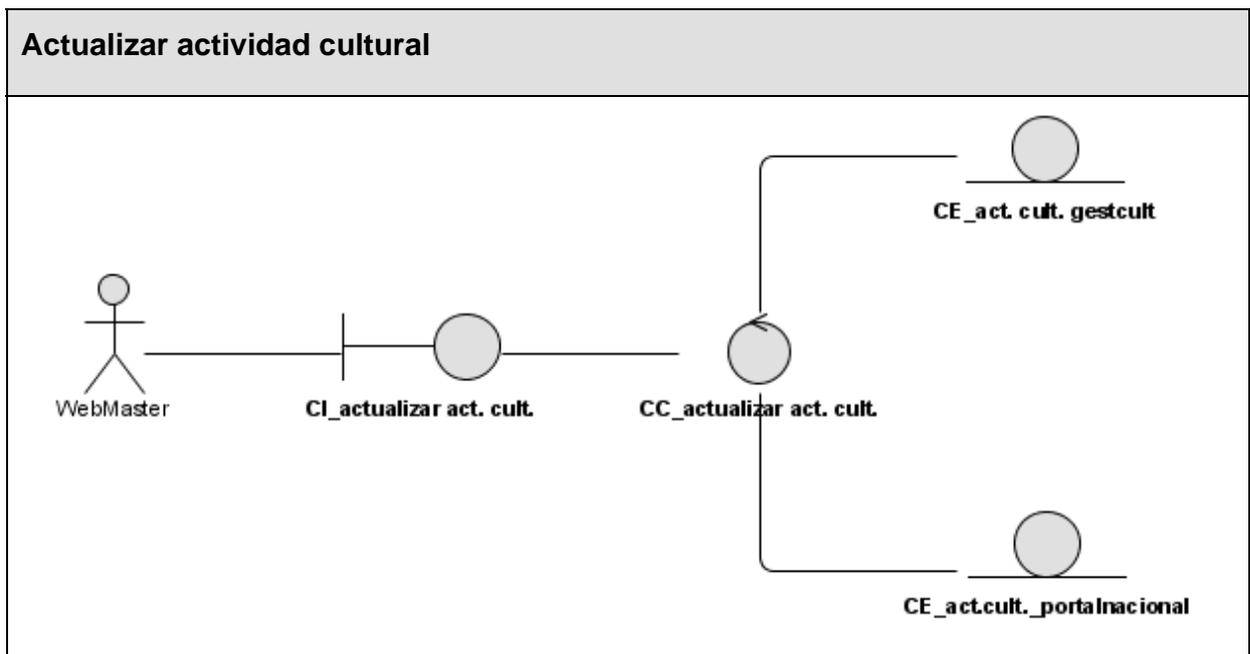
3.1 Introducción.

En este capítulo se comienza a construir la solución del capítulo anterior y se realizan los diagramas de clases de análisis y del diseño, el diagrama de colaboración y el diagrama de clases persistentes así como el modelo de datos.

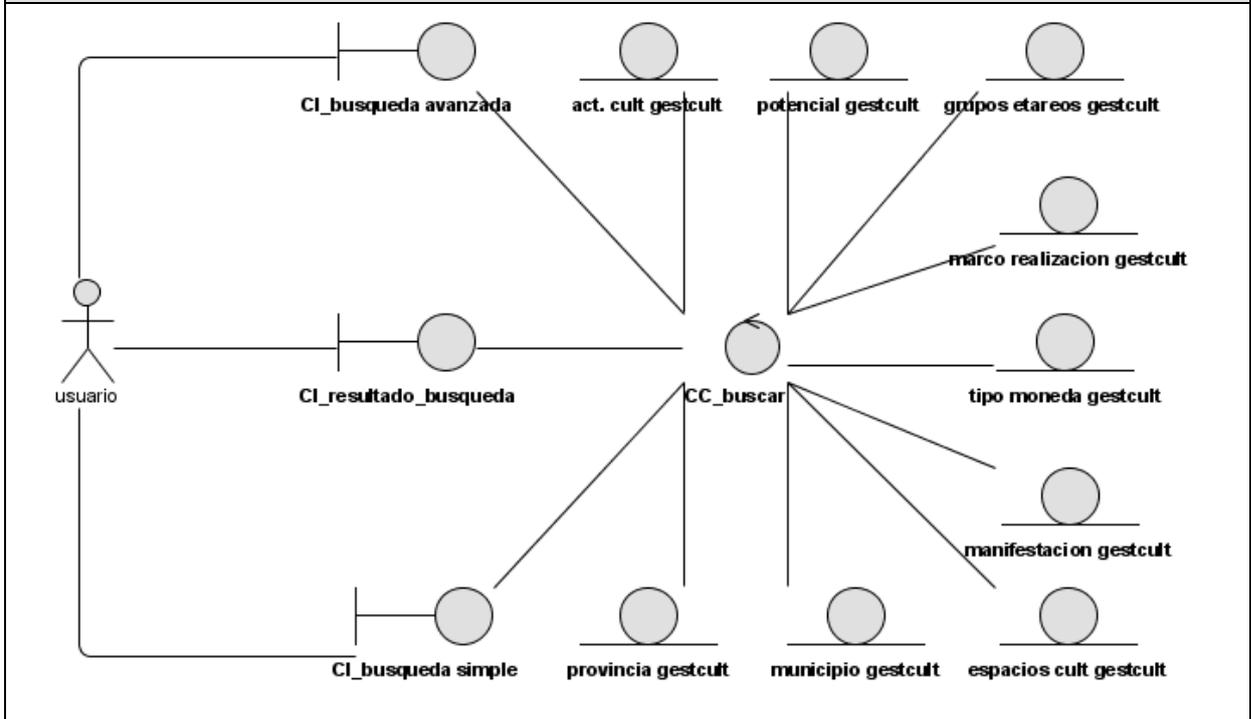
3.2 Modelo de Análisis

El modelo de análisis ayuda a refinar los requisitos, permite reflexionar sobre los aspectos internos del sistema y nos proporciona una estructura centrada en el mantenimiento, en aspectos tales como la flexibilidad ante cambios y la reutilización. Además ofrece un mayor poder expresivo y una mayor formalización (JACOBSON *et al.* 2000).

