



*Universidad de las Ciencias Informáticas, Facultad 1*

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

*Título: Desarrollo de un Analizador de Registros Web (ARWeb).*

*Autor(es): Rosalí Roche Tejeda.*

*Carlos Manuel Estrada Fernández.*

*Tutor(es): Ing. Maílín Diéguez Pavón.*

*Ing. Jorge Luis Betancourt.*

*Ing. Ariel Reyes Antuán.*

*La Habana, junio 2014  
"Año 56 de la Revolución"*

## DECLARACIÓN DE AUTORÍA

Se declara que somos los únicos autores de este trabajo y autorizamos a la Facultad 1 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste se firma la presente declaración jurada de auditoría en La Habana a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

### Autor(es):

\_\_\_\_\_  
Rosalí Roche Tejeda

\_\_\_\_\_  
Carlos Manuel Estrada Fernández

### Tutores:

\_\_\_\_\_  
Ing. Mailin Diéguez Pavón

\_\_\_\_\_  
Ing. Jorge Luis Betancourt

\_\_\_\_\_  
Ing. Ariel Reyes Antuán

## **DATOS DE CONTACTO**

- **Tutor:** Ing. Mailin Dieguez Pavón. Ingeniero en Ciencias Informáticas, UCI 2010. Especialista del centro de producción CIDI, Facultad 1.

Correo electrónico: [mdiequez@uci.cu](mailto:mdiequez@uci.cu) . Teléfono: 837 2353.

- **Tutor:** Ing. Jorge Luis Betancourt González. Ingeniero en Ciencias Informáticas, UCI 2012. Desarrollador del centro de producción CIDI, Facultad 1.

Correo electrónico: [jlbetancourt@uci.cu](mailto:jlbetancourt@uci.cu) . Teléfono: 837 3388.

- **Tutor:** Ing. Ariel Reyes Antuán.

Correo electrónico: [aareyes@uci.cu](mailto:aareyes@uci.cu) . Teléfono: 837 3173.



*“La memoria es la inteligencia de los tontos.”*

*Albert Einstein.*

**AGRADECIMIENTOS**

Rosalí

*A mi papá por confiar en mí y apoyarme en todo.*

*A mi hermana Caridad por ser el motor impulsor de esta vida.*

*A mi familia de forma general por su apoyo y comprensión.*

*A mi tutora Mailín, Jorge y Ariel por guiarme en todo momento.*

*A mi piquete que siempre me brindaron lo mejor en todos estos años de la carrera.*

*A Joel por brindarme todo su apoyo y consideración en este tiempo juntos.*

*A mi compañero de tesis por apoyarme y ayudarme cuando más lo necesité.*

Carlos Manuel

*A mis compañeros de apartamento por soportarme en estos 5 años.*

*A mis compañeros de aula y de laboratorio.*

*A mis tutores.*

*A todos los que de una forma u otra me ayudaron en la realización de este trabajo.*

DEDICATORIA

Rosali

*A mi hermana Cari por ser la luz de mis ojos.*

*A mi ahijada Stefany por ser lo mejor que me ha pasado en la vida después de mi bebé.*

*A mi abuela María que es más que una madre para mí.*

Carlos Manuel

*A mi padre, por ser mi ejemplo a seguir en esta vida.*

*A mi madre, por ser simplemente lo mejor de este mundo.*

*A mi hermana, que sin ella estoy literalmente incompleto.*

*En especial a mima, mi abuela que con tanto cariño me crio; te quiero no te imaginas cuánto.*

## Resumen

Los sistemas analizadores de registros web son medios informáticos utilizados por algunas empresas u organizaciones que han actualizado su accionar ubicándose en la red. Estos sistemas brindan una serie de reportes estadísticos con el objetivo de visualizar información importante y necesaria en la toma de decisiones, que influyen directamente en el correcto desempeño de las mismas. Con los avances alcanzados por Cuba en el uso de las Tecnologías de la Información y las Comunicaciones (TICs) crea un ambiente favorable para la utilización de herramientas de este tipo, por lo que la presente investigación tiene como finalidad desarrollar un sistema analizador de registros web para apoyar el proceso de toma de decisiones en empresas y organizaciones cubanas.

Para el desarrollo de la propuesta de solución se utilizaron los *frameworks* *Symfony2* en su versión 2.3.7 para el desarrollo y *Bootstrap* para el diseño de las interfaces de usuario, *PostgreSQL* versión 9.1 como Sistema Gestor de Bases de Datos para los datos administrativos y *Elasticsearch* como base de datos para los datos estadísticos recogidos. Además, se expone el proceso de desarrollo siguiendo la metodología *OpenUp* a partir de la cual se generaron un conjunto de artefactos que capturan la información relacionada con la solución. Para verificar el cumplimiento de las funcionalidades propuestas se realizaron varias pruebas al *software*. Como resultado se obtuvo un producto funcional con una documentación que sirve de base para futuras investigaciones o modificaciones a la propuesta de solución.

**Palabras Clave:** Analítica Web, internet, registro web, toma de decisiones, navegación.

## Índice

Introducción.....	1
<b>Capítulo 1: Marco Teórico Conceptual .....</b>	<b>6</b>
1.1 Introducción.....	6
1.2 Conceptos asociados al dominio del problema .....	6
1.3 Estado del arte.....	7
1.3.1 Marco nacional.....	8
1.3.2 Marco internacional.....	8
1.3.3 Análisis crítico de los sistemas estudiados.....	10
1.4 Metodología de desarrollo de software .....	10
1.4.1 <i>OpenUP</i> .....	11
1.5 Análisis del soporte tecnológico para el desarrollo del <i>software</i> .....	12
1.5.1 Lenguajes de programación .....	12
1.5.2 Lenguajes de enmarcado .....	13
1.5.3 Servidor Web.....	14
1.5.4 Sistema Gestor de Base de Datos .....	14
1.5.5 <i>Framework</i> de Desarrollo .....	15
1.5.6 <i>Framework</i> de Diseño .....	16
1.5.7 Librería para graficar.....	16
1.5.8 Entornos de desarrollos Integrados ( <i>IDE por sus siglas en inglés</i> ).....	17
1.5.9 Lenguaje de Modelado.....	17
1.5.10 Herramienta de ingeniería de software asistida por computación.....	18
1.5.11 Herramienta adicional.....	18
1.6 Conclusiones.....	19
<b>Capítulo 2: Diseño e Implementación del Sistema.....</b>	<b>20</b>
2.1 Introducción.....	20
2.2 Propuesta del sistema .....	20
2.2.1 Descripción de actores.....	20
2.3 Modelo de dominio.....	21
2.3.1 Descripción de clases del modelo del dominio .....	21



2.4 Modelado del sistema .....	22
2.4.1 Técnicas de obtención de requisitos .....	22
2.4.2 Requisitos funcionales (RF) .....	22
2.4.3 Requisitos no funcionales (RNF) .....	24
2.5 Diagrama de casos de uso (CU) del sistema .....	26
2.5.1 Especificación de CU .....	26
2.6. Descripción del sistema propuesto .....	33
2.6.1 Patrones de diseño implementados .....	33
2.6.2 Patrón arquitectónico implementado por <i>Symfony2</i> .....	35
2.6.3 Estructura en <i>bundles</i> de la aplicación .....	36
2.6.4 Seguridad en el sistema .....	37
2.7 Diagramas de clases del diseño .....	37
2.8 Diagramas de interacción .....	39
2.9 Modelo de datos .....	41
2.10 Modelo de implementación .....	42
2.10.1 Diagrama de componente .....	42
2.10.2 Diagrama de despliegue .....	43
2.11 Estándares de codificación .....	44
2.12 Conclusiones .....	46
Capítulo 3: Validación y Prueba del Sistema .....	47
3.1. Introducción .....	47
3.2. Pruebas funcionales .....	47
3.3. Pruebas de carga y estrés .....	49
3.4. Pruebas de seguridad .....	50
3.5. Conclusiones .....	56
Conclusiones .....	58
Recomendaciones .....	59
Bibliografía referenciada .....	60
Bibliografía consultada .....	63

<b>Anexos</b> .....	68
<b>Anexo A</b> .....	68
<b>Anexo B</b> .....	68
<b>Anexo C</b> .....	89
<b>Anexo D</b> .....	97
<b>Anexo E</b> .....	102

**ÍNDICE DE TABLAS**

Tabla 1: Definición de los actores..... 21

Tabla 2: CU Generar reportes..... 28

Tabla 3: CU Gestionar sitio. .... 33

Tabla 4: Descripción de las variables. .... 47

Tabla 5: Prueba funcional para el CU gestionar sitio, sección: Nuevo. .... 48

Tabla 6: Resultados de las pruebas con jMeter. .... 50

Tabla 7: Lista de chequeo prueba de autorización..... 51

Tabla 8: Lista de chequeo prueba de gestión de sesión. .... 52

Tabla 9: Lista de chequeo comprobación del sistema de autenticación. .... 54

Tabla 10: Lista de chequeo validación de datos..... 55

**ÍNDICE DE FIGURAS**

Figura 1: Modelo de dominio..... 21

Figura 2: Diagrama de casos de uso del sistema..... 26

Figura 3: Diagrama del patrón arquitectónico MVC..... 36

Figura 4: DCDEW para CU Generar reportes ..... 38

Figura 5: DCDEW para CU Gestionar sitio..... 38

Figura 6: Diagrama de secuencia para CU Generar reportes..... 39

Figura 7: Diagrama de secuencia para CU Gestionar sitio ..... 40

Figura 8: Modelo de datos para datos administrativos..... 41

Figura 9: Modelo de datos para datos estadísticos..... 42

Figura 10: Diagrama de componente general. .... 43

Figura 11: Diagrama de despliegue..... 44

Figura 12: Nombre de las clases, *namespace* y *use*..... 45

Figura 13: Nombre las funciones en *Lower Camel Case*. .... 45

Figura 14: Atributos en *Lower Camel Case*. .... 46

Figura 15: Resultados de las pruebas funcionales..... 49

Figura 16: Resultados de las pruebas de seguridad con *Acunetix*. .... 56

## Introducción

Internet también conocida como “red de redes”, es una enorme red que conecta computadoras distribuidas por todo el mundo, permitiendo comunicación, búsqueda y transferencia de información sin grandes requerimientos tecnológicos ni económicos relativos para el individuo (Vea, 2013). Una forma moderna y rápida de estos intercambios de datos se realiza mediante los sitios web, que no son más que un conjunto organizado y coherente de páginas web conectadas entre ellas por vínculos y que muestran de forma sencilla textos, imágenes, videos, animaciones, sonidos, y otras formas de representar información (Ganazhapa, 2012).