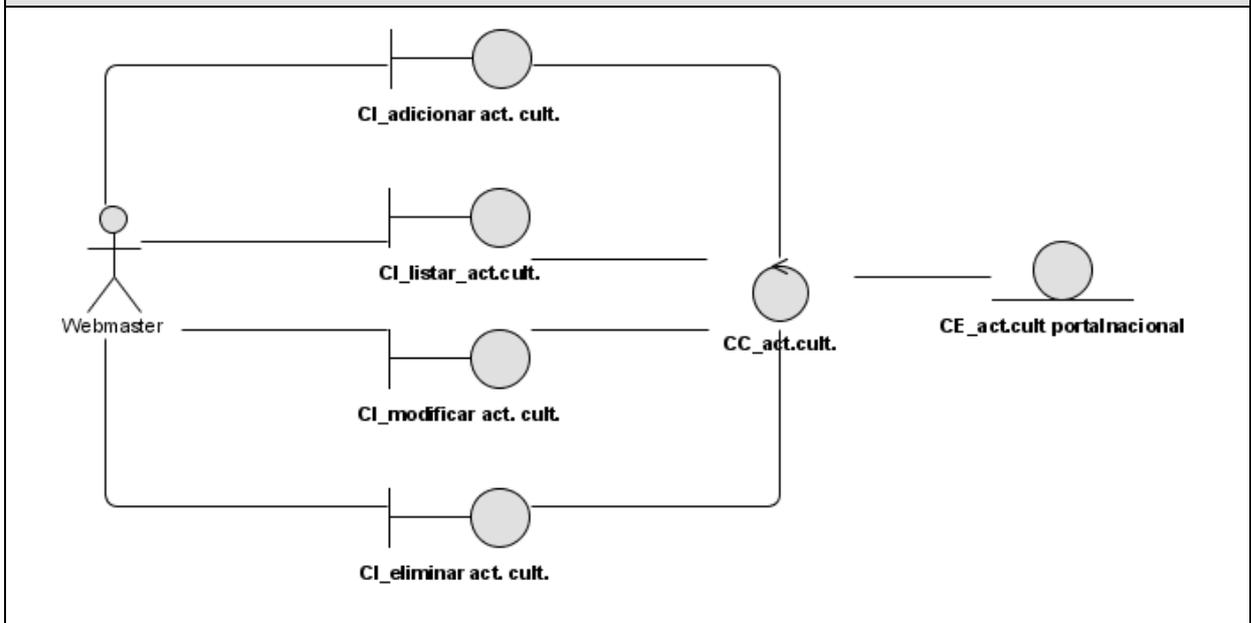
3.2.1 Diagramas de clases de análisis



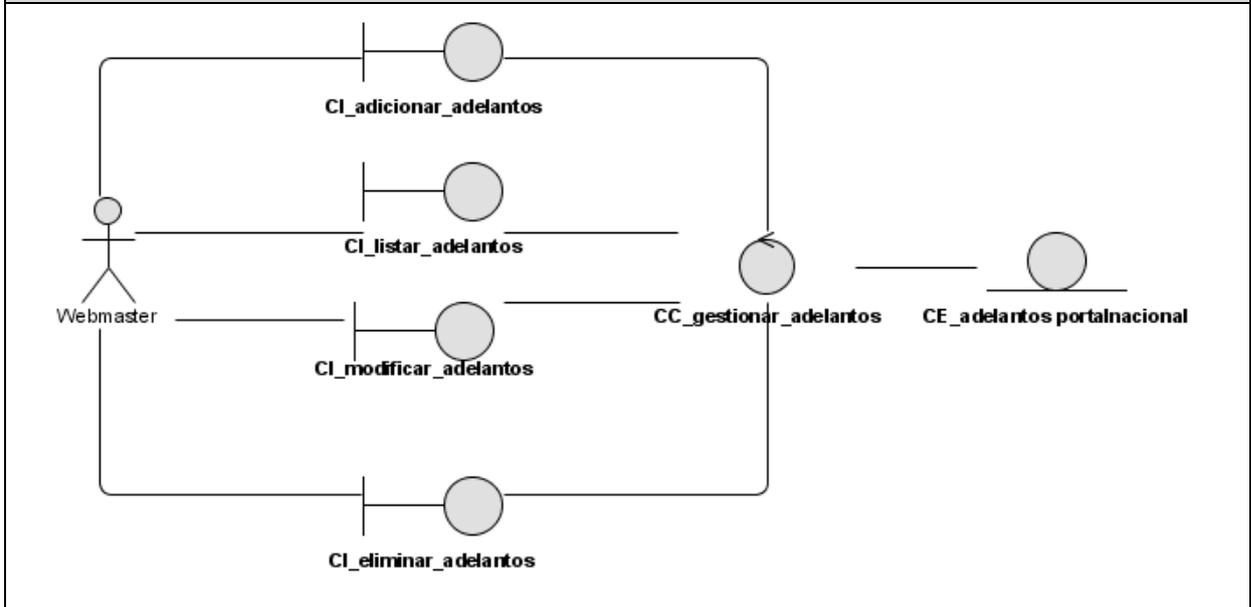
Búsquedas



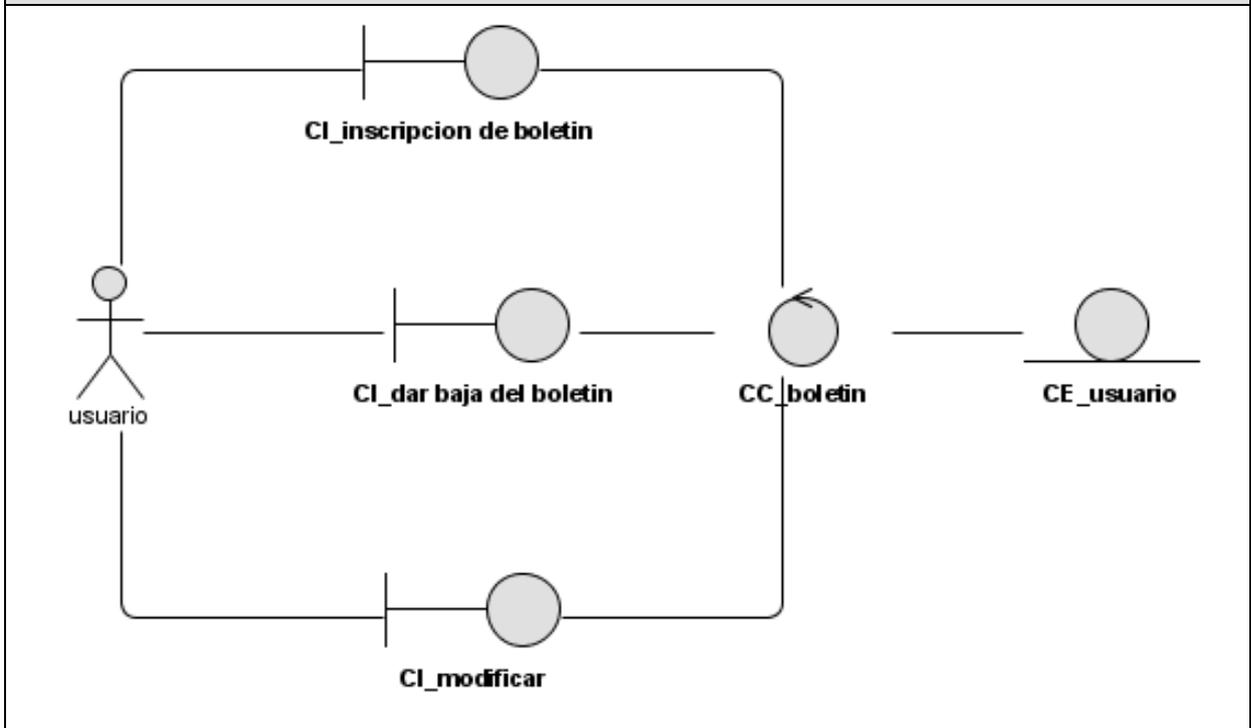
Gestionar actividad cultural



Gestionar Adelantos

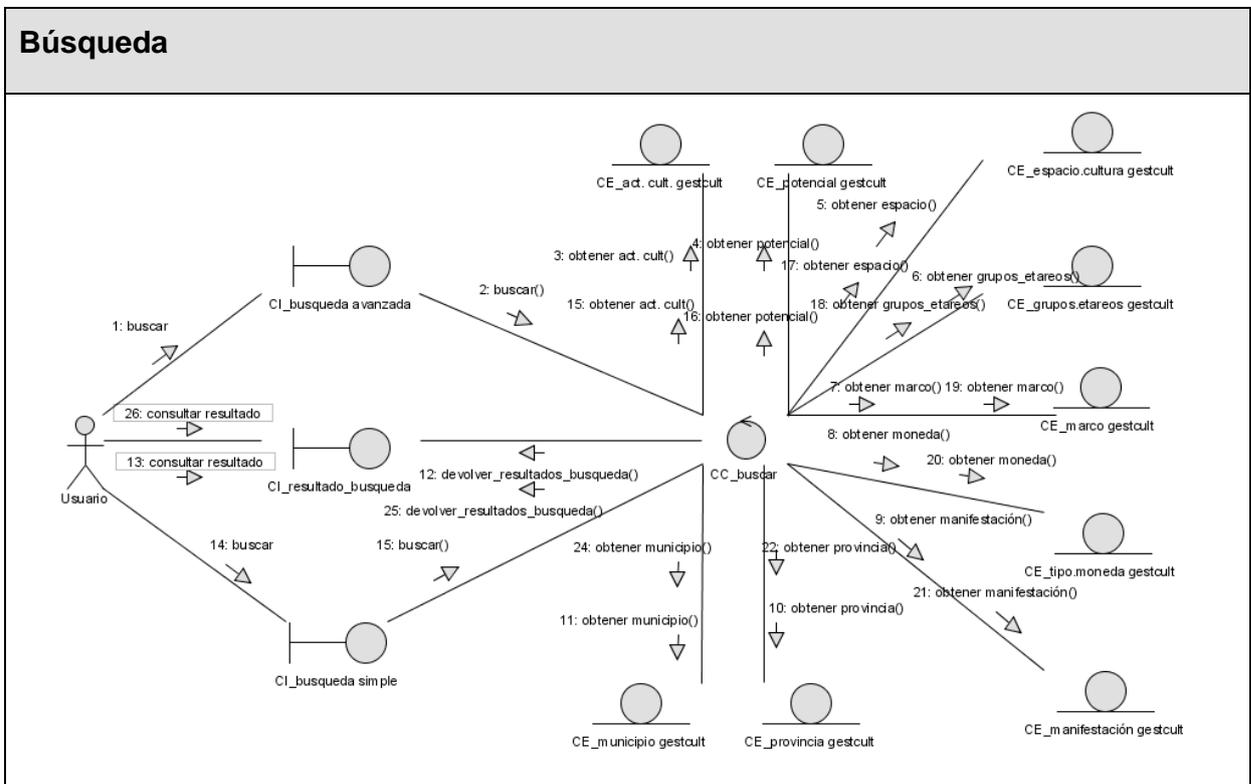
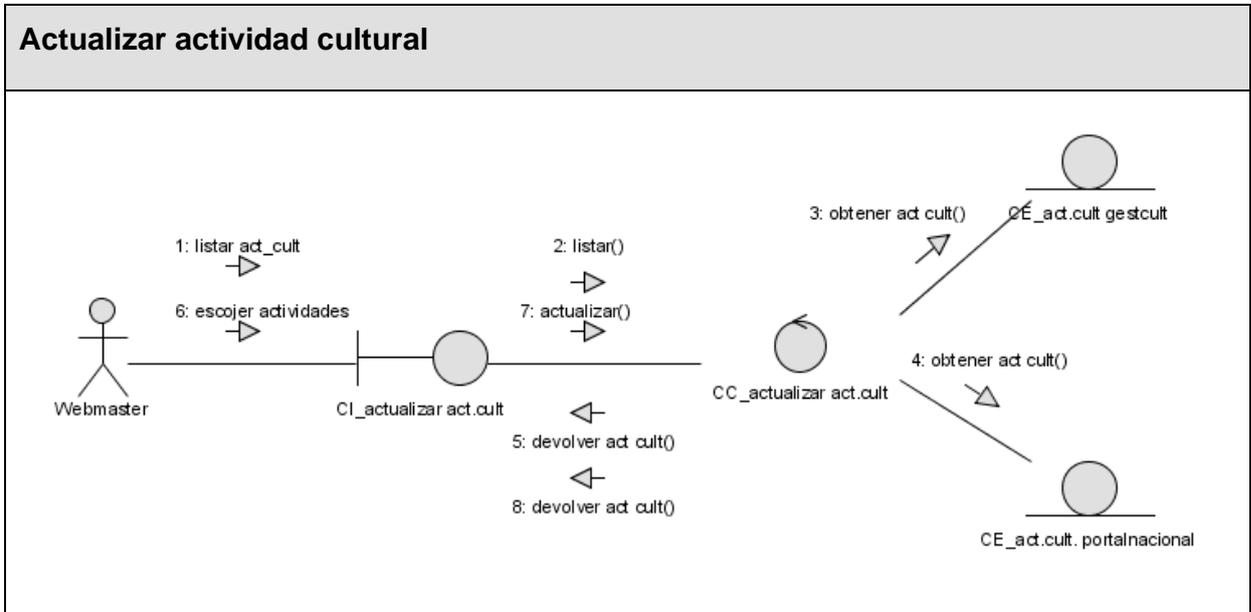


Boletín informativo

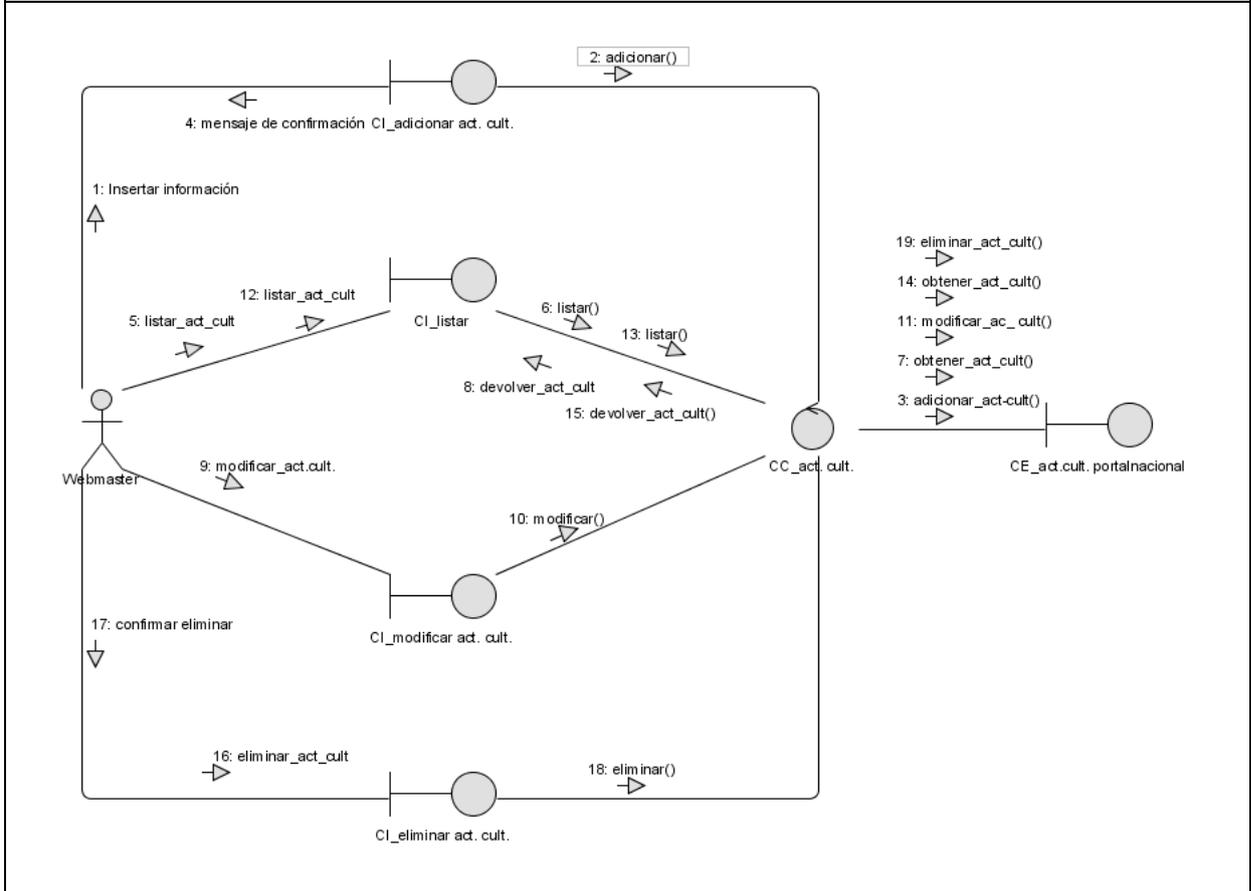


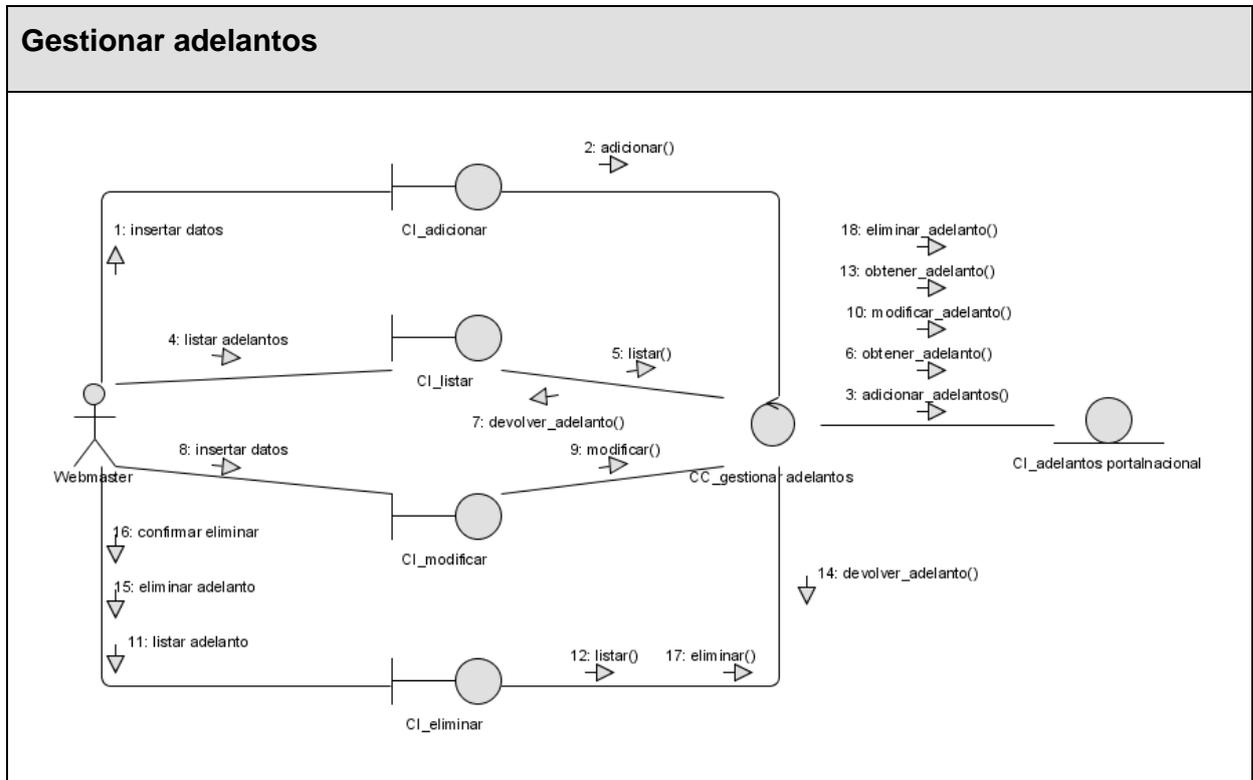
Se omiten algunos diagramas por su parecido con los ya ilustrados o por ser de menor importancia