En los últimos años el uso de internet se ha generalizado enormemente dentro de la población mundial, ampliándose los mercados y las áreas de interés tradicionales al mundo virtual. Esta circunstancia ha propiciado un incremento de la competitividad entre empresas que tienen su representación en la red de redes y una búsqueda de métodos más eficaces para llegar al usuario final. Estas empresas e instituciones se enfrentan a nuevos comportamientos de acceso a la información cada vez más dinámicos e inciertos, caracterizados por generar grandes volúmenes de datos, que una vez analizados pueden ser útiles para la organización.

En estos tiempos es imprescindible para empresas que compitan por la excelencia de sus servicios poder llegar a todos los usuarios y clientes potenciales, sin importar limitaciones de tiempo, distancia o físicas de estos, internet, aunque es la forma ideal para llegar a todo el mundo, no está exenta de obstáculos, muchos derivados de malos diseños tanto técnicos como gráficos de las páginas web, es en este punto donde la Analítica Web desempeña un rol importante, la misma puede ser definida como la disciplina profesional encaminada a extraer conclusiones, definir estrategias o establecer reglas de negocio sobre la base de los datos recabados en todos aquellos entornos web sobre los que una empresa ejerce control (Maldonado, 2012).

La historia se remonta a la década de los noventa del siglo pasado, específicamente en Estados Unidos, donde surgieron las primeras herramientas de Analítica Web. En aquel entonces los esfuerzos de las compañías proveedoras de este tipo de herramientas, se concentraban en el análisis de los archivos “log<sup>1</sup>” generados por los servidores de páginas web. Estos archivos, con estructura de texto plano, recogen importantes variables acerca de la interacción de un visitante con la página web, como por ejemplo páginas vistas, número de *hits*<sup>2</sup>, errores 404, entre otras. Sin embargo, a pesar del valor de esta información, en los

---

<sup>1</sup> Un log es un registro oficial de eventos durante un rango de tiempo en particular.

<sup>2</sup> Un hit es cada una de las peticiones a un servidor por un archivo o página web.

primeros años del siglo veintiuno, los diferentes departamentos integrantes de las organizaciones empezaron a demandar más información, naciendo el análisis en tiempo real, y el etiquetado de las páginas (Maldonado, 2012).

Teniendo en cuenta la evolución de la Analítica Web, la misma se presenta como una vía que permita a las entidades tomar decisiones de *marketing*<sup>3</sup> más eficientes, mejorar su estrategia de posicionamiento en la red, incrementar su cartera de clientes y adecuar el diseño de su página web a los intereses colectivos de los mismos, además de permitir la evaluación en tiempo real del retorno económico de todas estas actividades. Así, los *marketers* (vendedores) pueden conocer mejor a su público objetivo y determinar aquellos segmentos o individuos del mercado más rentables; decidir si la web de la empresa es susceptible de recibir más inversiones; detectar oportunidades de mejora o de negocio; comprender mejor los indicadores del rendimiento económico; averiguar qué productos son más rentables; hacer un adecuado seguimiento de clientes, tanto actuales como potenciales; averiguar la procedencia y destino de los visitantes; aumentar la fidelidad de sus clientes, en definitiva, gestionar todos los aspectos relevantes de su negocio en internet.

A pesar de que Cuba no esté a la par de los países punteros en el uso de las TICs, se puede destacar un incremento en los últimos cinco años, propiciando el escenario adecuado hacia la utilización de herramientas de Analítica Web para apoyar el proceso de tomas de decisiones en empresas y organizaciones que hayan actualizado su modelo económico hacia el mundo virtual.

En la actualidad muchas de las empresas cubanas que recurren a la Analítica Web para manejar a su favor toda la información que pueda ser generada por su sitio, lo hace a través de herramientas privativas. Esto trae como consecuencia que dichas entidades deban pagar por los servicios prestados, estando fuera del alcance de la totalidad de las empresas nacionales, también trae aparejado una serie de dificultades como son: la información recolectada es almacenada en servidores fuera del país atentando contra la soberanía tecnológica y que esta pueda ser utilizada con fines ajenos a los intereses de la empresa propietaria; otro inconveniente es que para acceder a estos servicios se debe poseer una conexión a internet, limitando el alcance para organizaciones que cuentan solamente con red nacional.

La UCI como entidad puntera dentro del desarrollo de *software* en la isla, cumpliendo con una de las aristas de su misión que es: servir de soporte a la industria cubana de la informática, y para dar solución al problema planteado anteriormente, desarrolló entre los años 2009 y 2011 una herramienta de Analítica Web llamada

---

<sup>3</sup> Según Philip Kotler, este es un proceso en el cual un grupo de individuos intercambia bienes y servicios para satisfacer sus necesidades

“AiresWeb” (Analizador Inteligente de Registros de Servidores Web). Dicho sistema no brinda reportes en tiempo real, parámetro que debe cumplir hoy en día toda herramienta de Analítica Web por lo importante que es para las organizaciones contar con la información más actualizada posible, siendo esta su principal deficiencia. Esta herramienta fue desarrollada con tecnologías que en el transcurso del tiempo fueron quedando obsoletas e incluso algunos de los medios utilizados en la construcción del sistema no se les brinda soporte, constituyendo un gran inconveniente si se decide agregar nuevas funcionalidades o perfeccionar las existentes.

De la situación problemática anterior se identifica como **problema de investigación**: ¿Cómo monitorizar en tiempo real las actividades que realizan los usuarios en los sitios web cubanos para asistir en la toma de decisiones?

Se precisó como **objeto de estudio**: Los sistemas de análisis de registros web. Identificándose como **campo de acción**: Los sistemas que hacen análisis web para aportar datos a la toma de decisiones. Para dar una solución al problema planteado se define como **objetivo general**: Desarrollar un Sistema Analizador de Registros Web que monitorice en tiempo real las actividades realizadas por los usuarios en los sitios web cubanos para asistir en la toma de decisiones, derivando en los siguientes **objetivos específicos**:

- Definir el marco teórico conceptual y el estado del arte respecto a las tecnologías actuales para los sistemas analizadores de registros web.
- Diseñar un sistema analizador de registros web que permita conocer las trazas<sup>4</sup> que se crean en los sitios web.
- Implementar una herramienta que permita la generación de estadísticas a partir de registros web.
- Validar el correcto funcionamiento del sistema analizador de registros web.

Se plantea como **idea a defender**: Con el desarrollo de un analizador de registros web se contribuirá positivamente al monitoreo de las actividades realizadas por los usuarios en los sitios web cubanos lo que apoyará el proceso de toma de decisiones de las organizaciones.

Con el fin de darle cumplimiento al objetivo general se desglosaron las siguientes **tareas de investigación**:

1. Definición de los conceptos asociados al marco teórico de la investigación.
2. Caracterización de los sistemas analizadores de registros web existentes.

---

<sup>4</sup> Traza: información que se genera acerca de las acciones producidas durante el procesamiento de las páginas web.

3. Identificación de las tecnologías para realizar el análisis de la información registrada durante el acceso de los usuarios al sitio web.
4. Selección de las bibliotecas necesarias para el desarrollo del sistema.
5. Definición de la estructura de base de datos a utilizar.
6. Definición de la arquitectura del sistema.
7. Diseño de la interfaz de usuario del sistema.
8. Desarrollo de las funcionalidades que permitan el análisis de las estadísticas generadas.
9. Validación del sistema implementado, teniendo en cuenta su funcionamiento.

Para asistir y dar alcance a las tareas propuestas se emplearon los siguientes **métodos científicos**:

**Métodos teóricos:**

- Histórico – Lógico: Este método fue de gran importancia en la elaboración de la fundamentación teórica de la investigación, su estudio permitió realizar un análisis profundo de los antecedentes y proyecciones del tema: La Analítica Web.
- Analítico-Sintético: Este método fue utilizado durante todo el proceso investigativo, permitiendo realizar un análisis detallado de la documentación relacionada con el tema de la Analítica Web, su aporte a las entidades y organizaciones, así como determinar las buenas prácticas asociadas al desarrollo de herramientas para el análisis de registros web.

**Métodos empíricos:**

- Entrevista: Este método fue de gran importancia en la obtención de información sobre la herramienta desarrollada en nuestro país, también para interactuar con personal capacitado sobre la Analítica Web en la universidad, con el fin de obtener información e ideas que contribuyan al desarrollo de la investigación.

El contenido de este documento se encuentra estructurado como se muestra a continuación:

**Capítulo 1 “Marco Teórico Conceptual”:** En este capítulo se fundamentan los elementos teóricos necesarios que sustentan la investigación. Se realiza un análisis crítico del estado del arte en el tema de la Analítica Web, los distintos tipos de herramientas existentes. Se hace un estudio de herramientas similares existentes a nivel nacional e internacional. Se fundamenta la utilización de la metodología y las diferentes herramientas y tecnologías para dar cumplimiento al objetivo general de la investigación.



**Capítulo 2 “Diseño e Implementación del Sistema”:** En este capítulo se delimitan las funcionalidades del sistema a través de requisitos funcionales y no funcionales. Se especifican a través de ejemplos concretos los patrones de diseño, los estilos arquitectónicos utilizados y la arquitectura definida para la implementación de la solución informática. Además, como principal resultado se evidencia la solución práctica propuesta a la investigación realizada.

**Capítulo 3 “Validación y Pruebas del Sistema”:** En este capítulo se presentan algunos de los diseños de casos de prueba a utilizar en la validación del sistema, así como los diferentes tipos de pruebas realizadas al mismo y los resultados asociados a estas.

## Capítulo 1: Marco Teórico Conceptual

### 1.1 Introducción

En el presente capítulo se exponen los principales conceptos asociados al dominio del problema y se realiza un estudio del estado del arte a nivel nacional e internacional, sobre los sistemas de Analítica Web existentes, con el objetivo de conocer si dan respuesta al problema planteado y tener una línea base para el desarrollo de la solución en caso que no satisfagan las condiciones del cliente. Se definen las tecnologías y herramientas además de la metodología a utilizar para el desarrollo de la investigación.

### 1.2 Conceptos asociados al dominio del problema

Para lograr una mejor comprensión de la lectura del capítulo es de vital importancia tener conocimiento de los principales conceptos que se abordan a lo largo del mismo, a continuación se propone una lista con la definición de algunos de ellos:

- **Sitio web:** Un sitio web no es más que un gran espacio documental organizado. Mayormente se encuentra dedicado a algún tema particular o designio definido donde se puede realizar compras, búsquedas, enviar mensajes, y otras actividades interactivas. Un sitio puede contener una combinación de gráficos, texto, audio, vídeo, y otros materiales dinámicos o estáticos (Ganazhapa, 2012).
- **Internet:** Se conoce como internet al conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de Protocolos de Control de Transmisión (TCP, acrónimo en inglés *Transmission Control Protocol*) lo cual garantiza que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial. Sus orígenes se remontan a 1969, cuando se estableció la primera conexión de computadoras, conocida como *Arpanet*, entre tres universidades en California y una en Utah, Estados Unidos. Con la llegada del internet se logró romper las barreras físicas entre regiones remotas (Lima, 2014).
- **Analítica Web:** La Analítica Web puede ser definida como la disciplina profesional encaminada a extraer conclusiones, definir estrategias o establecer reglas de negocio sobre la base de los datos recabados en todos aquellos entornos web sobre los que una empresa ejerce control. La Analítica Web puede ayudar a perfeccionar varios objetivos de negocio de una institución financiera (González, 2014).
- **Registro (log):** Un *log* es un registro oficial de eventos durante un rango de tiempo en particular. Para los profesionales en seguridad informática es usado para registrar datos o información sobre

quién, qué, cuándo, dónde y por qué un evento ocurre para un dispositivo en particular o aplicación. Un *log* es un registro de actividad de un sistema, que generalmente se guarda en un fichero de texto, al que se le van añadiendo líneas a medida que se realizan acciones sobre el sistema. Se utiliza en muchos casos distintos, para guardar información sobre la actividad de sistemas variados (Carrasco, 2014).

- **JavaScript:** *JavaScript* es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente, *JavaScript* es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con *JavaScript* se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. A pesar de su nombre, *JavaScript* no guarda ninguna relación directa con el lenguaje de programación *Java*. Legalmente, *JavaScript* es una marca registrada de la empresa *Sun Microsystems* (LibrosWeb, 2014).
- **Proceso de toma de decisiones:** La toma de decisiones es el proceso mediante el cual se realiza una elección entre las opciones o formas para resolver diferentes situaciones de la vida en diferentes contextos: a nivel laboral, familiar, sentimental, empresarial (utilizando metodologías cuantitativas que brinda la administración). La toma de decisiones consiste, básicamente, en elegir una opción entre las disponibles, a los efectos de resolver un problema actual o potencial (Paéz, 2010).

### 1.3 Estado del arte

En el desarrollo de un proyecto de *software* se define una etapa para el análisis de sistemas similares al que se desea desarrollar, para identificar funcionalidades y métodos que puedan ser incorporados en el producto final. En esta investigación se efectuó el estudio de algunos sistemas existentes en el mundo y en Cuba que se dedican al análisis de registros web para apoyar la toma de decisiones en empresas y organizaciones. Este estudio permitió obtener las principales características, ventajas y desventajas de dichos sistemas, posibilitando la justificación de la solución propuesta como respuesta al problema encontrado y determinando ciertas funcionalidades que deben ser incluidas en la misma. Según los indicadores y la búsqueda realizada se seleccionaron los siguientes sistemas por ser los que contaban con suficiente documentación disponible para su estudio y por ser los más utilizados a nivel mundial. A continuación se detallan algunas de las particularidades fundamentales de las herramientas analizadas.

## 1.3.1 Marco nacional

En Cuba se tiene conocimiento del desarrollo de una herramienta de Analítica Web por parte de la UCI. La implementación de AiresWeb, a petición del Centro de Información para la Prensa (CIPRE) se desarrolló en la antigua facultad 10 (actualmente facultad 1), sistema que en sus inicios realizaba un análisis inteligente de los registros web del servidor en el cual estaba ubicado. El mismo se diseñó para instalarse en un servidor *Debian GNU/Linux*; contaba con una base de datos *PostgreSQL* donde se almacenaba la información generada por el análisis de los registros web además de contar con una interfaz web (IW) para la gestión de la información procesada por el sistema.

Debido al rápido cambio y desarrollo de las tecnologías se identificaron nuevos requisitos que conllevaron a la realización de una nueva versión del sistema AiresWeb. Esta nueva versión del sistema se implementó en *Symfony1*, compone reportes estadísticos a partir de registros generados por un código *script* insertado en las páginas web que se quieren monitorizar; cuenta además con un módulo procesador de datos desarrollado en C/C++ que aumentó la eficiencia y la velocidad del mismo, también se le mejoró la estructura de la base de datos (Chavez, 2012).

## 1.3.2 Marco internacional

En el ámbito internacional existen infinidad de herramientas enmarcadas en esta área de la ciencia, a continuación se exponen algunas de las principales características de las seleccionadas:

- **Google Analytics:** Herramienta privativa que proporciona información muy valiosa sobre el tráfico del sitio web y la eficacia del plan de *marketing*. Posee funciones potentes, flexibles y fáciles de usar. *Google Analytics* ayuda a diseñar anuncios más orientados, a mejorar las iniciativas de *marketing* y a crear sitios web que generen más conversiones. Es una solución de analítica web para empresas que proporciona información muy valiosa sobre el tráfico del sitio web y la eficacia del plan de marketing. Ahora, gracias a unas funciones potentes, flexibles y fáciles de usar, podrá ver y analizar el tráfico desde una perspectiva totalmente distinta. *Google Analytics* le ayudará a diseñar anuncios más orientados, a mejorar sus iniciativas de marketing y a crear sitios web que generen más conversiones (Google).

Esta herramienta permite ver las palabras claves con las que llegan los usuarios a las páginas web, fuentes del tráfico, visitantes, contenidos y comercio electrónico. También se puede obtener informes como el seguimiento de usuarios exclusivos, el rendimiento del segmento de usuarios, los resultados de la campaña de *marketing*, el *marketing* de motores de búsqueda, las pruebas de versión de

anuncios, el rendimiento del contenido, el análisis de navegación, los objetivos y proceso de redireccionamiento o los parámetros de diseño web (2009).