3.2.2 Diagramas de interacción

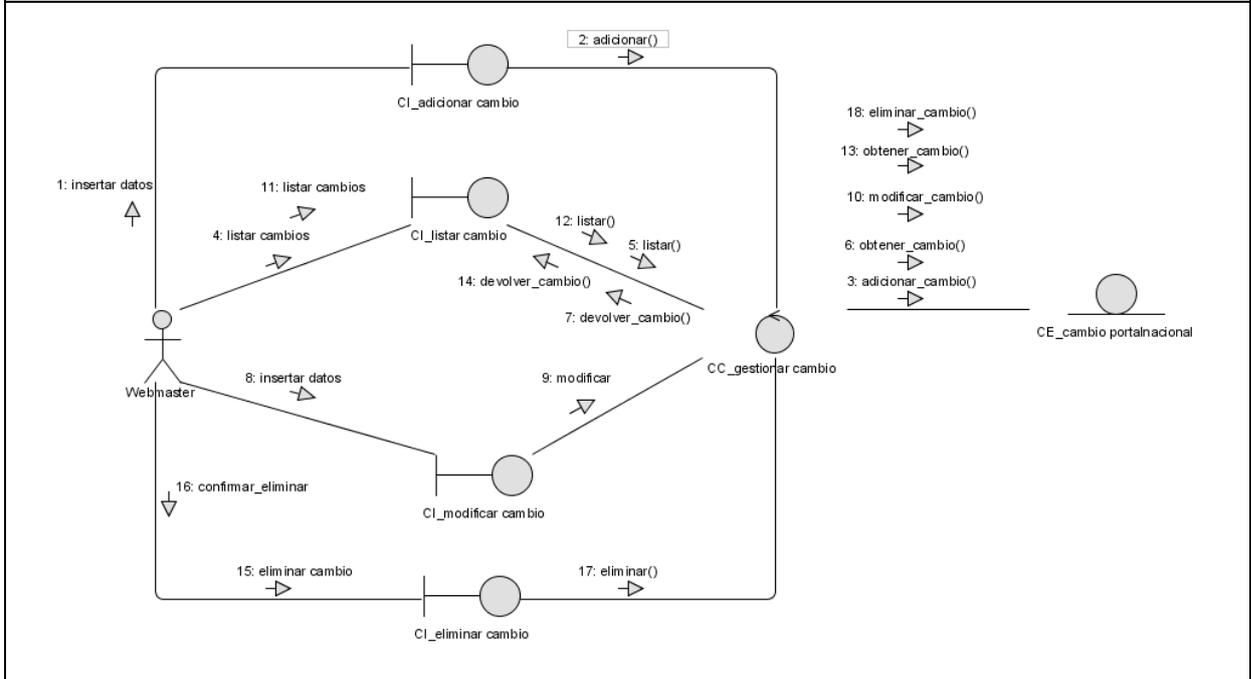


Gestionar actividad cultural

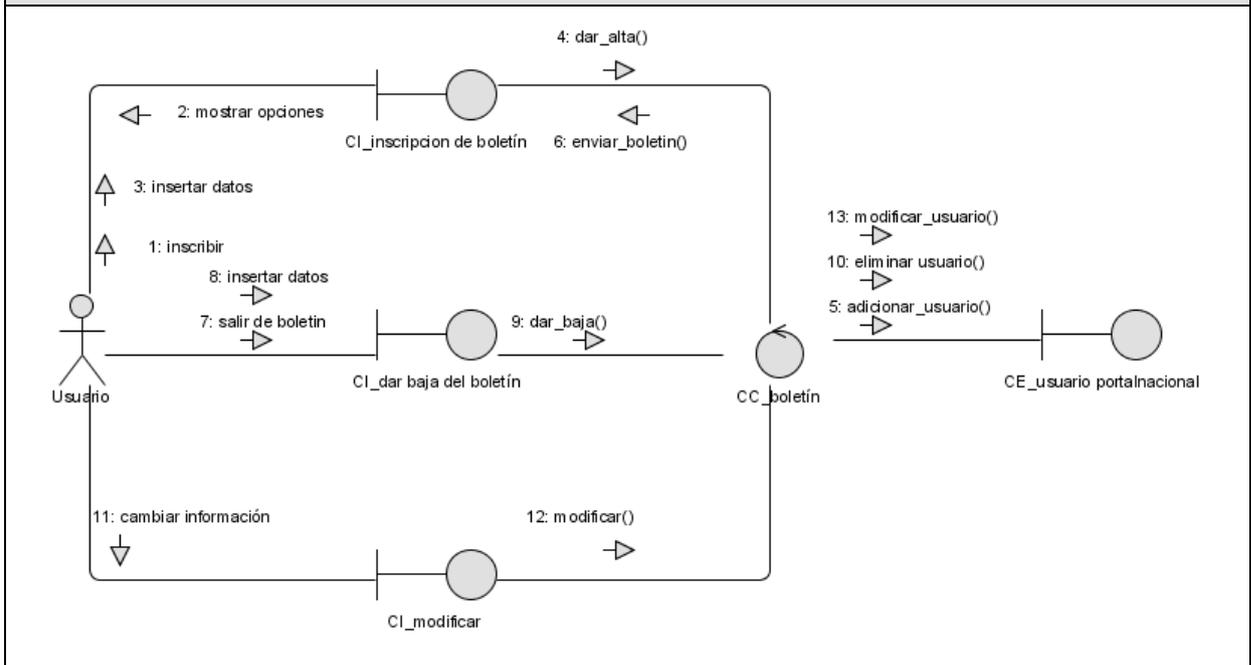




Gestionar cambios



Boletín Informativo

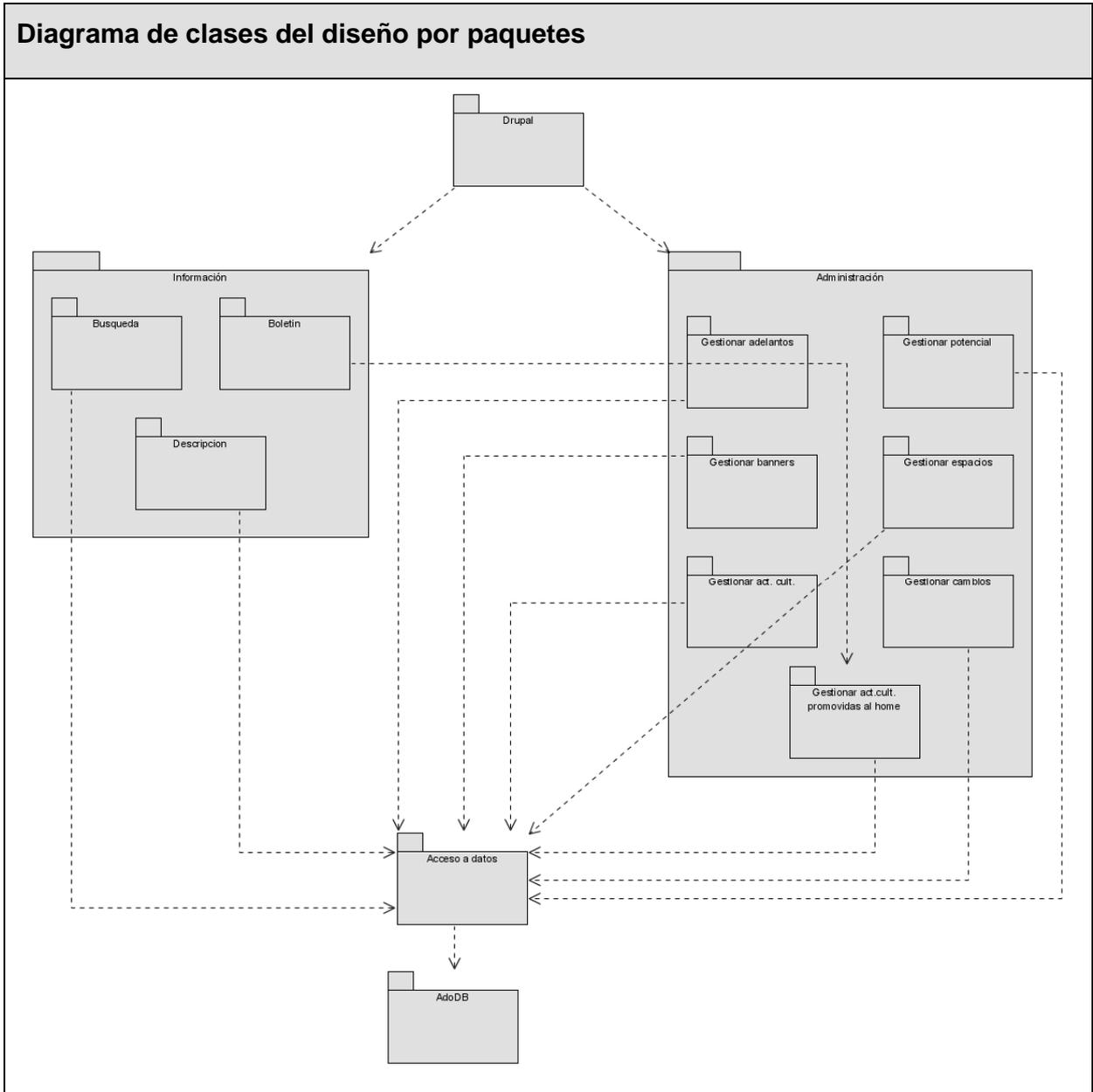


Se omiten algunos diagramas debido a su semejanza con los ya expuestos o porque son de menor importancia.

3.3 Modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además sirve de abstracción de la implementación del sistema y es de ese modo, utilizada como una entrada fundamental de las actividades de implementación (JACOBSON *et al.* 2000).

3.3.1 Diagrama de clases del diseño



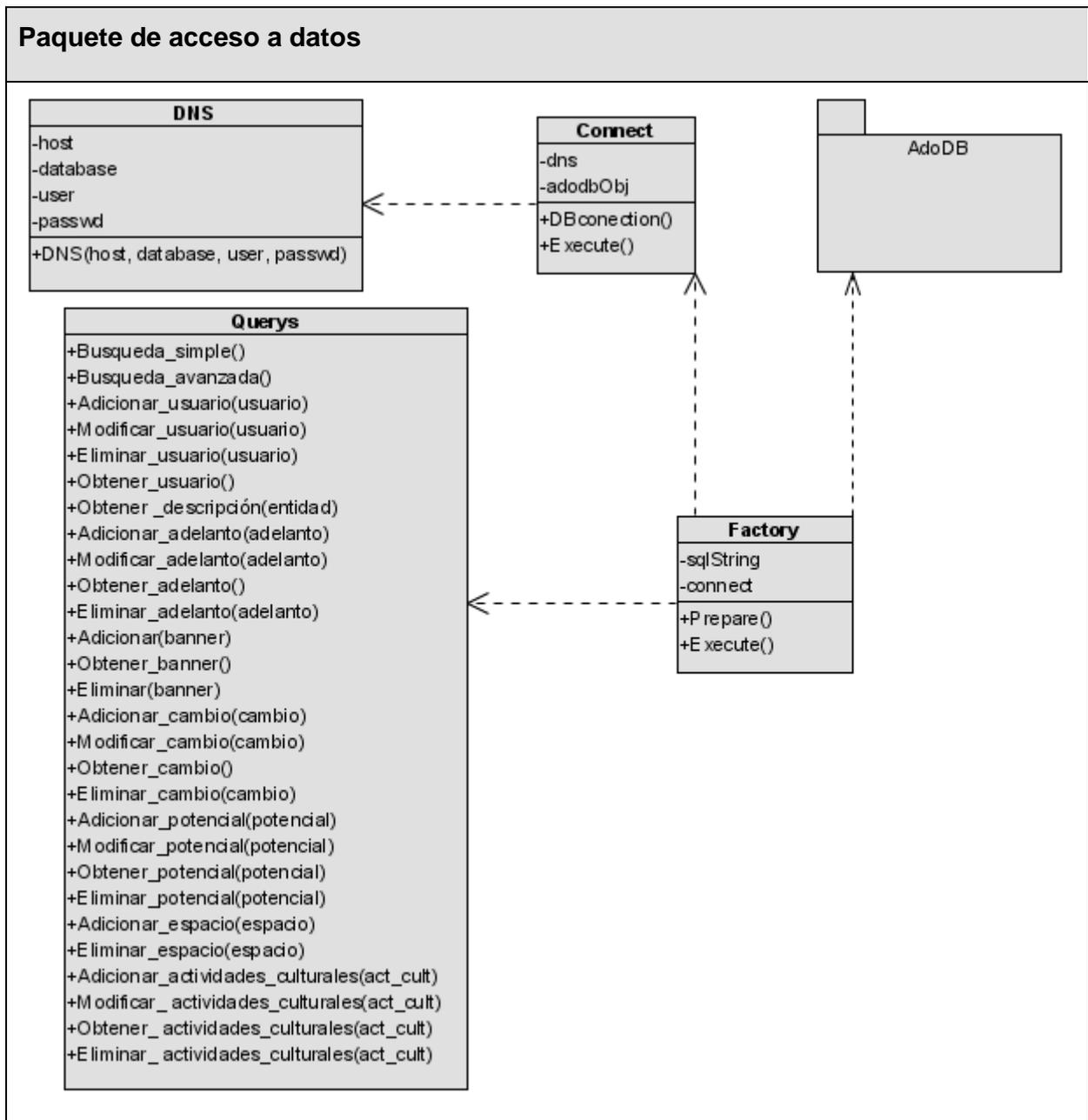


Diagrama de clases del diseño. Boletín informativo. Paquete de información.

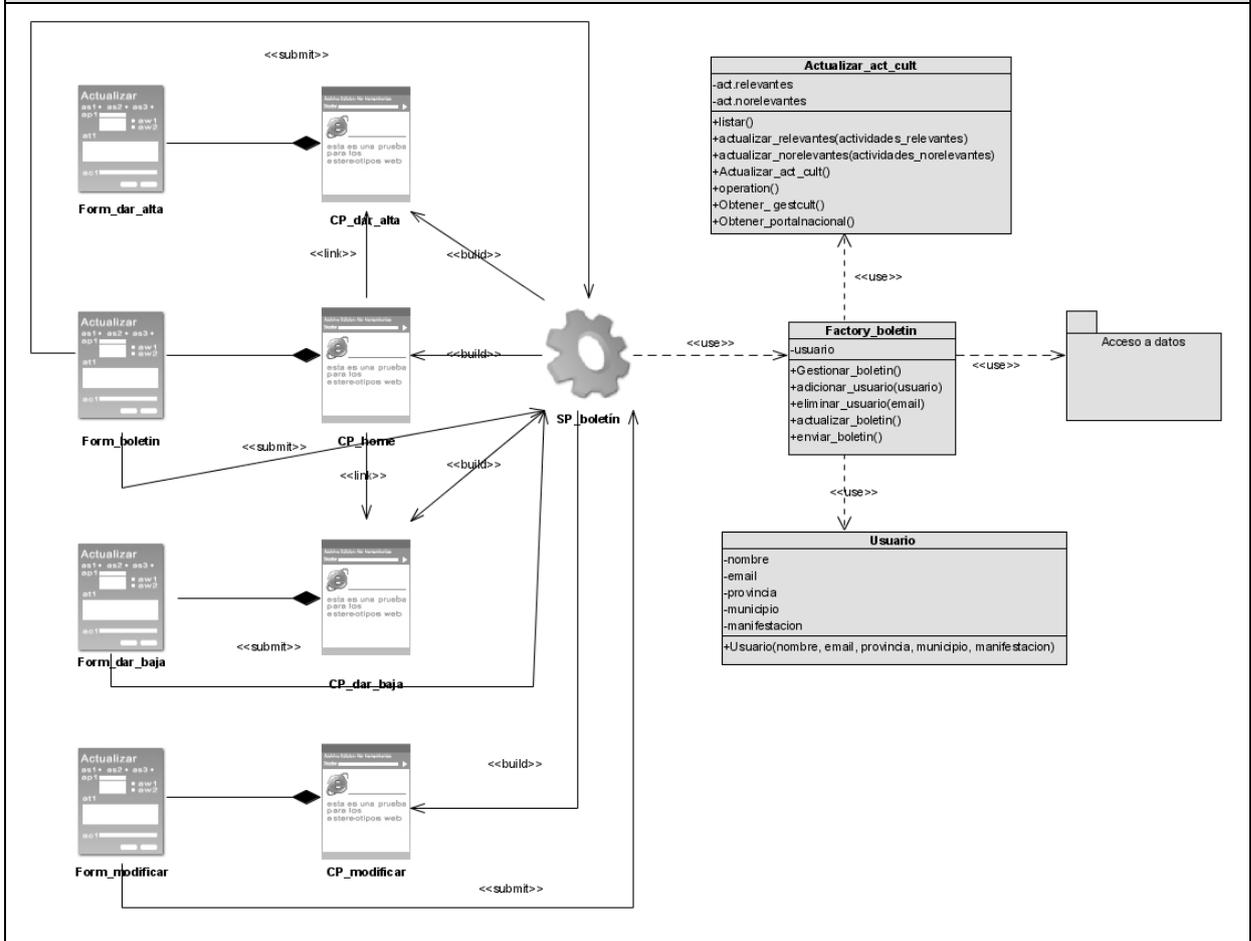


Diagrama de clases del diseño. Búsqueda. Paquete de información.

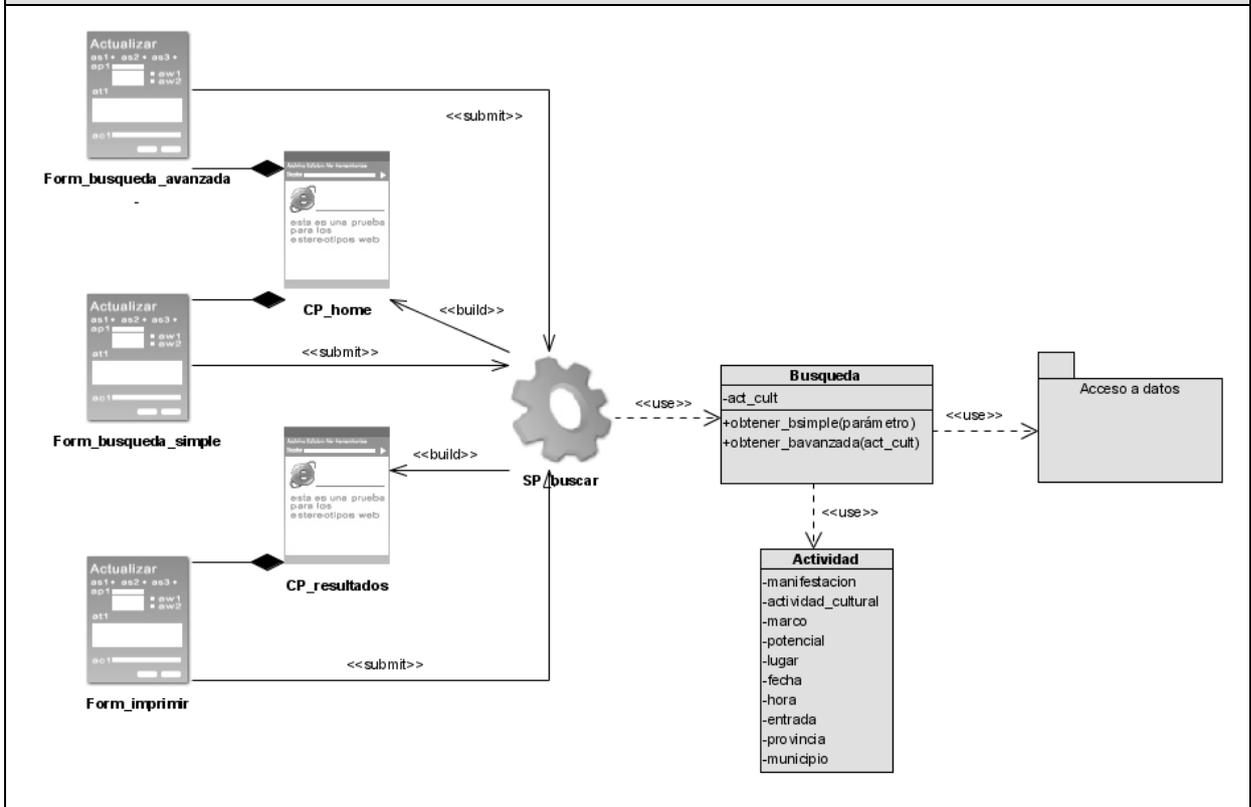


Diagrama de clases del diseño. Actualizar actividad cultural. Paquete de Administración

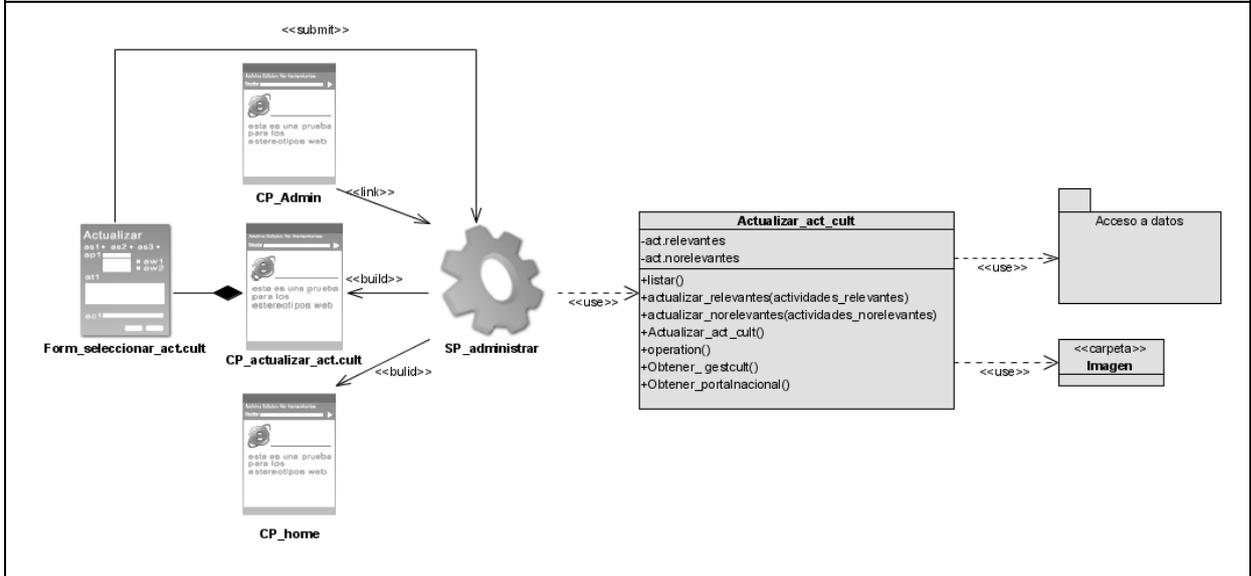


Diagrama de clases del diseño. Gestionar actividad cultural. Paquete de administración