- **Chartbeat:** Esta herramienta privativa es una de las mejores alternativas a *Google Analytics*. Es un servicio para analizar la audiencia de cualquier sitio web, grande o pequeño. El análisis se realiza totalmente en tiempo real. Esto permite obtener una imagen instantánea de qué está sucediendo en la web. El sistema mide lo que los usuarios tardan en cargar las páginas y calcula el promedio para toda la audiencia recibida. Los datos se comparan automáticamente con días anteriores, lo que permite ver si la audiencia va mejorando o empeora, según el registro histórico, y a qué hora sucede todo eso. Se pueden programar alertas, que envíen avisos por correo cuando sucede algo inesperado: la web deja de funcionar, se recibe un tráfico intenso repentinamente o se bate un récord semanal o mensual de audiencia (Gauita, 2009).
- **Reinvigorate:** Esta herramienta privativa posee un arranque de análisis que ofrece el seguimiento en tiempo real y algunas características únicas, como una aplicación de escritorio que le avisa de los usuarios en su sitio. Incluye mapas de calor, además de una interfaz pulida estupenda. El sistema está construido para manejar los sitios de mayor tráfico. Además que brinda actualizaciones en tiempo real en el tablero de instrumentos para mostrar lo que está sucediendo con su sitio en todo momento (Kaushik, 2011).
- **CrazyEgg:** Es una herramienta privativa especializada en analizar de forma visual cómo interactúan los visitantes con sitios web y disponer de datos para intentar mejorar la experiencia de usuario. Cuenta con mapas de calor que permiten identificar zonas calientes donde no esperábamos recibir clics, esta información puede ser de utilidad para potenciar visualmente estos elementos o reubicar contenidos. Permite la visualización de clics aleatorios, por ejemplo clics fuera de un botón que es demasiado pequeño, nos indicará que debemos hacerlo más grande (Kaushik, 2011).
- **Piwik:** Es un *software* de código abierto para el análisis del tráfico web. Este analizador de servidores permite obtener datos y gráficos de los número de visitas de los usuarios en la web, las páginas más vistas, los motores de búsqueda más usados, las palabras clave que utilizan, el lenguaje que hablan incluso hasta sus páginas más populares. *Piwik* reúne toda la información necesaria para sacar conclusiones acerca del tráfico en la web que se desee analizar, proporcionando informes detallados de los usuarios que visitan el sitio. Posee una interfaz personalizable por medio de *widgets*

(componente gráfico, o control, con el cual el usuario interactúa). Esta herramienta pretende ser una alternativa de código abierto a *Google Analytics* (Miller, 2012).

### **1.3.3 Análisis crítico de los sistemas estudiados**

Los sistemas analizados anteriormente no constituyen una respuesta al problema planteado puesto que: elegir una alternativa privativa no es una solución extensible para todas las empresas del país por el tema de costos de licencia, algunas de las cuales almacenan la información en servidores fuera del país, atentando contra la soberanía tecnológica del mismo, por otra parte, puede que los datos guardados sean utilizados con fines ajenos a los intereses de la entidad propietaria; para acceder a muchos de estos sistemas es necesario tener una conexión a internet y existen organizaciones nacionales que no cuentan con la misma; de las opciones distribuidas bajo licencias de *software* libre se tiene como principal inconveniente que no brindan reportes en tiempo real e inclusive algunas están desarrolladas en plataformas prácticamente obsoletas, lo que impide cualquier reproducción parcial o total de su código, también se puede decir que en el caso de *Piwik*, aunque está desarrollado con tecnologías recientes, su código no es totalmente modular, por tanto agregar nuevas funcionalidades o perfeccionar las existentes sería un trabajo muy costoso en tiempo y esfuerzo.

### **1.4 Metodología de desarrollo de software**

Una metodología de desarrollo, en ingeniería de *software*, es un conjunto de herramientas, técnicas, procedimientos y soporte documental encaminados a estructurar, planificar y controlar el proceso de desarrollo de forma organizada y lógica, que tienen como objetivo apoyar a los desarrolladores en la creación de un nuevo *software* (Zambrano, 2013).

Desarrollar un buen *software* depende de un sinnúmero de actividades y etapas, donde el impacto de elegir la mejor metodología para un equipo, en un determinado proyecto es trascendental para el éxito del producto. El papel preponderante de las metodologías es sin duda esencial en un proyecto y en el paso inicial, que debe encajar en el equipo, guiar y organizar actividades. Todo lo anterior conlleva al cumplimiento de las metas trazadas en el grupo.

Las metodologías ágiles se han convertido en herramientas muy utilizadas por las empresas desarrolladoras de software debido al bajo costo en la producción de software y a la reducción en el tiempo de entrega del producto, sin descuidar la calidad y fiabilidad que debe mantener un producto de software que fue desarrollado por un método tradicional.

### 1.4.1 *OpenUP*

Esta metodología es un proceso ágil y unificado, que contiene el conjunto mínimo de prácticas que ayudan a los equipos a ser más fuertes en el desarrollo de software. La selección de la metodología a utilizar en el desarrollo de la solución propuesta está sustentada a que esta es una decisión del centro CIDI, igualmente fue seleccionada debido a que establece una filosofía ágil que se centra en la naturaleza colaborativa del desarrollo de software. También es apropiada para pequeños proyectos donde se disminuyen las probabilidades de que estos fracasen, y se pueden detectar errores en etapas tempranas del desarrollo de un producto de software a través de un ciclo iterativo. También esta metodología genera la documentación necesaria para el desarrollo del proyecto, lo que hace que se entreguen en el tiempo estimado.

#### **Beneficios de esta metodología:**

- Es apropiado para proyectos pequeños y de bajos recursos, permite disminuir las probabilidades de fracaso en los proyectos pequeños e incrementar las probabilidades de éxito.
- Permite detectar errores tempranos a través de un ciclo iterativo.
- Evita la elaboración de documentación, diagramas e iteraciones innecesarios requeridos en la metodología RUP.
- Por ser una metodología ágil tiene un enfoque centrado al cliente y con iteraciones cortas.

Esta metodología está compuesta por cuatro fases:

- **Concepción:** se enfoca en el entendimiento del propósito y objetivos del proyecto que se desarrollará, obteniendo suficiente información para confirmar que se debe hacer. En la misma se debe entender qué construir, identificar los actores y quién está interesado en este sistema.
- **Elaboración:** se tratan los riesgos significativos para la arquitectura. El propósito de esta fase es establecer la base de elaboración de la arquitectura del sistema. Además obtener una comprensión más detallada de los requerimientos que debe cumplir el mismo.
- **Construcción:** se centra en el diseño, implementación y prueba de las funcionalidades para desarrollar un sistema completo. El propósito de la misma es completar el desarrollo del sistema basado en la arquitectura definida.
- **Transición:** su propósito es asegurar que el sistema es entregado a los usuarios, y evalúa la funcionalidad del último entregable de la fase de construcción.

*OpenUP* es un proceso mínimo, completo y extensible. Promueve técnicas ágiles y principios, mientras que tiene un ciclo de vida estructurado que hace hincapié en la continua entrega de *software*, que es valioso para los interesados en el desarrollo rápido de aplicaciones de calidad (Ana P. Rodriguez, 2013).

### **1.5 Análisis del soporte tecnológico para el desarrollo del *software***

La correcta selección de las tecnologías a emplear en el desarrollo de un *software* es uno de los pasos más importantes en el ciclo de vida del mismo debido a que influye directamente en la calidad del producto final y en el esfuerzo del equipo de desarrollo para obtenerlo.

#### **1.5.1 Lenguajes de programación**

##### **PHP5**

Actualmente existe gran diversidad de lenguajes de programación para desarrollar aplicaciones web tales como: *Java*, *PHP* y *Python*, los cuales permiten interactuar con los usuarios y utilizar diversos sistemas de bases de datos. Se decide seleccionar *PHP5*, pues es un lenguaje multiplataforma, es compatible con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, es un lenguaje de programación del lado del servidor y muy utilizado para generar páginas dinámicas. 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. No requiere definición de tipos de variables y cuenta con manejo de excepciones. Su selección también viene dado por el *framework* de desarrollo escogido ya que el mismo fue una especificación del cliente (Garcia, 2010).

##### **JavaScript**

*JavaScript* es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Entre las acciones típicas que se pueden realizar en *JavaScript* se tienen dos vertientes:

1. Por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo.
2. Por el otro, permite ejecutar instrucciones como respuesta a las acciones del usuario, lo cual posibilita crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.

*JavaScript* se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento (Groussard, 2010).



Se decidió utilizar este lenguaje de programación debido a que es bastante sencillo de trabajar con él. Incluso las personas que no tengan una experiencia previa en la programación podrían aprender este lenguaje con facilidad y utilizarlo en toda su potencia, con sólo un poco de práctica, garantizando optimizar el tiempo de desarrollo de las aplicaciones web. Además la selección de este lenguaje viene sustentada por la amplia documentación existente, la experiencia que poseen los desarrolladores con el mismo y porque el *framework* de diseño utiliza este lenguaje.

### 1.5.2 Lenguajes de enmarcado

#### HTML5

El Lenguaje de Marcas de Hipertexto, *HyperText Markup Language* (HTML por sus siglas en inglés), en la versión 5, es la quinta revisión del lenguaje básico de la *World Wide Web* (WWW). Este lenguaje de programación es usado por múltiples desarrolladores web por sus avances, mejoras y ventajas con respecto a las versiones anteriores. El principal criterio de diseño de HTML5 ha sido el de resolver problemas prácticos, y con este objetivo adopta soluciones dirigidas a facilitar el trabajo en situaciones reales.

- **Estructura del cuerpo:** la mayoría de las webs tienen un formato común, formado por elementos como cabecera, pie, navegadores, etc. HTML5 permite agrupar todas estas partes de una web en nuevas etiquetas que representarán cada uno de las partes típicas de una página.
- **Etiquetas para contenido específico:** hasta ahora se utilizaba una única etiqueta para incorporar diversos tipos de contenido enriquecido, como animaciones *Flash* o vídeo. Ahora se utilizarán etiquetas específicas para cada tipo de contenido en particular, como audio y vídeo.
- **Canvas:** es un nuevo componente que permitirá dibujar, por medio de las funciones de una Interfaz de Programación de Aplicaciones, API (del inglés *Application Programming Interface*), en la página todo tipo de formas, que podrán estar animadas y responder a interacción del usuario.
- **Fin de las etiquetas de presentación:** todas las etiquetas que tienen que ver con la presentación del documento, es decir, que modifican estilos de la página, serán eliminadas. La responsabilidad de definir el aspecto de una web correrá a cargo únicamente de CSS (Hipertextual, 2013).

Además de todas las ventajas anteriores, otro criterio por el cual se seleccionó este lenguaje es porque el *framework* de diseño lo utiliza.

#### CSS3

CSS3 es la nueva versión del CSS (*Cascading Style Sheet*, Hojas de Estilo en Cascada). Es simplemente el lenguaje con el cual se puede dar diseño y apariencia a las páginas HTML o XML. Es desarrollado y

distribuido por la W3C (*World Wide Web Consortium*) con el fin especificado anteriormente. A continuación se muestran algunas características que facilitan su uso.

### **Características**

- CSS3 propone una navegación más rápida y menores tiempos de respuesta producidos por una reducción de imágenes, las cuales ya no serán requeridas para diseñar botones o efectos de texto.
- Deja atrás una excesiva dependencia de *JavaScript* para fines de representación visual, como lo son las animaciones, dando como resultado menos código y mejor rendimiento.
- Representa una futura menor dependencia de software para gráficos que resultan bastante caros (Ojeda, 2012).

Otro de los motivos por las cuales se empleó este lenguaje se debe a que el *framework* de diseño lo utiliza.

### **1.5.3 Servidor Web**

El servidor HTTP Apache es un servidor de código abierto. Tiene amplia aceptación en la red siendo el servidor HTTP más usado. La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Está diseñado para ser un servidor web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos que a menudo sean necesarias sus características o funcionalidades. Apache se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular.

Para el desarrollo de la solución se decide utilizar Apache pues es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a nuevos protocolos. Es multiplataforma, con los diferentes módulos de apoyo que proporciona. Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos. Se desarrolla de forma abierta. Es extensible ya que se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor (Bowen, 2007).

Otra de las razones por la cual se utilizó Apache fue porque el cliente especificó su inclusión en la solución

### **1.5.4 Sistema Gestor de Base de Datos**

Un Sistema Gestor de Base de Datos (SGBD, en inglés DBMS: *Database Management System*) es un sistema de *software* que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación *PostgreSQL*.

*PostgreSQL* es un sistema gestor de base de datos relacionales orientada a objetos, derivado de *Postgres*. Es un gestor de bases de datos de código abierto y multiplataforma. Soporta gran parte de la sintaxis SQL y cuenta con un extenso grupo de enlaces con lenguajes de programación. El motor de datos puede incluir las sub-consultas, los valores por defecto, las restricciones a valores en los campos y los disparadores. Permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos. Se escogió *PostgreSQL* ya que es un sistema ampliamente popular e ideal para tecnologías web. Es fácil de administrar. Su sintaxis SQL es estándar y fácil de aprender. Es multiplataforma, cuenta con capacidades de replicación de datos y brinda soporte empresarial disponible (Peter Eisentraut, 2008).

### 1.5.5 *Framework* de Desarrollo

Un *framework*<sup>5</sup> de aplicaciones web permite el desarrollo de sitios web dinámicos, servicios web y aplicaciones web. Su propósito es permitir a los desarrolladores construir aplicaciones web y centrarse en los aspectos interesantes, aliviando la típica tarea repetitiva asociada con patrones comunes de desarrollo web. La mayoría de los *frameworks* de aplicaciones web proporcionan los tipos de funcionalidad básica común, tales como sistemas de plantillas, manejo de sesiones de usuario, interfaces comunes con el disco o el almacenamiento en base de datos de contenido cacheado, y persistencia de datos. Normalmente, los *frameworks* de aplicación web además promueven la reutilización y conectividad de los componentes, así como la reutilización de código, y la implementación de bibliotecas para el acceso a base de datos.

Como rasgos descriptivos de *Symfony2* tenemos:

- **Flexibilidad ilimitada:** Cuales quieran que seas las necesidades, *Symfony2* será adaptable. Su inyector de dependencias y su manejador de eventos, lo hacen enteramente configurable, con cada una de las partes siendo completamente independientes.
- **Estable y sostenible:** La mayor parte de las versiones son soportadas 3 años por la compañía.
- **Facilidad de uso:** Completamente flexible para satisfacer las necesidades de profesionales y usuarios avanzados y es muy accesible. Cuenta con documentación abundante, soporte y permiten a un principiante sentirse a gusto muy rápido (Fabien Potencier, 2014).

---

<sup>5</sup> *Framework* (marco de trabajo) es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de *software* concretos, que puede servir de base para la organización y desarrollo de *software*.

*Symfony2* es un *framework* fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas *Windows* y *Unix* estándares). Propone diferentes y patrones de diseño para la web. Puede integrarse con diferentes gestores de bases de datos. Código fácil de leer que incluye comentarios de *phpDocumentor* y que permite un mantenimiento muy sencillo. Es extensible, flexible como para adaptarse a los casos más complejos. Es orientado a objeto. Se escogió *Symfony2* por todas las ventajas anteriormente señaladas, además, el cliente decidió el uso del mismo en el desarrollo de la solución.

### 1.5.6 Framework de Diseño

*Bootstrap* es un *framework* que simplifica el proceso de creación de diseños web combinando CSS y *JavaScript*. Ha sido desarrollado por *Twitter* que recientemente liberó su versión 3.x La mayor ventaja es que podemos crear interfaces que se adapten a los distintos navegadores (*responsive design*) apoyándonos en un *framework* potente con numerosos componentes webs que nos ahorrarán mucho esfuerzo y tiempo (Genveta:dev, 2012).

Características principales:

- *Bootstrap* ofrece una serie de plantillas CSS y ficheros *JavaScript* que permiten integrar el *framework* de forma sencilla y potente en proyectos webs.
- Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como *tablets* y móviles a distintas escalas y resoluciones.
- Se integra perfectamente con las principales librerías *JavaScript*, por ejemplo *JQuery*.
- Ofrece un diseño sólido usando LESS y estándares como CSS3/HTML5.
- Es un *framework* ligero que se integra de forma limpia en nuestro proyecto actual.
- Funciona con todos los navegadores, incluido *Internet Explorer* usando *HTML Shim* para que reconozca los *tags* HTML5.
- Dispone de distintos *layout* predefinidos con estructuras fijas a 940 píxeles de distintas columnas o diseños fluidos (Genveta:dev, 2012).

Se incluirá *Bootstrap* en la solución propuesta para el desarrollo de las interfaces de usuario, ya que brinda enormes facilidades en este tema, además de ser un requerimiento del cliente.

### 1.5.7 Librería para graficar

*AmCharts* es un programa que te dará la posibilidad de crear infinidad de gráficos ilustrativos, que actuarán de complemento a tu información estadística.

Gracias a esta aplicación, podrás insertar los datos de manera tal que será más simple que editar un documento XML o CSV. Al finalizar puedes exportar los datos a *Excel*.

Los gráficos pueden ser tanto en 2D como es 3D y en forma de barras o columnas. Cada uno de ellos ocupa un espacio estimado de 20Kb.

*AmCharts* te permitirá además modificar la posición de los objetos, los colores de fondo y los elementos que componen el gráfico (Softbull, 2012).

Se utilizará esta librería para generar todas las gráficas de la solución, con el objetivo de hacer llegar al usuario la información de una manera fácil de interpretar y agradable a la vista.

### **1.5.8 Entornos de desarrollos Integrados (IDE por sus siglas en inglés)**

Son programas que reúnen funcionalidades para el trabajo con uno o varios lenguajes de programación. Son por lo general un conjunto de herramientas empaquetadas en una sola aplicación informática, así como un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDE pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

Netbeans 7.4 es un entorno de desarrollo integrado de código abierto para desarrolladores de *software*. Cuenta con todas las herramientas necesarias para crear aplicaciones profesionales de escritorio, empresariales, web y aplicaciones móviles con la plataforma Java, así como con PHP y JavaScript. NetBeans IDE 7.4 amplía el soporte avanzado de desarrollo HTML5, ofrece un nuevo soporte para el desarrollo web móvil. Está disponible en inglés, chino, portugués, japonés y ruso. Se selecciona Netbeans 7.4 como herramienta IDE, por su soporte a tecnologías, su flexibilidad, y su potencia para el desarrollo de aplicaciones web (Oracle, 2013).

### **1.5.9 Lenguaje de Modelado**

Los lenguajes de modelado, tanto los gráficos como los textuales, permiten la representación del comportamiento y la estructura de los sistemas del mundo real. Los lenguajes de modelado gráfico utilizan modelos visuales y diagramas que realizan esa representación de manera precisa, entendible y económica.

#### **Lenguaje Unificado de Modelado:**

El Lenguaje Unificado de Modelado (UML por sus siglas en inglés) es un lenguaje de modelado estandarizado de propósito general en el campo de la ingeniería de *software* orientada a objetos. UML incluye un conjunto de técnicas de notación gráfica para crear modelos visuales de programación orientada a objetos.

Los estándares propuestos por el lenguaje UML ofrecen amplias ventajas para el desarrollo de la solución, por lo que se escoge como lenguaje de modelado a utilizar en la realización del presente trabajo (Groussard, 2010).