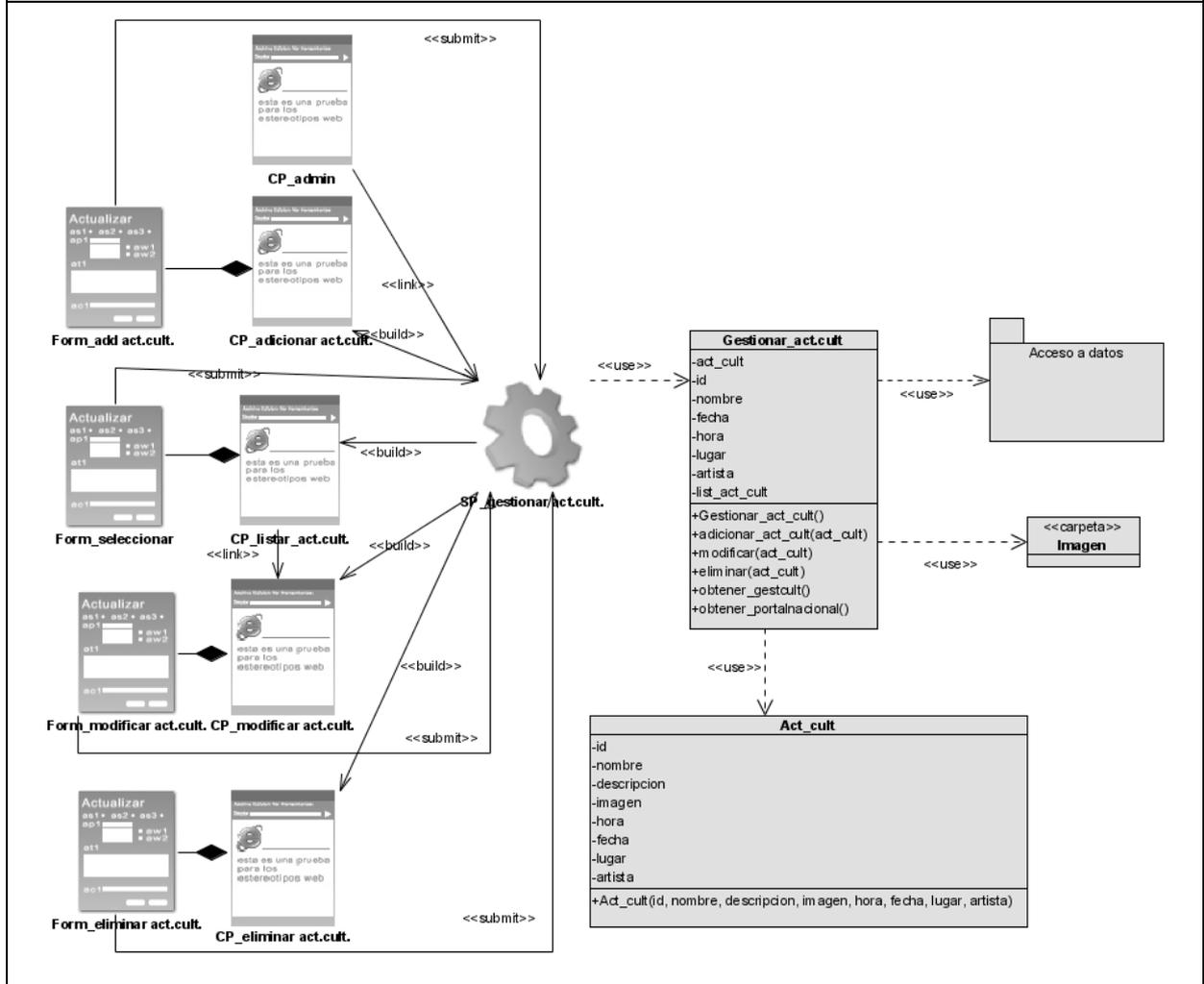
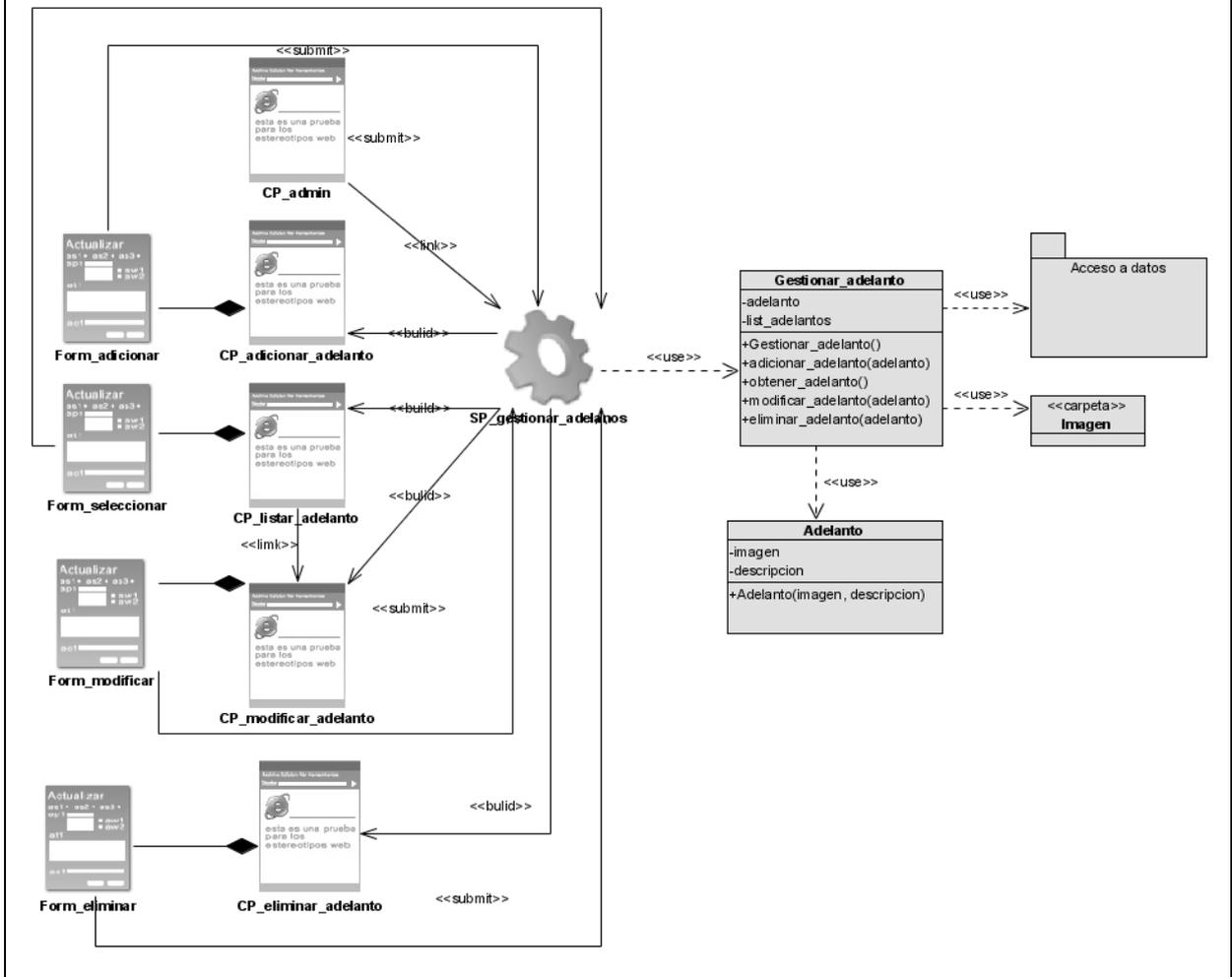


Diagrama de clases del diseño. Gestionar adelanto. Paquete de administración



Se omiten algunos diagramas debido a que son muy parecidos a los ya expuestos o son de menor importancia.

3.3.1.1 Descripción de las clases de diseño

A continuación se explican mediante tablas las clases del diseño, dando a conocer para ello, datos como, el tipo de clase, atributos y métodos.

Capítulo3: Análisis y diseño del sistema

Nombre: Connect	
Tipo de clase: Controladora	
Atributos	Tipo
Adodb	Object
Dns	Object
Métodos:	
Nombre:	Connect()
Descripción:	Constructor de la clase.
Nombre:	Execute(query, params)
Descripción:	Ejecuta y devuelve los resultados de la consulta.
Nombre:	Conexion_gestult()
Descripción:	Abre la conexión a la BD gestcult
Nombre:	Conexion_potalnacional()
Descripción:	Abre la conexión a la BD portalnacional

Nombre: Factory	
Tipo de clase:	
Atributos	Tipo
sqlString	string
Parameters	Array()
Conn	Object
Métodos:	
Nombre:	Factory()
Descripción:	Constructor de la clase.
Nombre:	Prepare(\$query)
Descripción:	Le dice a la clase qué consulta se va a ejecutar, no la ejecuta, solo la asigna a la variable sqlString.
Nombre:	addParameter()
Descripción:	Adiciona a la consulta un parámetro de forma tal que sea válida la consulta para las clases ADOdb.
Nombre:	Execute()
Descripción:	Ejecuta la consulta que esta en prepare ().

Nombre: Gestionar_act_cult	
Tipo de clase: Controladora.	
Atributos	Tipo
act_cult	Object
Id	int
Nombre	string
Fecha	date
Hora	int
Lugar	string
Artista	string
list_act_cult	Array()
Métodos:	
Nombre:	Gestionar_act_cult()
Descripción:	Constructor de la clase
Nombre:	Obtener_gestcult()
Descripción:	Obtiene de la BD gestcult el identificador, nombre, lugar, fecha, hora y los artistas que participan en las actividades culturales.
Nombre:	Adicionar_act_cult(act_cult)

Capítulo3: Análisis y diseño del sistema

Descripción:	Adiciona una nueva actividad cultural a la BD del portal con los datos resultantes de la función "Obtener_gestcult()" y los datos: "descripción" e "imagen".
Nombre:	Obtener _portal nacional()
Descripción:	Obtiene y lista las actividades culturales que están en la BD portal nacional.
Nombre:	Modificar(act _cult)
Descripción:	Modifica la actividad cultural pasada como parámetro y que se encuentra registradas en la BD portal nacional.
Nombre:	Eliminar(act _cult)
Descripción:	Elimina de la BD del portal la actividad cultural que se le pasa como parámetro.

Nombre: Actualizar _act _cult	
Tipo de clase: Controladora.	
Atributos	Tipo
Actividades _relevantes	Array(20)
Actividades _no relevantes	Array(10)
Métodos:	
Nombre:	Actualizar _act.cult ()

Capítulo3: Análisis y diseño del sistema

Descripción:	Constructor de la clase
Nombre:	Obtener_ gestcult()
Descripción:	Es el encargado de listar el nombre, hora, lugar y artistas o agrupaciones culturales que participan en todas las actividades culturales registradas en la BD gestcult desde la fecha actual. Retorna una lista de actividades culturales no relevantes.
Nombre:	Obtener _portal nacional()
Descripción:	Es el encargado de listar la descripción, imágenes, nombre, hora, lugar y artistas o agrupaciones culturales que participan en todas las actividades culturales registradas en la BD portal nacional desde la fecha actual. Retorna una lista de actividades culturales relevantes.
Nombre:	Listar()
Descripción:	Es la función encargada de listar las actividades culturales de ambas BD sin que se repita alguna
Nombre:	Actualizar _relevantes (actividades _relevantes)
Descripción:	Lista en el home sólo las actividades culturales que son relevantes; en caso de que ya existan actividades publicadas las actualiza.
Nombre:	Actualizar _no relevantes (actividades _no relevantes)
Descripción:	Lista en el home las actividades culturales de menos importancia; en caso de que ya existan actividades publicadas las actualiza.

Nombre: Gestionar_boletín	
Tipo de clase: Controladora.	
Atributos	Tipo
Usuario	Object
Métodos:	
Nombre:	Gestionar_boletín()
Descripción:	Constructor de la clase
Nombre:	Adicionar _usuario(usuario)
Descripción:	Adiciona un nuevo usuario a la BD portal nacional si no existe.
Nombre:	Eliminar _usuario(email)
Descripción:	Elimina a un usuario de la BD.
Nombre:	Actualizar _boletín()
Descripción:	Pasa la información que está publicada en la página principal del portal para el boletín que se va a enviar. Esta acción se realiza cada 7 días. En este método se hace una llamada a la función Enviar ().
Nombre:	Enviar _boletín()
Descripción:	Envía el boletín al buzón de correo del usuario.

Nombre: Búsqueda	
Tipo de clase: Controladora	
Atributos	Tipo
act_cult	Object
Métodos:	
Nombre:	Obtener_bsimple(parámetro)
Descripción:	Obtiene y retorna los resultados de la búsqueda simple
Nombre:	Obtener_bavanzada(act_cult)
Descripción:	Obtiene y retorna los resultados de la búsqueda avanzada.

Nombre: Gestionar_adelantos	
Tipo de clase: Controladora.	
Atributos	Tipo
adelanto	Object
List _adelantos	Array()
Métodos:	
Nombre:	Gestionar _adelantos()
Descripción:	Constructor de la clase
Nombre:	Adicionar _adelanto(adelanto)
Descripción:	Método para adicionar adelantos.
Nombre:	Obtener _adelanto()
Descripción	Método para obtener y listar los adelantos.
Nombre:	Modificar _adelanto(adelanto)
Descripción:	Se usa para modificar el adelanto pasado por parámetro almacenado en la BD portal nacional.
Nombre:	Eliminar _adelanto(adelanto)
Descripción	Se usa para eliminar el adelanto pasado por parámetro almacenado en la BD portal nacional.

Nombre: Gestionar_banner	
Tipo de clase: Controladora.	
Atributos	Tipo
banner	Object
list_banner	Array()
Métodos:	
Nombre:	Adicionar(banner)
Descripción:	Se usa para adicionar un nuevo banner.
Nombre:	Obtener()
Descripción:	Este método es para obtener y listar los banners almacenados.
Nombre:	Eliminar(banner)
Descripción	Su función es la de eliminar un banner determinado.