### **1.5.10 Herramienta de ingeniería de software asistida por computación**

Las herramientas de ingeniería de software asistida por computación, también conocidas por sus siglas en inglés *CASE* (*Computer Aided Software Engineering*) facilitan el trabajo de ingeniería de un proyecto, aportando funcionalidades para la organización, el modelado y el seguimiento de un *software*.

*Visual Paradigm* es una herramienta CASE. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.

*Visual Paradigm* ha sido concebido para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta de *software* libre de probada utilidad para el analista que fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de *software* de forma fiable a través de la utilización de un enfoque orientado a objetos (Sommerville, 2005).

#### **Entre sus principales características se encuentran:**

- Entorno de creación de diagramas para UML.
- Diseño centrado en casos de uso y enfocado al negocio que genera un *software* de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo. Disponibilidad de múltiples versiones, para cada necesidad.
- Generación de bases de datos, transformación de diagramas de Entidad-Relación en tablas de base de datos.

Se selecciona *Visual Paradigm* como herramienta CASE. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del *software* a través de la representación de todo tipo de diagramas. Constituye una herramienta privada disponible en varias ediciones.

### **1.5.11 Herramienta adicional**

*Elasticsearch* es un motor de búsqueda que puede indexar nuevos documentos en tiempo casi real y hacerlo inmediatamente disponible para realizar consultas. Esta herramienta se basa en Apache y permitiendo la creación de grupos de nodos que almacenan un número de índices de forma distribuida y tolerante a fallos.

Si un nodo desaparece, el clúster reequilibra los índices en los nodos restantes. Puede configurar el número de fragmentos que conforman cada índice y el número de réplicas de estos fragmentos que debería haber. Está orientada a documentos y, opcionalmente, sin esquema. Cuenta con una potente interfaz de consulta y filtro para la búsqueda en campos particulares, o hacer búsquedas de texto completo. Entre otras de sus características encontramos que *Elasticsearch* se distribuye, las réplicas son casi en tiempo real además apoya plenamente la cerca de búsqueda en tiempo real de *Apache* y posee un mecanismo de copias de seguridad muy fácil de utilizar (Byteflair, 2013).

Se escoge esta herramienta con el objetivo de gestionar los datos estadísticos obtenidos de las interacciones por los usuarios con las páginas web, tuvo gran peso en la selección de la misma las características de poseer una gran velocidad en operaciones de almacenamiento y búsqueda de datos, siendo una alternativa de solución al problema de la entrega de reportes en tiempo real, y la gran escalabilidad que brinda dicha herramienta.

## **1.6 Conclusiones**

En el presente capítulo se trataron conceptos generales acerca de los analizadores de registros web con el objetivo de ubicar al lector en el dominio del problema, se caracterizaron los mismos como una poderosa herramienta para el proceso de toma de decisiones en las organizaciones que han expandido su modelo de negocio hacia la red de redes. Además la investigación exhaustiva de sistemas homólogos demostró que los mismos no cumplen con los requisitos necesarios para darle solución al problema planteado, sin embargo, poseen una serie de funcionalidades que permiten tener una línea base de desarrollo para la realización de la propuesta de solución.

Como resultado del estudio realizado se seleccionó la metodología de desarrollo de software *OpenUP* para la planeación y elaboración del sistema. Se eligió *Symfony2* como *framework* para el desarrollo del sistema y *Bootstrap* para el diseño, combinación que permitirá la realización de un sistema con calidad y agilizar el proceso de implementación por las enormes facilidades que brindan. Se escogió Apache como servidor web, se optó por *PostgreSQL* como sistema gestor de base de datos para los datos administrativos y *Elasticsearch* para los datos estadísticos. Para el entorno de desarrollo integrado se decidió utilizar *NetBeans*, para el modelado de los diagramas propuestos se definió el *Visual Paradigm* y como librería para graficar Amcharts.

## Capítulo 2: Diseño e Implementación del Sistema

### 2.1 Introducción

En el presente capítulo se detallará el funcionamiento del sistema propuesto para dar solución a la problemática planteada. También se especificarán y describirán los requerimientos que dicho sistema debe cumplir para satisfacer al cliente final.

Durante la etapa de diseño se modelará el sistema para que dé respuesta a todos los requisitos, incluyendo los no funcionales y se mostrarán algunos de los artefactos generados por la metodología seleccionada como evidencia del trabajo realizado.

Durante la implementación se lograrán las tareas de programación definidas para darle cumplimiento al desarrollo de la solución propuesta, además se presentarán aspectos como estándares de codificación utilizados durante la misma.

### 2.2 Propuesta del sistema

Se tiene como propósito desarrollar un sistema analizador de registros web (ARWeb). Dicha solución podrá monitorizar las actividades realizadas por los usuarios en los sitios web cubanos, y brindar reportes estadísticos sobre las mismas en tiempo real. La misma contará con una interfaz web para la presentación de dichos reportes en forma de tablas y gráficas, permitiendo esto que sea más entendible y asimilable para el usuario del sistema. El análisis de estas estadísticas ofrecerá a directivos o personal administrativo de la entidad en general, nuevos elementos para apoyar la toma de decisiones dentro de la misma.

De manera general su funcionamiento se basará en un código *JavaScript* el cual se insertará en las páginas que se deseen monitorizar. Mediante este código se recogerán disímiles datos sobre la entrada de usuarios a la página en cuestión, como son: página de origen, sistema operativo y navegador que utiliza así como sus respectivas versiones, fecha y hora de la visita, duración de la visita, entre otros. Con esta información ARWeb mediante diferentes reglas de negocio analizará, almacenará y graficará los datos recibidos.

#### 2.2.1 Descripción de actores

Actores	Descripción
Usuario	Persona que tendrá acceso al sistema como un cliente del mismo.
Administrador	Persona encargada de realizar cualquier modificación en el sistema teniendo acceso total al mismo.



Anónimo	Persona que solo podrá solicitar su cuenta al sistema y posteriormente activarla.
---------	---

Tabla 1: Definición de los actores.

### 2.3 Modelo de dominio

Según Craig Larman en la 2da edición del libro UML y Patrones “Un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes software. No se trata de un conjunto de diagramas que describen clases software, u objetos software con responsabilidades” (Larman, 2003).

En la Figura 1 se muestra el diagrama de modelo de dominio que representa las relaciones que existen entre los principales conceptos asociados al tema de investigación.

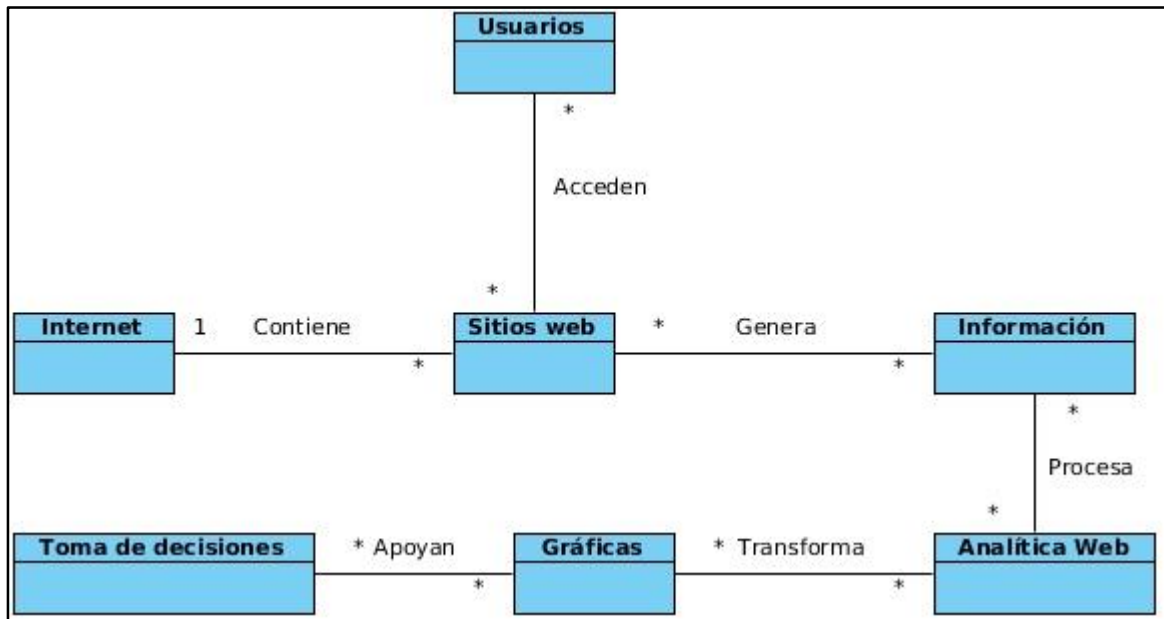


Figura 1: Modelo de dominio.

#### 2.3.1 Descripción de clases del modelo del dominio

- **Internet:** Red informática que conecta computadoras distribuidas por todo el mundo, permitiendo comunicación, búsqueda y transferencia de información.
- **Sitios web:** Espacios virtuales ubicados en internet que se encuentran dedicados a algún tema particular o donde se pueden realizar compras, búsquedas, enviar mensajes, y otras actividades interactivas.

- **Usuarios:** Personas naturales las cuales utilizan los sitios web en su actuar cotidiano con un fin específico.
- **Información:** Resultado de la interacción de los usuarios con los sitios web.
- **Analítica Web:** Área de la ciencia que se encarga de analizar la información generada por los usuarios en la web.
- **Gráficas:** Modo en que las herramientas de Analítica Web presentan los resultados del análisis realizado.
- **Toma de decisiones:** Proceso de evaluación de múltiples alternativas con relación a un resultado esperado por parte de entidades y organizaciones, el cual se nutre con la información gráfica brindada por de las herramientas de Analítica Web.

## 2.4 Modelado del sistema

### 2.4.1 Técnicas de obtención de requisitos

Para la realización de este trabajo se emplearon diferentes técnicas en la obtención de los requisitos tanto funcionales como no funcionales, los cuales permitieron comprender el dominio del sistema, buscar y recolectar información para definir sus límites y restricciones, así como identificar a las personas involucradas en el mismo.

Las técnicas antes mencionadas fueron: la entrevista y los cuestionarios; a continuación se describirá brevemente como las mismas fueron utilizadas:

- **Entrevista:** se realizaron reuniones entre el equipo de desarrollo y el cliente que permitieron entender el dominio del problema y sus necesidades. Éstas se basaron en un conjunto de preguntas y respuestas, buscando obtener una comprensión general de lo qué y cómo el cliente deseaba.
- **Cuestionarios:** se realizaron preguntas previamente elaboradas al cliente para recopilar opiniones y criterios relevantes para la fase de implementación (ver Anexo A). Combinando esta técnica con las entrevistas, se pudo obtener información vital para complementar una propuesta que cumpliera con las expectativas del cliente.

### 2.4.2 Requisitos funcionales (RF)

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. Se mantienen invariables sin importar con qué propiedades o cualidades se relacionen y muestran cómo los casos de uso serán llevados a la práctica (Pressman, 2005. ). Las entrevistas realizadas arrojaron un total de 34 requisitos funcionales, que a continuación se presentan:

- RF1. El sistema debe permitir iniciar sesión de usuario.
- RF2. El sistema debe permitir finalizar la sesión de usuario.
- RF3. El sistema debe permitir que el usuario edite su información personal.
- RF4. El sistema debe permitir solicitar cuenta.
- RF5. El sistema debe permitir activar cuenta.
- RF6. El sistema debe permitir insertar usuario por parte del administrador.
- RF7. El sistema debe permitir modificar usuario por parte del administrador.
- RF8. El sistema debe permitir eliminar usuario por parte del administrador.
- RF9. El sistema debe permitir mostrar usuario por parte del administrador.
- RF10. El sistema debe permitir insertar sitio.
- RF11. El sistema debe permitir modificar sitio.
- RF12. El sistema debe permitir eliminar sitio.
- RF13. El sistema debe permitir mostrar sitio.
- RF14. El sistema debe permitir insertar rol por parte del administrador.
- RF15. El sistema debe permitir modificar rol por parte del administrador.
- RF16. El sistema debe permitir eliminar rol por parte del administrador.
- RF17. El sistema debe permitir mostrar rol por parte del administrador.
- RF18. El sistema debe permitir seleccionar sitio a monitorizar por la aplicación.
- RF19. El sistema debe permitir seleccionar un rango de fechas específicas a monitorizar.
- RF20. El sistema debe brindar reportes sobre las páginas visitadas.
- RF21. El sistema debe brindar reportes sobre las visitas únicas.
- RF22. El sistema debe brindar reportes sobre todas las visitas realizadas a un sitio determinado.
- RF23. El sistema debe brindar reportes sobre las visitas distribuidas anualmente.
- RF24. El sistema debe brindar reportes sobre las visitas distribuidas mensualmente.
- RF25. El sistema debe brindar reportes sobre las visitas distribuidas semanalmente.

RF26. El sistema debe brindar reportes sobre las visitas distribuidas diariamente.

RF27. El sistema debe brindar reportes sobre las visitas distribuidas por horas.

RF28. El sistema debe brindar reportes sobre los principales navegadores.

RF29. El sistema debe brindar reportes sobre los principales sistemas operativos utilizados.

RF30. El sistema debe brindar reportes sobre las direcciones IP<sup>6</sup> y las páginas desde las que accedieron al sitio.

RF31. El sistema debe brindar reportes en forma de gráfico de pastel.

RF32. El sistema debe brindar reportes en forma de gráfico de barras.

RF33. El sistema debe generar el código de seguimiento.

RF34. El sistema debe mostrar el código de seguimiento.

### **2.4.3 Requisitos no funcionales (RNF)**

Los requerimientos no funcionales son cualidades o propiedades que el producto debe poseer, o restricciones en el producto que está siendo desarrollado. Estos no describen lo que el *software* hará, sino cómo lo hará. Estos requisitos son de gran significación en la aceptación del *software*, debido a que representan las ventajas más visibles al usuario y repercuten en el óptimo funcionamiento y mantenimiento del sistema. Son aquellos requisitos que especifican criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que éstos corresponden a los requisitos funcionales. Por tanto, se refieren a todos los requisitos que ni describen información a guardar, ni funciones a realizar (Pressman, 2005. ). Como parte de la captura de requisitos no funcionales se identificaron un total de 16, agrupados en 10 categorías:

#### **Seguridad:**

RNF1. La aplicación podrá ser utilizada sólo por los usuarios que estén registrados en el sistema.

#### **Navegabilidad:**

RNF2. La aplicación web tendrá una interfaz intuitiva.

#### **Confiabilidad:**

---

<sup>6</sup> Una dirección IP es una etiqueta numérica que identifica, de manera lógica y jerárquica, a una interfaz (elemento de comunicación/conexión) de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP (*Internet Protocol*).

RNF3. La información no podrá ser modificada por usuarios no autorizados, protegiendo así la integridad de los datos.

## **Licencia:**

RNF4. Se requiere el uso de herramientas y tecnologías de software libre, las cuales se podrán usar, modificar y distribuir libremente.

## **Diseño e Implementación:**

RNF5. Se requiere el uso de PHP 5.4 en adelante como lenguaje de programación del lado del servidor.

RNF6. Se requiere el uso de *Symfony 2.3.7* como *framework* para el desarrollo y *Bootstrap* para el diseño de la interfaz de usuario.

RNF7. Se requiere *PosgreSQL 9.1* como gestor de bases de datos para los datos administrativos y *Elasticsearch* como base de datos para la información estadística.

RNF8. Se requiere el uso de *Apache2* como servidor web.

## **Soporte:**

RNF9. El sistema debe dar la posibilidad de ser actualizado, así como de incorporarle nuevas funcionalidades en caso de ser necesarias.

## **Hardware:**

RNF10. Para garantizar un buen funcionamiento del sistema, el servidor donde estará desplegado deberá cumplir como mínimo las siguientes condiciones: ser Pentium IV o sus equivalentes de otras compañías, con una velocidad de 3.00 GHz, 3 GB de RAM y 160 GB de disco duro, esta última restricción está condicionada a la cantidad de tráfico generada por los sitios monitorizados.

## **Software:**

RNF11. Para el cliente como versión más antigua del navegador: *Mozilla Firefox 25*, *Google Chrome 15*, *Opera 9* e *Internet Explorer 11*.

RNF12. Para el servidor web *Apache 2.2* o superior con módulo *PHP 5.x* disponible.

## **Eficiencia:**

RNF13. El sistema debe ser capaz de responder con rapidez a las peticiones de los usuarios, demorando como promedio en una transición de 1 a 5 segundos.

RNF14. El sistema debe permitir trabajar de manera concurrente a 150 usuarios.

## 2.5 Diagrama de casos de uso (CU) del sistema

Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. Estos muestran la relación entre los actores y los casos de uso en un sistema. Se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo (24). A continuación se presenta el diagrama de casos de uso de la aplicación ARWeb. (Ver figura 2).