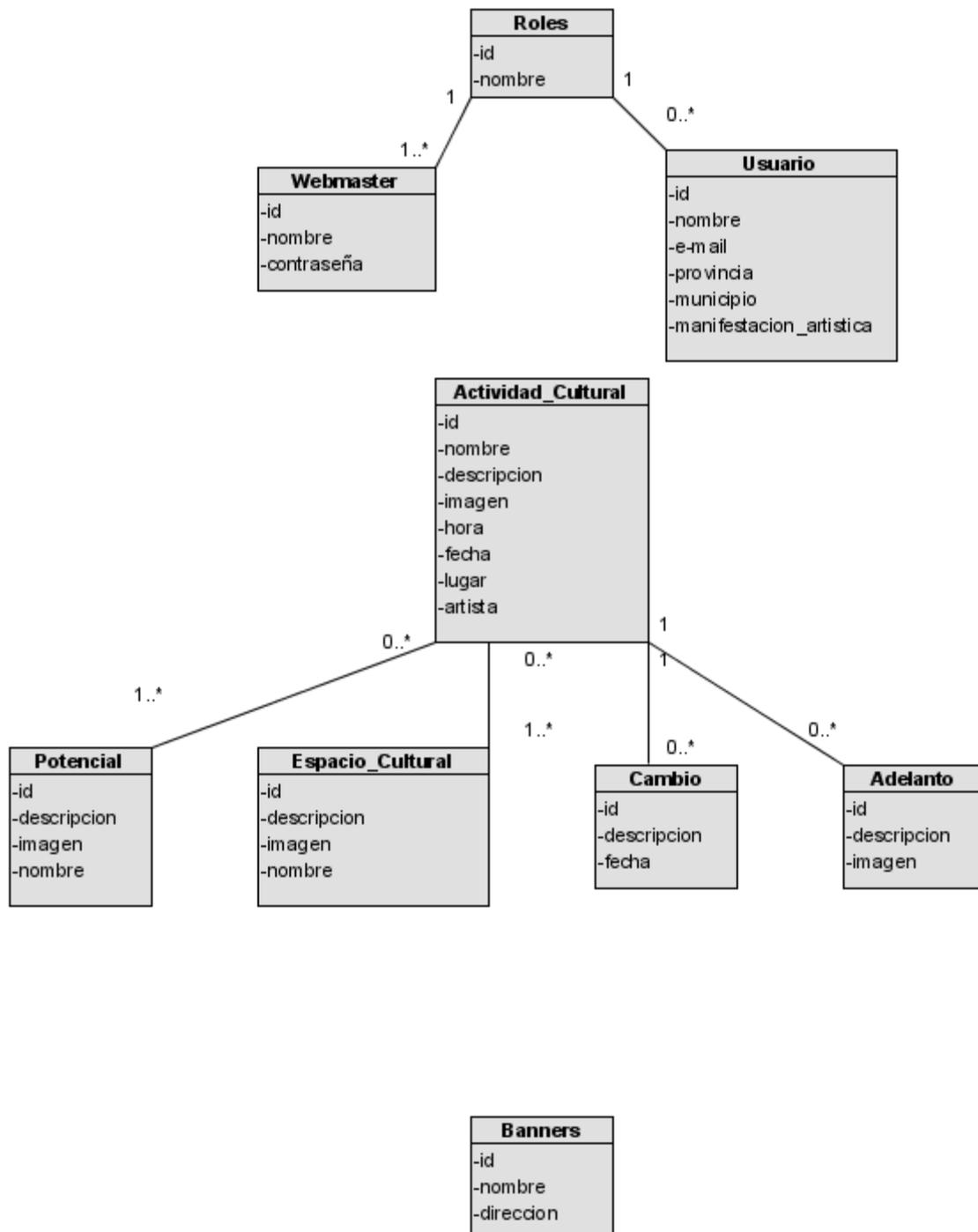
Nombre: Gestionar_cambio	
Tipo de clase: Controladora.	
Atributos	Tipo
cambio	Object
List _cambios	Array()
Métodos:	
Nombre:	Gestionar _cambio()
Descripción:	Constructor de la clase
Nombre:	Adicionar _cambio(cambio)
Descripción:	Método para adicionar un cambio
Nombre:	Obtener()
Descripción:	Se usa para obtener y listar todos los cambios almacenados en la BD.
Nombre:	Modificar(cambio)
Descripción	Se usa para modificar la información referente a los cambios culturales.
Nombre:	Eliminar(cambio)
Descripción	Elimina de la BD el cambio que se le pasa por parámetro.

Nombre: Gestionar_espacios	
Tipo de clase: Controladora.	
Atributos	Tipo
espacio	Object
List _espacios	Array()
Métodos:	
Nombre:	Gestionar _espacios()
Descripción:	Constructor de la clase
Nombre:	Adicionar _espacio(espacio)
Descripción:	Adiciona nuevos espacio culturales a la BD
Nombre:	Obtener _espacio()
Descripción	Obtiene todos los espacios de la BD y los lista.
Nombre:	Modificar _espacio(espacio)
Descripción	Modifica los datos de un espacio especificado.
Nombre:	Eliminar(espacio)
Descripción:	Elimina el espacio pasado como parámetro

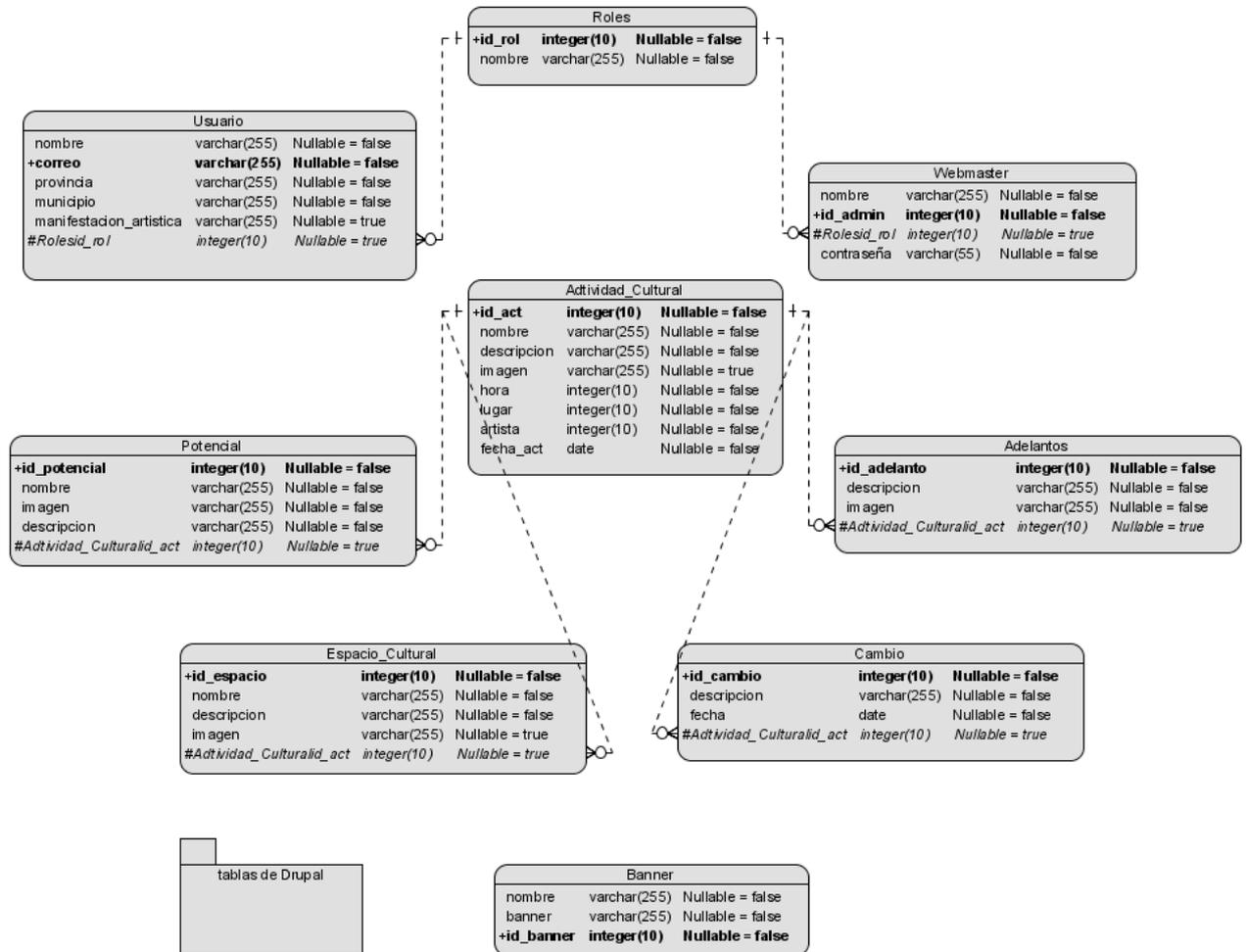
Nombre: Gestionar_potencial	
Tipo de clase: Controladora.	
Atributos	Tipo
potencial	Object
List _potencial	Array()
Métodos:	
Nombre:	Gestionar _potencial()
Descripción:	Constructor de la clase.
Nombre:	Adicionar _potencial(potencial)
Descripción:	Adiciona un potencial a la BD
Nombre:	Obtener _potencial()
Descripción:	Obtiene y lista todos los potenciales de la tabla.
Nombre:	Modificar _potencial(potencial)
Descripción:	Modifica la información del potencial pasado por parámetro.
Nombre:	Eliminar _potencial(potencial)
Descripción:	Elimina al potencial entrado como parámetro.

3.3.2 Diseño de la Base de Datos

3.3.2.1 Diagrama de clases persistentes



3.3.2.2 Modelo de datos



3.3.2.3 Descripción de las tablas

Seguidamente se describe cada tabla de la base de datos.

Nombre: Roles		
Descripción: Tabla que permite almacenar los tipos de roles que va a tener el portal. Dichos roles pueden ser “webmaster” o usuario común.		
Atributo	Tipo	Descripción
id_rol	integer(10)	Llave primaria de la tabla que asegura

Capítulo3: Análisis y diseño del sistema

		la identificación de cada rol.
nombre	varchar(50)	Nombre del rol.

Nombre: Usuario		
Descripción: Tabla que permite almacenar los datos de los usuarios a los que va dirigido el boletín informativo.		
Atributo	Tipo	Descripción
correo	varchar(100)	Llave primaria de la tabla que asegura la no repetición de un usuario.
nombre	varchar(50)	Nombre del usuario.
provincia	varchar(50)	Provincia de la que se desea la cartelera
municipio	varchar(50)	Municipio del que se desea la cartelera
manifestación	varchar(50)	Tipo de manifestación que solicita el usuario.

Capítulo3: Análisis y diseño del sistema

Nombre: Webmaster		
Descripción: Tabla que permite almacenar el nombre y contraseña de todos los webmaster.		
Atributo	Tipo	Descripción
id_webmaster	integer(10)	Llave primaria de la tabla que asegura la identificación única de cada webmaster.
contraseña	varchar(50)	Contraseña del webmaster.
nombre	varchar(50)	Nombre del webmaster

Nombre: Actividad cultural		
Descripción: Tabla que permite almacenar la información significativa para el portal de las actividades culturales.		
Atributo	Tipo	Descripción
id_act.cult	integer(10)	Llave primaria de la tabla que asegura la identificación de cada actividad.
Nombre	varchar(50)	Nombre de la actividad.
descripción	varchar(50)	Breve descripción sobre la actividad
imagen	varchar(50)	Imagen sobre la actividad. Este campo puede ser nulo.

Capítulo3: Análisis y diseño del sistema

Nombre: Potencial		
Descripción: Tabla que permite almacenar a todos los artista o agrupaciones que participan en las actividades culturales.		
Atributo	Tipo	Descripción
id_potencial	integer(10)	Llave primaria de la tabla que asegura la identificación del artista.
nombre	varchar(50)	Nombre del artista
descripción	Varchar()50	Breve reseña de su trayectoria en el mundo del arte.
imagen	varchar(50)	Foto del artista.

Nombre: Espacio cultural		
Descripción: Tabla que permite almacenar los lugares donde se realizan las actividades culturales		
Atributo	Tipo	Descripción
id_espacio	integer(10)	Llave primaria de la tabla que asegura la identificación de cada espacio cultural.
nombre	varchar(50)	Nombre del espacio.
descripción	varchar(50)	Descripción de los espacios culturales, esta puede ser una breve reseña.
Imagen	varchar(50)	Ubicación de las imágenes correspondientes a los espacios

Capítulo3: Análisis y diseño del sistema

		culturales.
--	--	-------------

Nombre: Cambios

Descripción: Tabla que permite almacenar los cambios que puedan sufrir las actividades culturales

Atributo	Tipo	Descripción
id_cambio	integer(10)	Llave primaria de la tabla que asegura la identificación de cada cambio.
descripción	varchar(50)	Descripción de las causas por las que sufrió cambios la actividad cultural
fecha	date	Fecha en la que se realizó el cambio.

Nombre: Adelantos

Descripción: Tabla que permite almacenar los adelantos de las actividades culturales mas significativas.

Atributo	Tipo	Descripción
id_adelanto	integer(10)	Llave primaria de la tabla que asegura la identificación de cada adelanto.
nombre	varchar(50)	Nombre del adelanto.
descripción	varchar(50)	Descripción de los adelantos.
Imagen	varchar(50)	Ubicación de las imágenes correspondientes a los adelantos.

Nombre: Banner		
Descripción: Tabla que permite almacenar los banners de portal		
Atributo	Tipo	Descripción
id_banner	integer(10)	Llave primaria de la tabla que asegura la identificación de cada banner.
Nombre	varchar(50)	Nombre del banner.
Banner	varchar(50)	Ubicación de los banners.

3.4 Patrón de diseño utilizado

Para esta aplicación se utilizó el patrón de *modelo vista controlador* el cual establece que los objetos modelos no deberían conocer directamente los objetos vistas ni estar directamente acoplados a ellos (LARMAN 1999).

Los métodos de una clase de modelo no deberían contener instrucciones que envíen mensajes directos a un objeto vista; este tipo de clase no debería conocer las interfaces del usuario ni relacionarse con ellas mediante códigos.

En el diseño del portal se puede observar que la clase controladora es la que media entre las clases interfaz, que sería el objeto vista, e identidad que viene siendo el objeto modelo.

La utilización de este patrón permite vistas simultáneas y múltiples de un mismo objeto modelo. Permite el desarrollo de la interfaz independiente del objeto modelo. Reduce al mínimo el impacto de un cambio de los requerimientos de la clase interfaz.

3.5 Interfaces

May 18 2008

la PAPELETA

Cartelera Cultural Cubana

Búsqueda Rápida

¡Suscríbese a nuestro boletín!
Manténgase informado con lo mejor de la semana directamente en su buzón.

nombre e-mail

¿darse baja?

PARA HOY

Recomendamos:



Gran concierto "Sin palabras" de Kelvis Ochoa y su grupo en el teatro Karl Marx a las 8:30pm. Como invitados estarán Vivanco, Telmaris y Sexto Sentido entre otros. Aquí se pueden ofrecer datos adicionales sin exceder las 6 líneas de texto

[ver más detalles](#)

Además:

Nuevo Festival de Cine Latinoamericano
por: Alberto González Prieto, Gustavo Roselló García, Tomás Rodríguez Mortalván.
(cine) [ver más detalles](#)

Exposición "Tubosutra"
por: René Francisco
(artes plásticas) [ver más detalles](#)

Concierto "With a little help from my friends"
por: Síntesis, Gerardo Alfonso, Silvio Rodríguez, Acetuna sin hueso, Carlos Varela, David Torres, Orichas, Rochi.
(música) [ver más detalles](#)

"La Cucarachita Martínez"
por: La Colmenita
(teatro) (infantiles) [ver más detalles](#)

Peña humorística "Misión Imposible"
por: Adrián Verazaín, Alberto Dorado, Julio Bacallao, David Álvarez, Luis Silva.
(combinadas) [ver más detalles](#)

[Ver todo lo que hay para hoy](#)

PRÓXIMOS DÍAS

Recomendamos:



Exposición "A través de mí" del joven artista Javier González González, en la Casa de la Amistad. El trabajo muestra el mundo surrealista del artista con un énfasis marcado en la mujer. Las pinturas estarán a la venta para la población.

[ver más detalles](#)

Además:

Nuevo Festival de Cine Latinoamericano
por: Alberto González Prieto, Gustavo Roselló García, Tomás Rodríguez Mortalván.
(cine) [ver más detalles](#)

Exposición "Tubosutra"
por: René Francisco
(artes plásticas) [ver más detalles](#)

Concierto "With a little help from my friends"
por: Síntesis, Gerardo Alfonso, Silvio Rodríguez, Acetuna sin hueso, Carlos Varela, David Torres, Orichas, Rochi.
(música) [ver más detalles](#)

"La Cucarachita Martínez"
por: La Colmenita
(teatro) (infantiles) [ver más detalles](#)

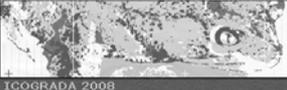
Peña humorística "Misión Imposible"
por: Adrián Verazaín, Alberto Dorado, Julio Bacallao, David Álvarez, Luis Silva.
(combinadas) [ver más detalles](#)

[Ver más para los próximos días](#)

PRÓXIMOS EVENTOS



XXII Feria Latinoamericana del Turismo



ICOGRADA 2008



Circuit Rock Dur 2008

¡ADELANTOS!

La compañía **Teatro de la Luna** prepara una nueva versión de la obra "Los Siervos". Según su director, estará lista para el mes de agosto.

[via La Jiribilla](#)

El grupo humorístico **Punto y Coma** anunció que estará presentándose próximamente en las provincias orientales.

[via Cubarte](#)



El famoso Groove DJ declaró que desea dar un espectáculo en la tribuna antipemalista de la capital cubana este año.

[via Movimiento](#)

CAMBIOS DE ÚLTIMA HORA

Suspendido el concierto "Civilización" de X. Alfonso por lluvias en la Tribuna Antipemalista. El concierto se cambia para el próximo 3 de Junio. Para ver más información sobre esta actividad [haga click aquí](#).

Se adelanta la inauguración de la exposición "Cubaneando" del artista Sergio Armando García. Ahora tendrá lugar el día 25 de Abril a las 5:00pm en el mismo lugar. Para ver más información sobre esta actividad [haga click aquí](#).

Suspendido el concierto "Civilización" de X. Alfonso por lluvias en la Tribuna Antipemalista. El concierto se cambia para el próximo 2 de Junio. Para ver más información sobre esta actividad [haga click aquí](#).

Se adelanta la inauguración de la exposición "Cubaneando" del artista Sergio Armando García. Ahora tendrá lugar el día 25 de Abril a las 5:00pm en el mismo lugar. Para ver más información sobre esta actividad [haga click aquí](#).

Búsquedas del momento

- Música en Ciudad de la Habana
- Artes plásticas en Pinar del Río
- Charanga Habanera
- Humor en Ciudad de la Habana
- Teatro en Matanzas
- VAN VAN
- Festival de Cine Francés
- Punto y Coma
- Danza en Santiago de Cuba
- Ballet en Ciudad de la Habana

Otros sitios de interés

- www.cubarte.cult.cu
- www.lajiribilla.cu
- www.eltrueno.cu
- www.granma.cu
- www.caimanbarbudo.cu
- www.cubasi.cu
- www.juventudrebeld.cu
- www.cualquierotro.cu
- www.otromas.cu

¿Quiénes somos?

La Papeleta es una cartelera desarrollada por el **Ministerio de Cultura** que intenta cubrir todas las actividades culturales que se realizan sobre el territorio cubano. Usted puede **contactarnos a través de nuestro formulario** o puede **escribirnos directamente a webmaster@lapapeleta.co.cu**.

Webmaster: Dunieska Volodia Asimov
Diseño: Eddie Leo Padrón :-)
Edición: Francisco González Leyva
Redacción: Wendy López López

Copyright La Papeleta 2008. Todos los derechos reservados.

Capítulo3: Análisis y diseño del sistema

En este capítulo se realizó el análisis del sistema como lo propone RUP obteniendo como resultado los diagramas de análisis y los de interacción para luego pasar al modelo de diseño mediante el cual se obtuvieron los diagramas de clases del diseño. Además se diseñó la base de datos del sistema.

Conclusiones

Con este trabajo de diploma se concluye que:

- Las tecnologías analizadas son las más indicadas en el desarrollo del portal seleccionando a Drupal como sistema gestor de contenido debido a su alto nivel de flexibilidad y configurabilidad, PHP como lenguaje de programación por ser compatible con Drupal y especialmente con MySQL además de la capacidad que tiene para aumentar su potencial debido a la cantidad de módulos que se le pueden insertar. MySQL como gestor de BD por ser ligero y adaptarse a las especificaciones del sistema y como metodología de desarrollo a utilizar RUP ya que es configurable y puede ser utilizado como una metodología ágil.
- Los módulos diseñados responden a las funcionalidades del portal, ellos son:
 - Módulo que permitirá realizar tanto búsquedas simples como avanzadas al usuario.
 - Módulo que permitirá al Webmaster gestionar la información que va a ser publicada en el home,
 - Módulo para generar y enviar al buzón del usuario boletines electrónicos
 - Módulo para la gestión de la información que se encontraba de manera parcial en la BD del sistema nacional de programación cultural.
- El análisis y diseño realizado para desarrollar una aplicación Web soluciona el problema de promover las actividades culturales de nuestro país a través de la Web y gestionar la información que necesita el MINCULT para llevar a cabo esta tarea; posibilitando una mejor comprensión de la aplicación para la implementación y quedando listo para la próxima fase que propone RUP.

Recomendaciones

A continuación se hacen recomendaciones para mejorar el funcionamiento y los servicios del portal:

- Hacer uso de la tecnología de los servicios Web en el Sistema Nacional de Programación Cultural para lograr una comunicación entre el mismo y el portal.
- Agregarle en futuras versiones carteleras que abarquen también actividades culturales que se realizan en el sector turístico.

Referencias Bibliográficas

JACOBSON, I.; BOOCH, G. y RUMBAUGH, J. *El proceso unificado de desarrollo de software*. 2000, Disponible en: <http://bibliodoc.uci.cu/pdf/reg00060.pdf>.

LARMAN, C. *Uml y patrones*.1999, Disponible en:<http://bibliodoc.uci.cu/pdf/reg00061.pdf>.

Bibliografía Consultada:

ALVAREZ, R. *Prólogo al manual de HTML*, 2006. [Disponible en:

<http://www.desarrolloweb.com/articulos/533.php>

APCNOTICIAS, F. P. *Las ONG le toman el gusto a Drupal*, 2006. [Disponible en:

<http://www.apc.org/es/news/las-ong-le-toman-el-gusto-drupal>

CARRERO, R. G. *10 razones para usar CSS*, 2007. [Disponible en: 10 razones para usar CSS

CMS Matrix. 2008. [Disponible en: [http://cmsmatrix.org/matrix/cms-](http://cmsmatrix.org/matrix/cms-matrix?func=viewDetail;listingId=1050)

[matrix?func=viewDetail;listingId=1050](http://cmsmatrix.org/matrix/cms-matrix?func=viewDetail;listingId=1050)

Conozca más sobre la tecnología Java. 2008. [Disponible en:

<http://www.java.com/es/about/>

Documentación de Python en castellano. 2007. [Disponible en:

<http://www.rzw.com.ar/seguridad-informatica-3101.html>

Drupal. 2008. [Disponible en: [http://cmsmatrix.org/matrix/cms-](http://cmsmatrix.org/matrix/cms-matrix?func=viewDetail;listingId=1050)

[matrix?func=viewDetail;listingId=1050](http://cmsmatrix.org/matrix/cms-matrix?func=viewDetail;listingId=1050)

Drupal Wins Overall 2007 Open Source CMS Award. 2007. [Disponible en:

<http://www.packtpub.com/article/drupal-wins-overall-2007-open-source-cms-award>

Drupal, premio al mejor CMS Open Source del 2007. 2007. [Disponible en:

<http://bitelia.com/2007/11/02/drupal-premio-al-mejor-cms-open-source-del-2007/>

El rol de analista en Rup. 2007. [Disponible en: [http://hancocchi.net/el-rol-del-analista-](http://hancocchi.net/el-rol-del-analista-en-rup/)

[en-rup/](http://hancocchi.net/el-rol-del-analista-en-rup/)

GÓMEZ, J. C. G. and T. S. PÉREZ. *Los portales en Internet.*, Universidad de Murcia,

2007. [Disponible en: <http://www.um.es/gtiweb/curso/seis.htm>

HIDALGO, A. F. P. *DESCRIPCIÓN DE LOS ASPECTOS FUNDAMENTALES DEL LENGUAJE DE*

PROGRAMACIÓN PERL, 2007.

- HINOSTROZA, R. R. *Características de PHP*, 2007 [Disponible en: <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>
- HTML*. 2008. [Disponible en: http://es.wikipedia.org/wiki/C%C3%B3digo_HTML
- Introducción a la Ingeniería de Software* 2007. [Disponible en: <http://oacosta334.blogspot.es/tags/RUP/>
- Joomla!* , 2008. [Disponible en: <http://es.wikipedia.org/wiki/Joomla>
- Lenguaje de programación Java*. 2008. [Disponible en: http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java
- Mambo*. 2008. [Disponible en: <http://cmsmatrix.org/matrix/cms-matrix?func=viewDetail;listingId=1074>
- MENDOZA, I. J. A. *Los Portales, una nueva dimensión en Internet*, 2000. [Disponible en: <http://www.informaticamilenium.com.mx/Paginas/mn/articulo25.htm>
- Moodle*. 2008. [Disponible en: <http://cmsmatrix.org/matrix/cms-matrix?func=viewDetail;listingId=1171>
- MySQL Licensing Policy*. 2008. [Disponible en: <http://www.mysql.com/about/legal/licensing/>
- Otras características de Umbrello UML Modeller*. 2008. [Disponible en: <http://docs.kde.org/kde3/es/kdesdk/umbrello/other-features.html>
- Perl*. 2008. [Disponible en: <http://es.wikipedia.org/wiki/Perl>
- PHP*. 2008. [Disponible en: <http://es.wikipedia.org/wiki/.php>
- Portal vs. sitio web. Breve historia de los portales*. 2007. [Disponible en: <http://www.realidadaparte.es/archives/2007/08/18/portal-vs-sitio-web-breve-historia-de-los-portales-breve-historia-de-internet/>
- Portal(internet)*. 2008. [Disponible en: [http://es.wikipedia.org/wiki/Portal_\(Internet\)](http://es.wikipedia.org/wiki/Portal_(Internet))
- Proceso Unificado de Rational*. 2008. [Disponible en: http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational
- Python*. 2008. [Disponible en: <http://es.wikipedia.org/wiki/Python>

Qué es Mambo? . 2006. [Disponible en: <http://www.portalmambo.com/que-es-mambo-.html>]

¿Qué es MySQL? . 2003. [Disponible en: <http://www.mysql-hispano.org/page.php?id=2>]

¿Que es un Portal? , 2006. [Disponible en:

http://eca.europa.eu/portalHelp2/ohw/state/content/locale.es/vtTopicFile.welchelp_hs_es%7Cwelcport~htm/navId.3/navSetId. /]

¿Qué es un Sitio web y Página web? - Definición de Sitio web y Página web. 2006.

[Disponible en: <http://www.masadelante.com/faq-sitio-web.htm>]

VALDÉS, D. P. *¿Qué es Javascript?*, 2007. [Disponible en:

<http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>]

Ventajas y características de Joomla CMS. 2007. [Disponible en:

<http://www.joomlaos.net/caracteristicas-de-joomla.php>]

VIGNAGA, A. *Transformación de un Modelo de Dominio y Diagramas de Comunicación en un Diagrama de Clases de Diseño*, 2006.

Visual Paradigm for UML. 2008. [Disponible en: [http://www.visual-](http://www.visual-paradigm.com/documentation/)

[paradigm.com/documentation/](http://www.visual-paradigm.com/documentation/)]

Visual Paradigm for UML (ME) 2007. [Disponible en:

[http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/)]

Visual Paradigm UML Community. 2008. [Disponible en:

<http://ceds.nauta.es/informes/case04.htm>]

ZAYAS, C. A. D. *Metodología de la investigación científica*, 1995.

Glosario de Términos

MINCULT: Ministerio de Cultura de Cuba, es el organismo que atiende todo lo que tiene que ver con el quehacer cultural de nuestro país.

CUBARTE: Departamento del Ministerio de cultura encargado de promocionar las actividades culturales

Webmaster: Es el administrador del Portal.

Portal: Sitio Web que contiene la información referente a la Programación cultural del MINCULT.

Instituciones: Responde tanto a las de carácter nacional como la de CREART, otras más generales como la Jiribilla, Cubaescena, Cubacine así como las provinciales y municipales.

BD: Base de datos.

SGBD: Sistema gestor de base de datos.

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.

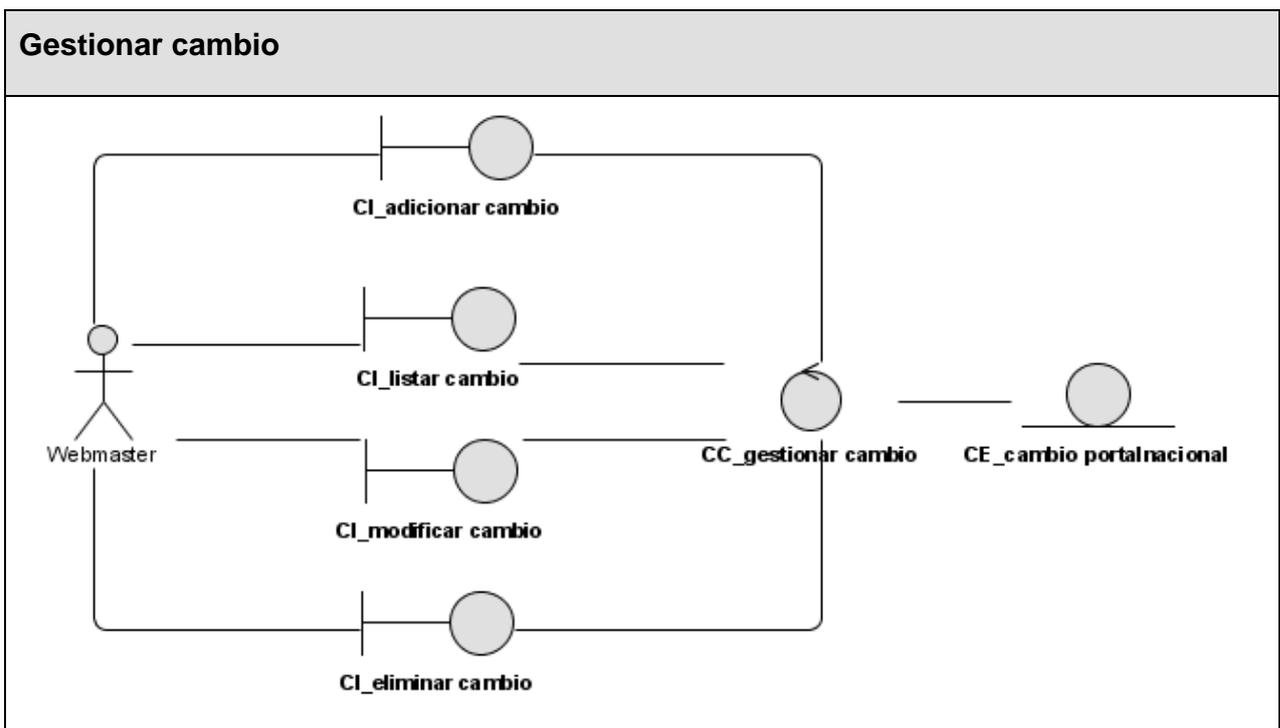
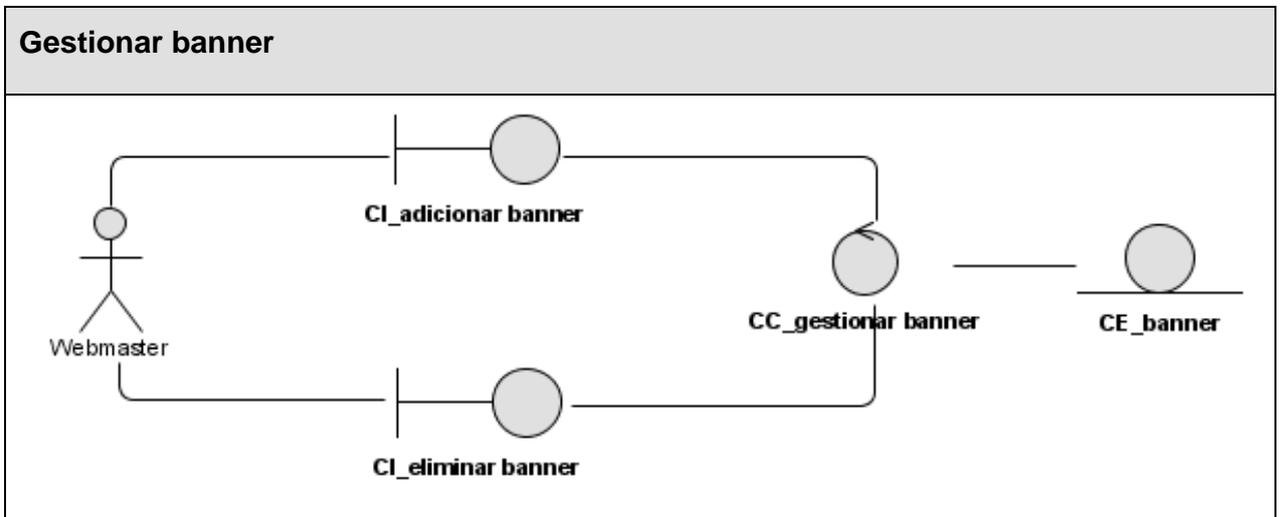
ADOdb: Librería de PHP para acceso a distintos gestores de bases de datos.

XML: Metalenguaje capaz de definir sus propias etiquetas.

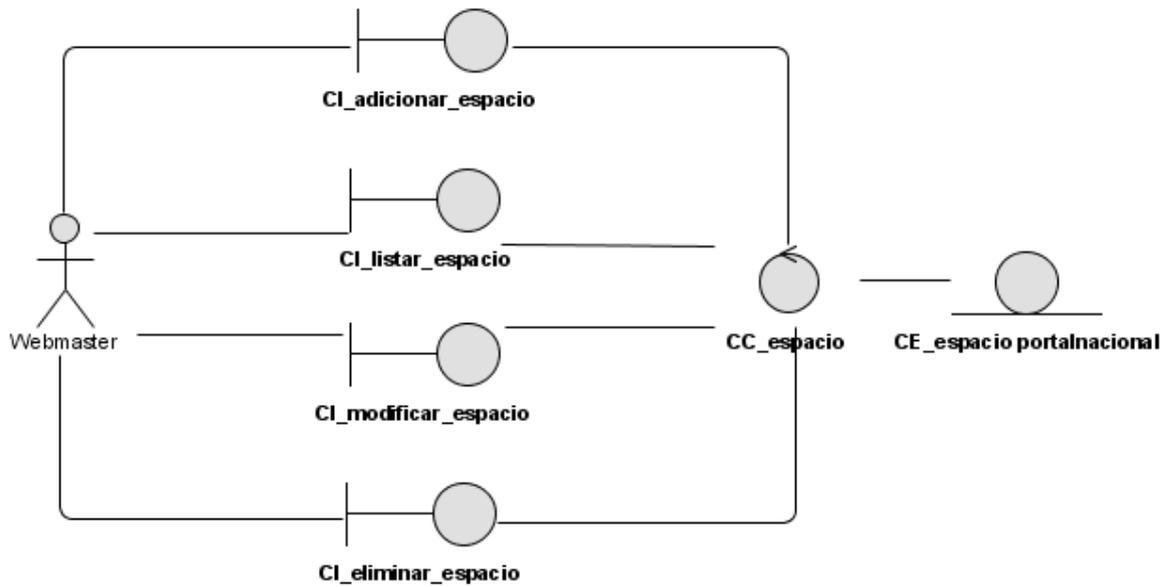
CSS: Hojas de estilos para darle al HTML el aspecto deseado.

Anexos

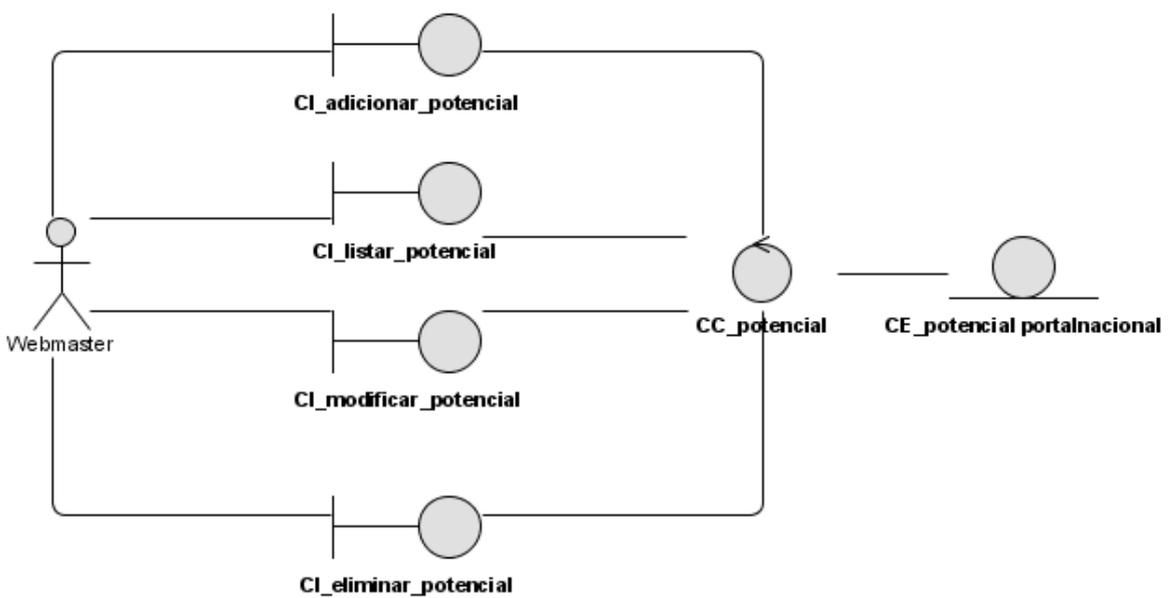
Anexo 1. Diagramas de clases de análisis

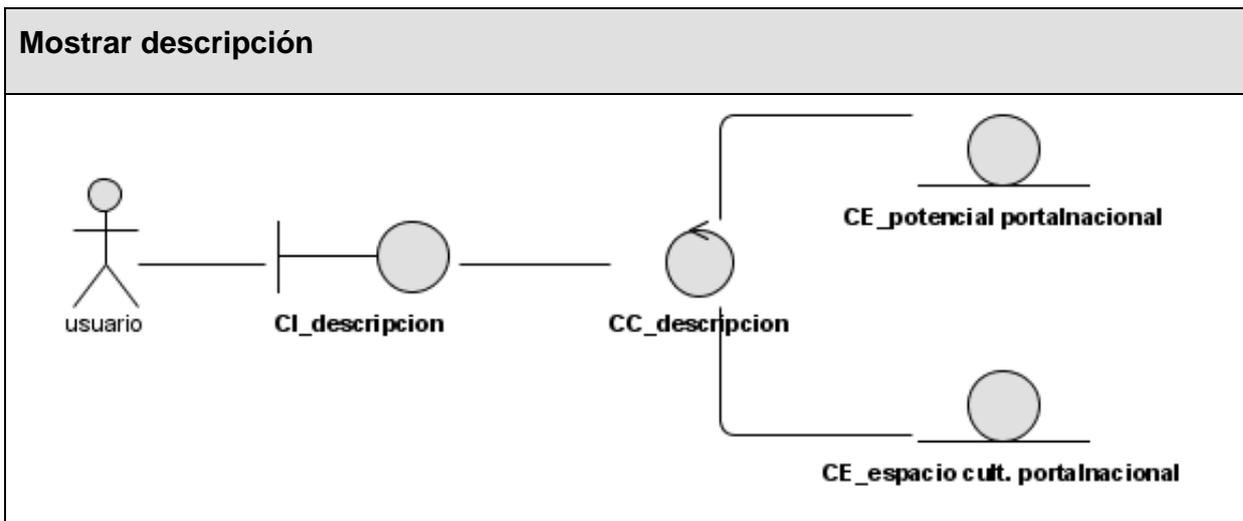


Gestionar espacio cultural

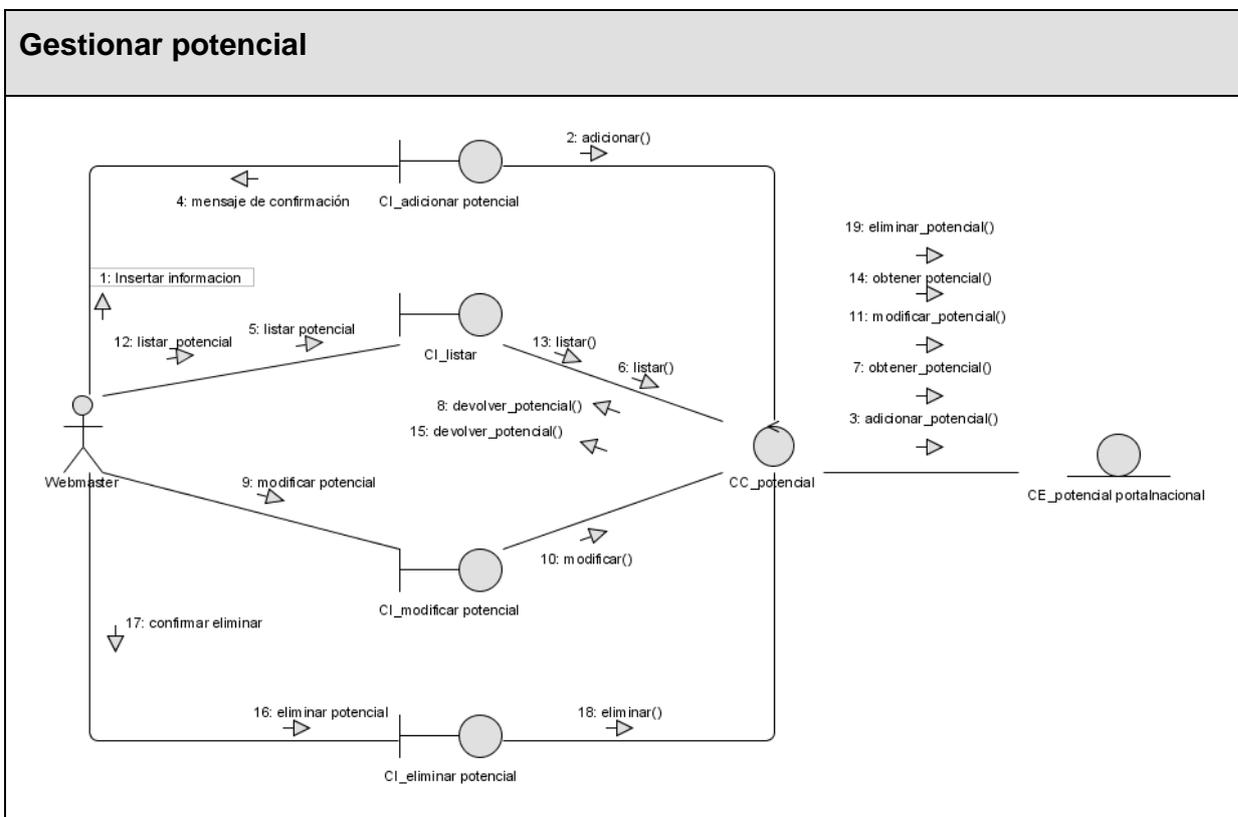


Gestionar potencial

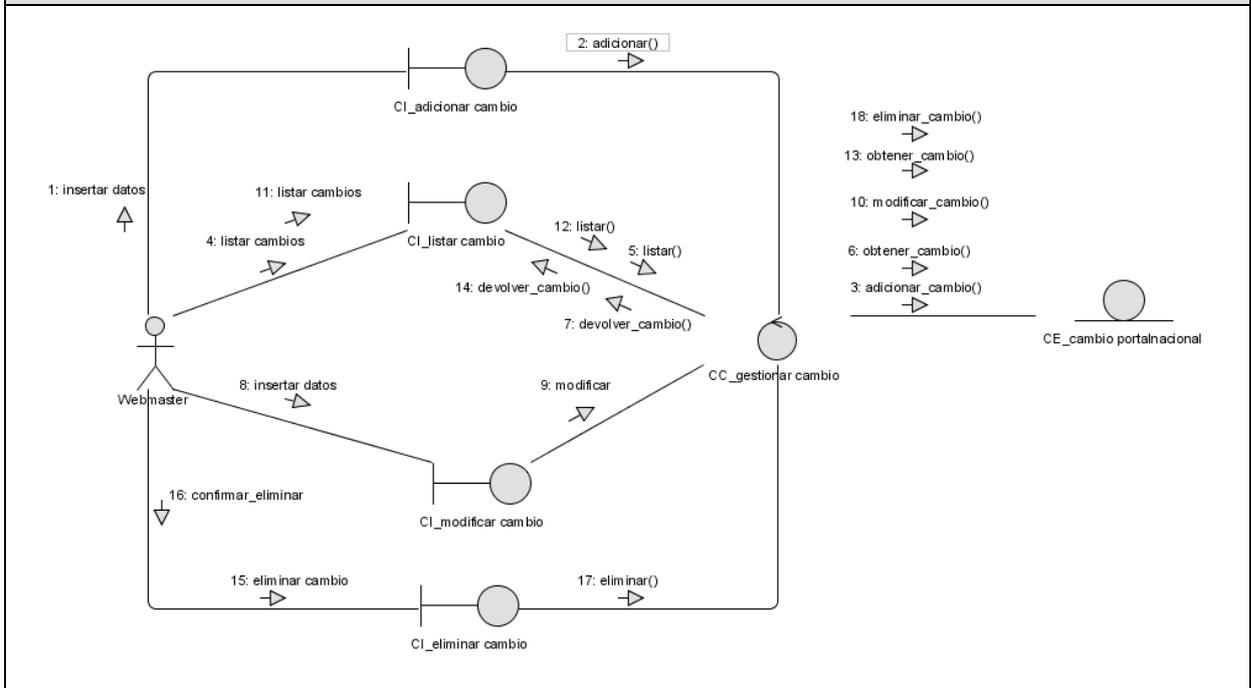




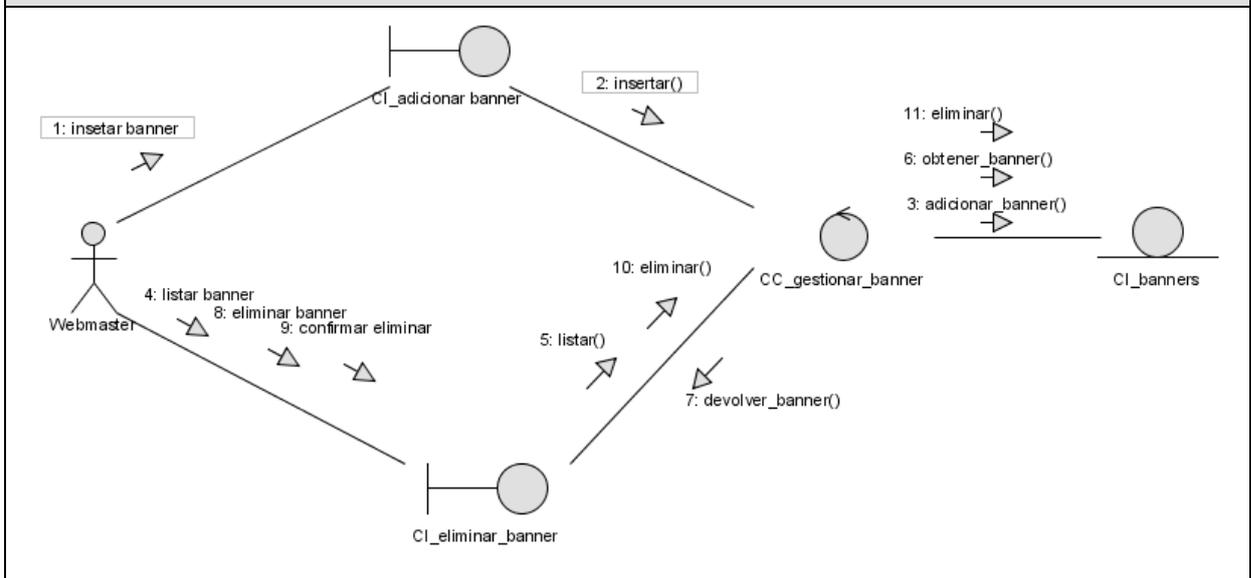
Anexo 2. Diagramas de colaboración



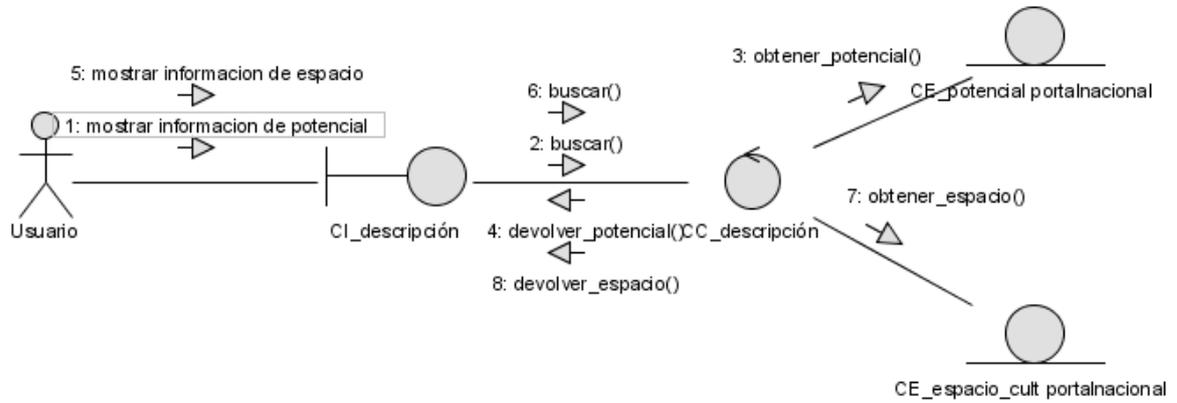
Gestionar cambio



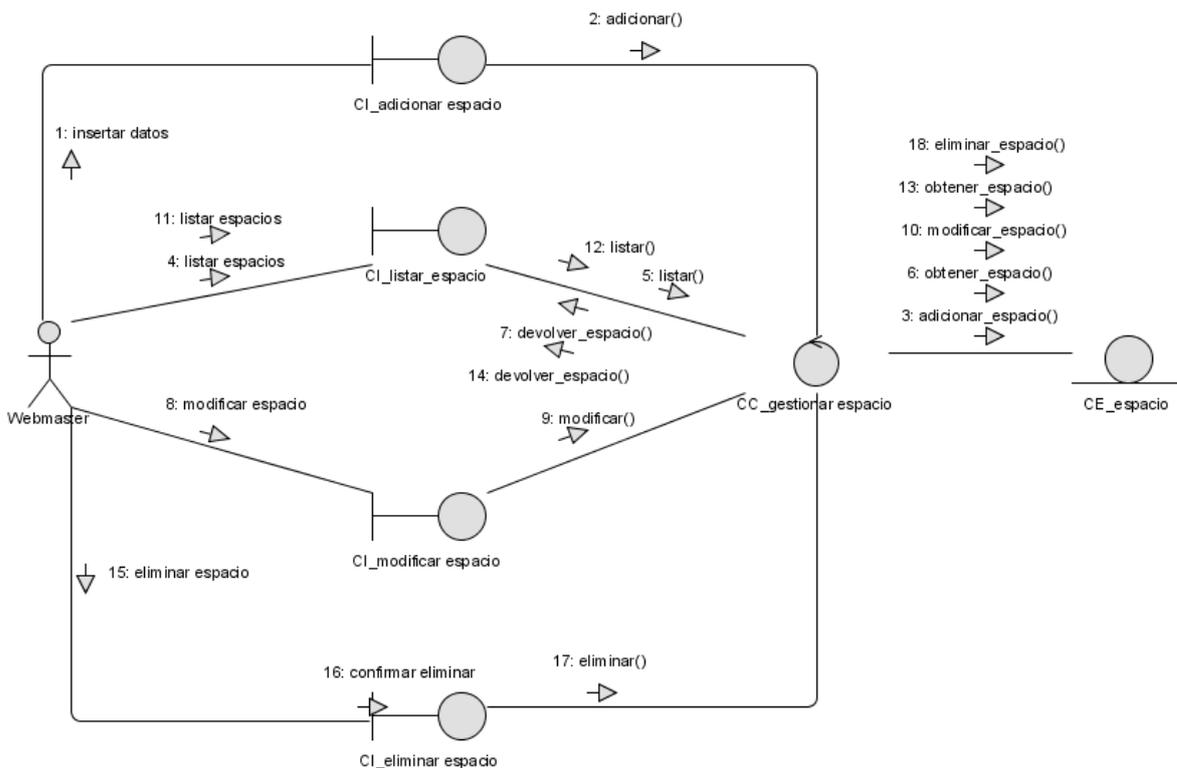
Gestionar banner



Mostrar descripción



Gestionar espacio cultural



Anexo 3. Diagramas de clases del diseño

Diagrama de clases del diseño. Gestionar cambio. Paquete de administración

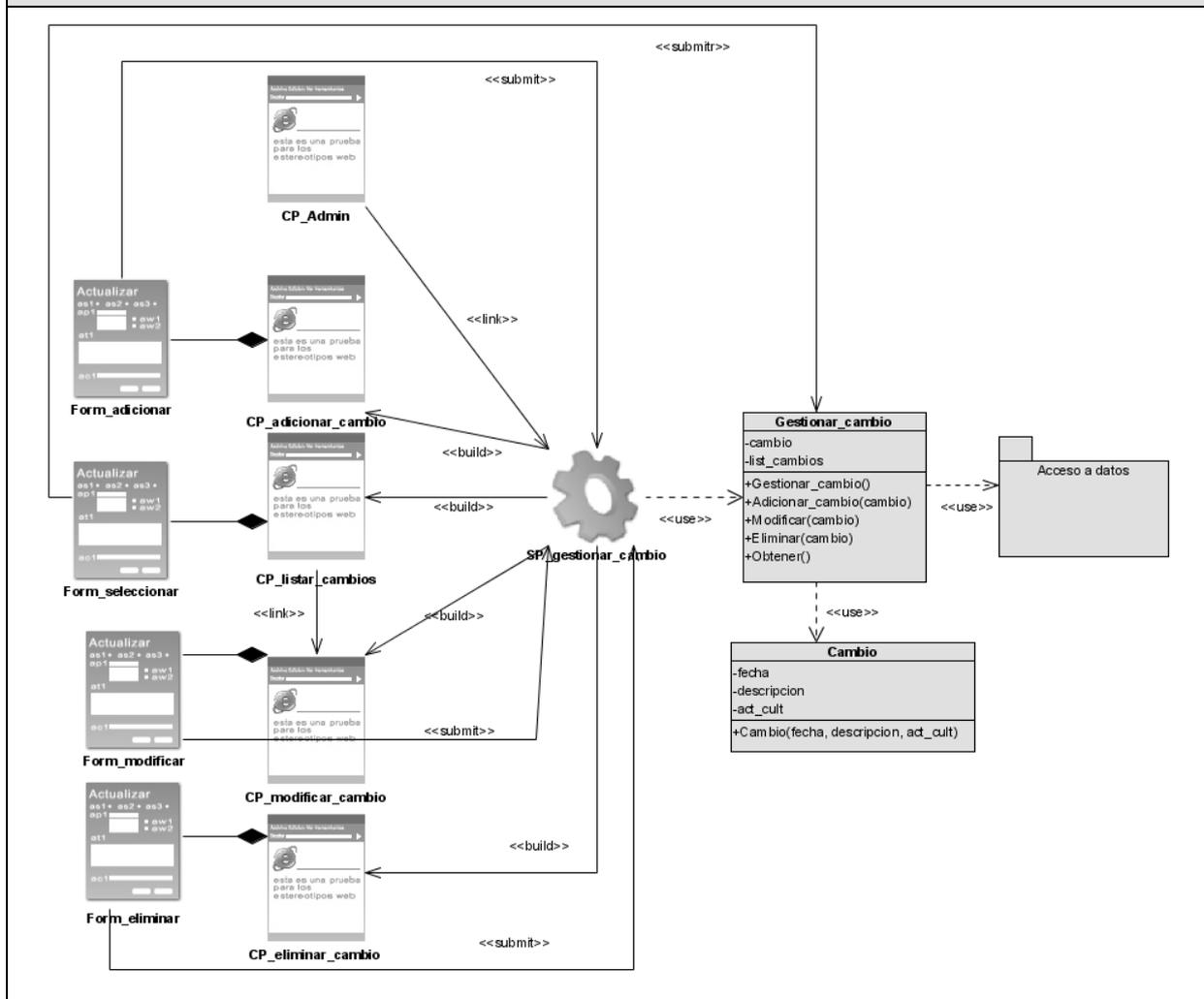


Diagrama de clases del diseño. Gestionar banner. Paquete de administración

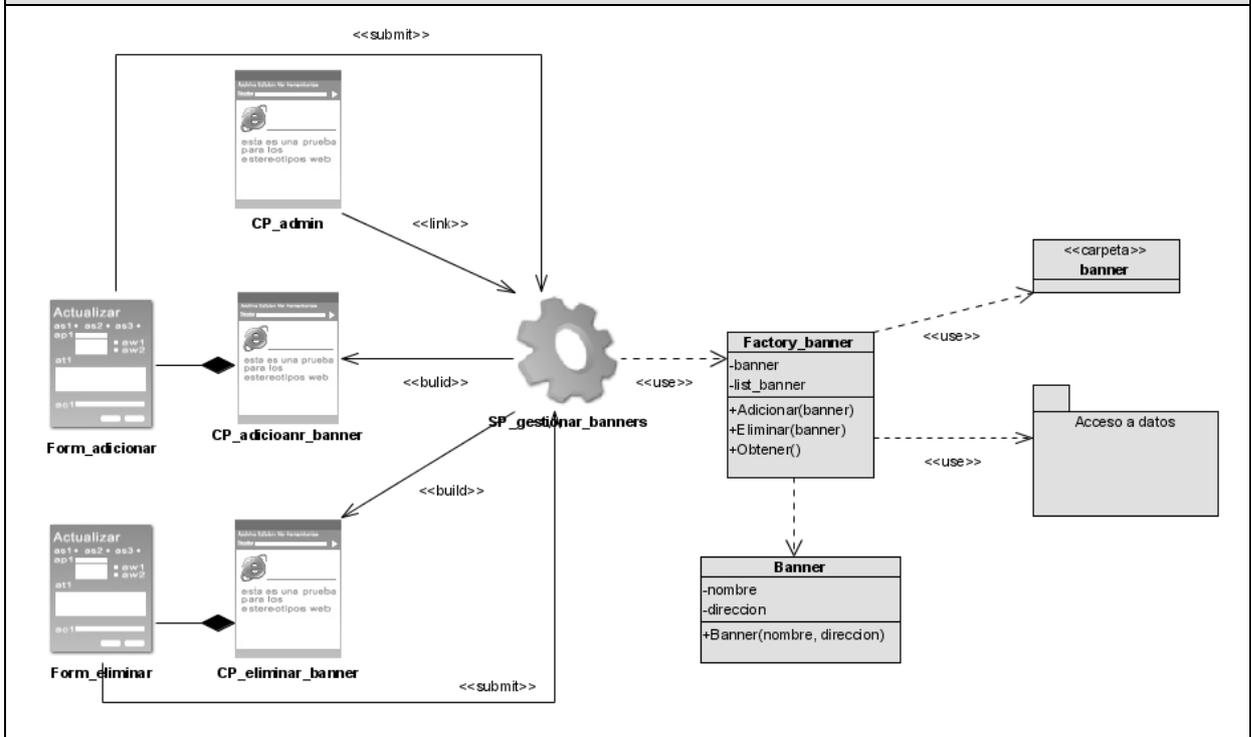


Diagrama de clases del diseño. Mostrar descripción. Paquete de información

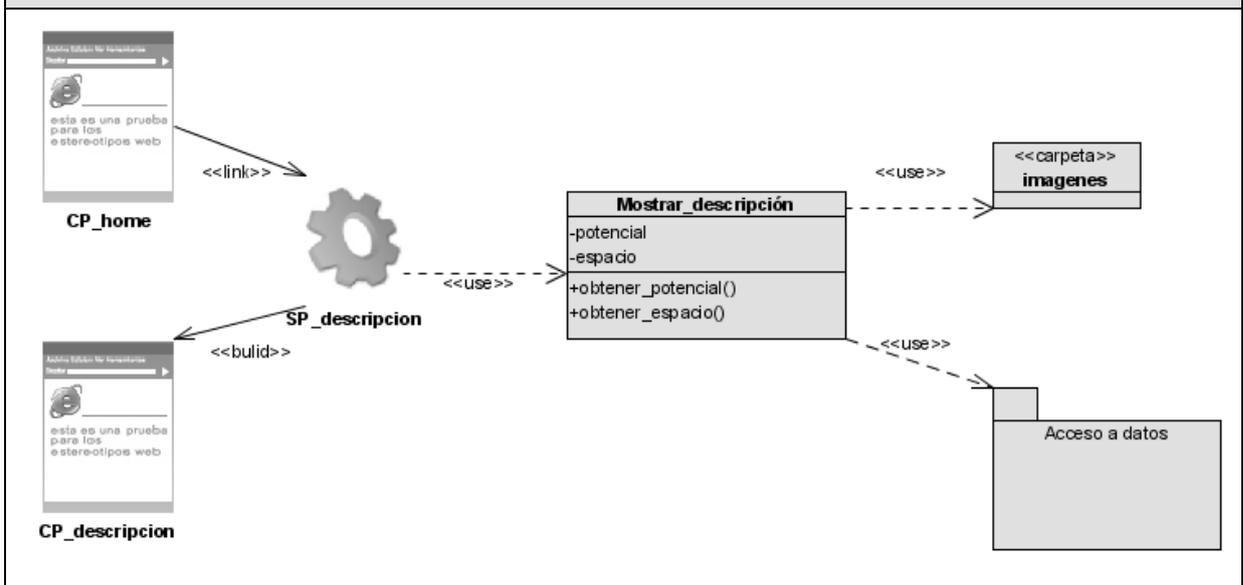


Diagrama de clases del diseño. Gestionar espacios culturales. Paquete de administración

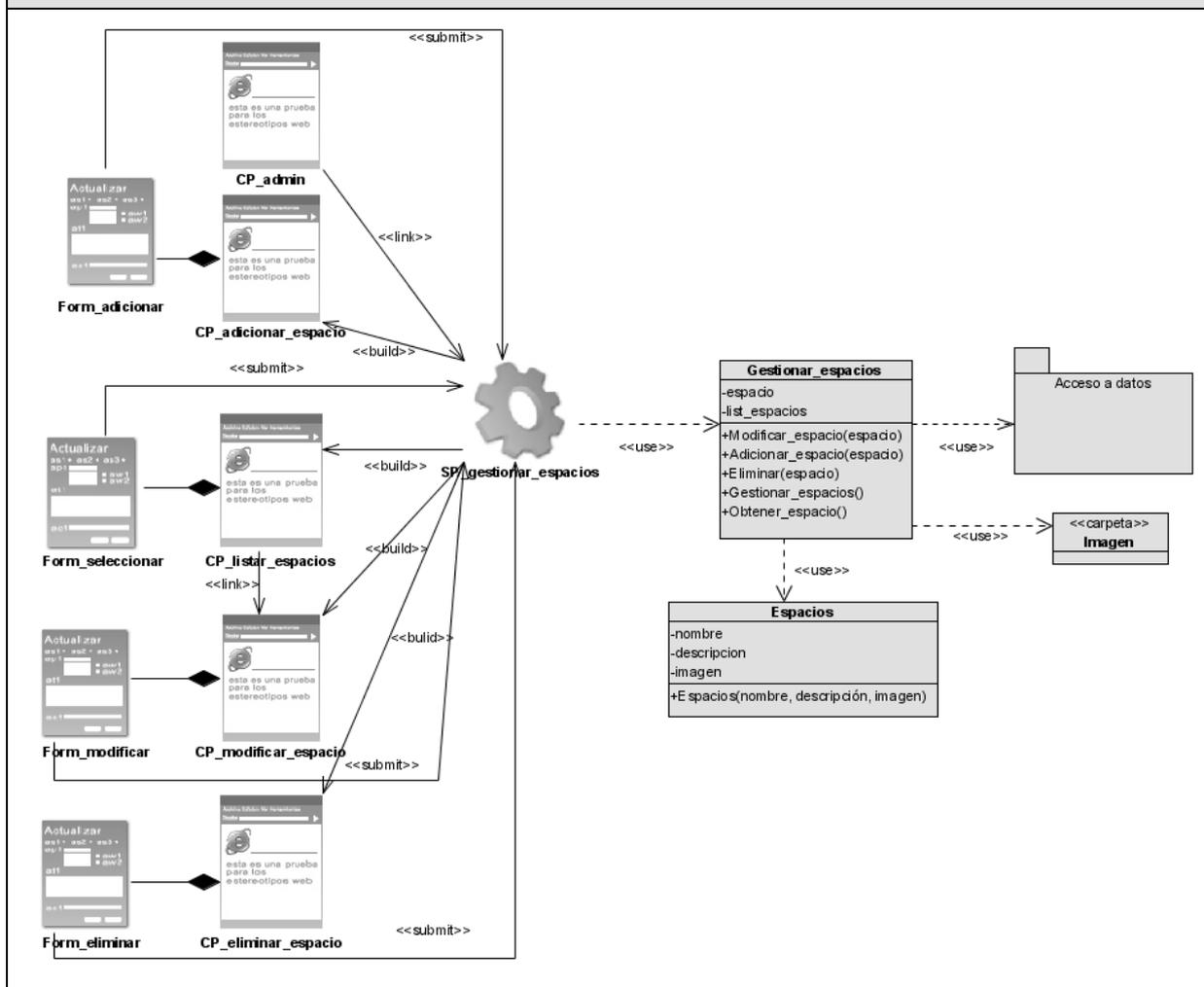


Diagrama de clases del diseño. Gestionar potencial artístico. Paquete de administración