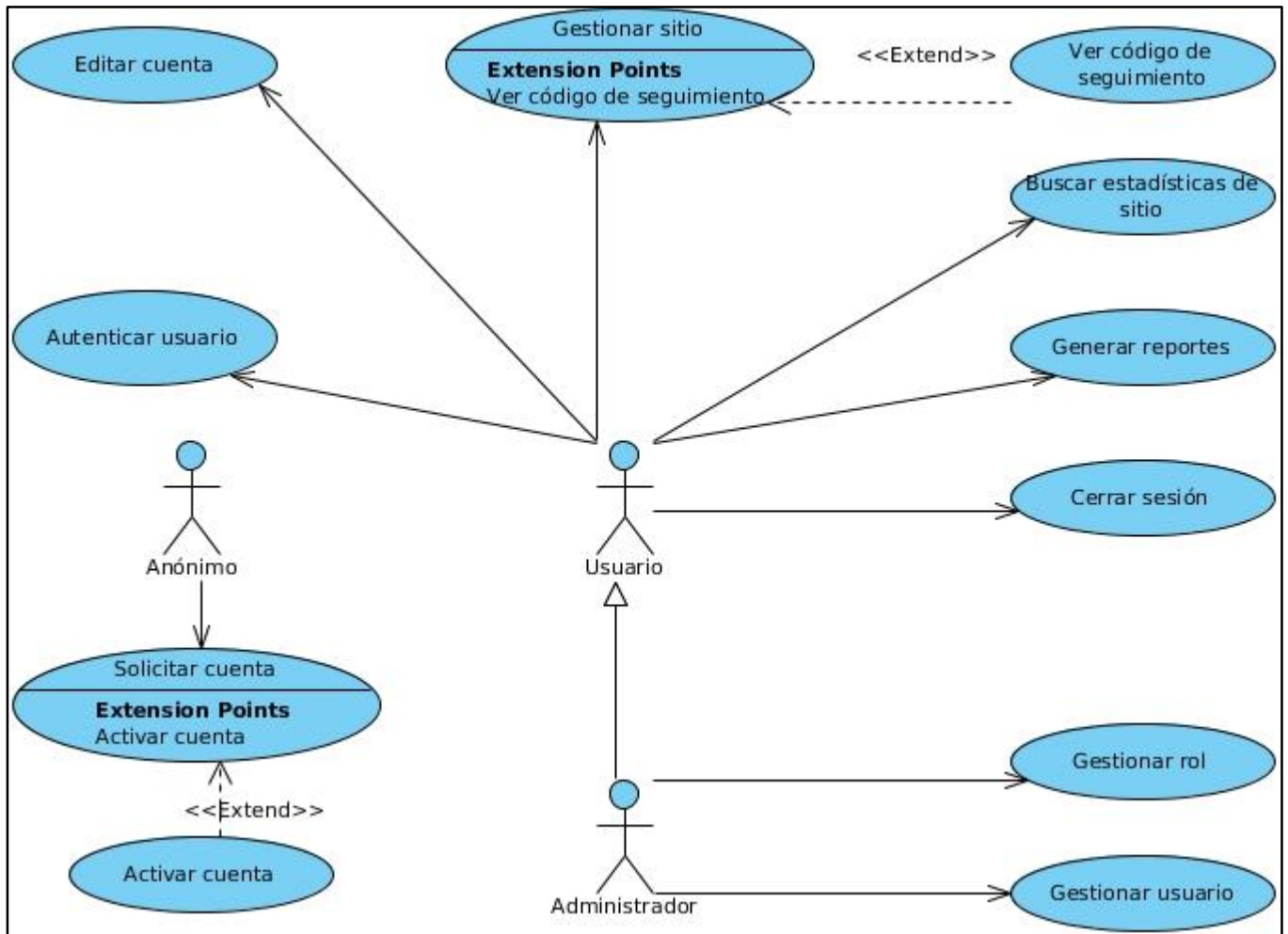


Figura 2: Diagrama de casos de uso del sistema.

### 2.5.1 Especificación de CU

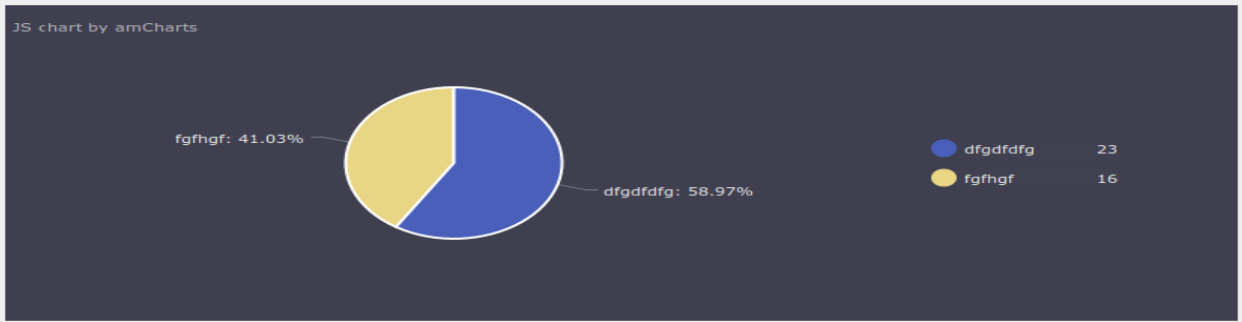
#### CU Generar Reportes

Objetivo	Generar los distintos tipos de reportes.
Actores	Administrador, Usuario

Resumen	El CU se inicializa cuando el actor (en lo adelante usuario), desea visualizar la información estadística generada por sus sitios.
Complejidad	Alta.
Prioridad	Crítico.
Precondiciones	El usuario ha sido autenticado en la aplicación. El usuario ha registrado al menos un sitio. El o los sitios registrados han recibido al menos una visita.
Postcondiciones	Se mostraron los tipos de reportes deseados.

**Prototipo**

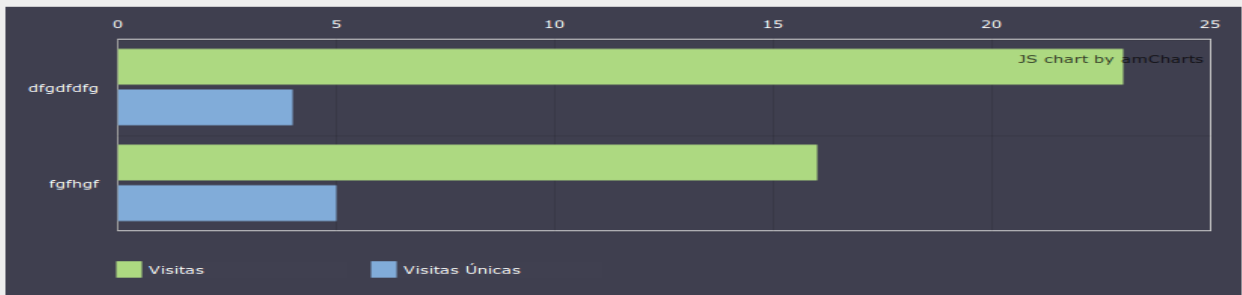
**Visitas Generales**

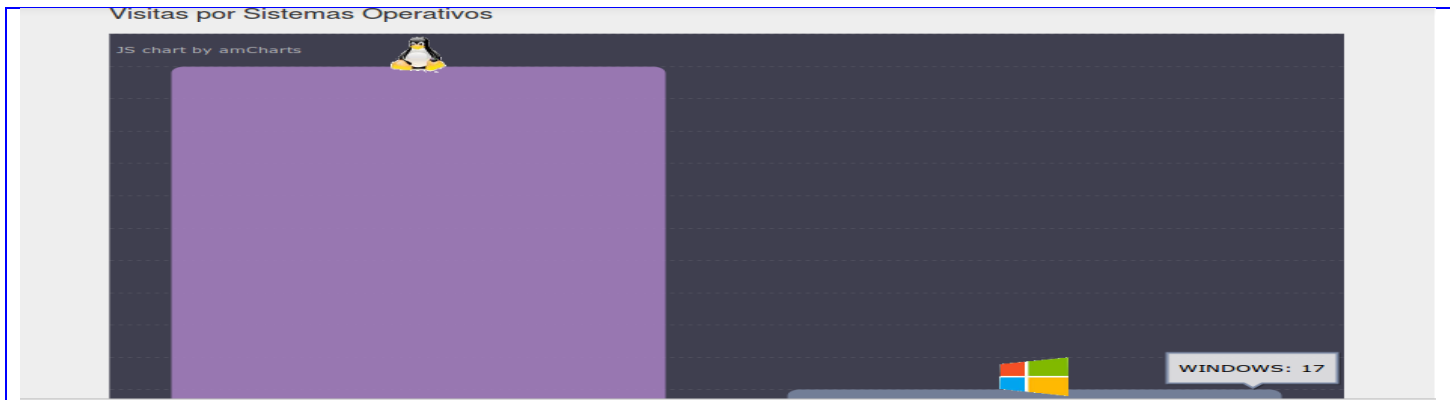


**Visitas por Navegadores**



**Visitas\Visitas Únicas Generales**





Flujo de eventos		
Flujo básico Generar reportes		
	Actor	Sistema
1	Accede a la página principal de la aplicación.	
2		<p>Muestra los siguientes reportes:</p> <ul style="list-style-type: none"> <li>- Visitas generales</li> <li>- Visitas\Visitas Únicas Generales</li> <li>- Visitas por Navegadores</li> <li>- Visitas por Sistemas Operativos</li> </ul> <p>También da la opción de elaborar una búsqueda más detallada. (ver CU Buscar estadística de sitio)</p>
3		Termina CU
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Tabla 2: CU Generar reportes.

CU Gestionar sitio

<b>Objetivo</b>	Gestionar sitio
<b>Actores</b>	Administrador, Usuario
<b>Resumen</b>	El CU se inicializa cuando el actor (en lo adelante usuario) desea gestionar algún sitio.



<b>Complejidad</b>	Alta
<b>Prioridad</b>	Crítico
<b>Precondiciones</b>	El usuario ha sido autenticado en el sistema.
<b>Postcondiciones</b>	Se realizaron satisfactoriamente las acciones sobre cualquier sitio.

**Prototipo**

Id	Nombre del sitio	Url	Sitio activo	Acciones	Código
20	gffg	http://sddf.d.com	no	2 3 • <a href="#">Mostrar</a> • <a href="#">Editar</a>	• <a href="#">Ver código</a>
17	sdfdfg	http://swds.sd.com	no	• <a href="#">Mostrar</a> • <a href="#">Editar</a>	• <a href="#">Ver código</a>
12	dfgdfdfg	kchareo/otro.html	si	• <a href="#">Mostrar</a> • <a href="#">Editar</a>	• <a href="#">Ver código</a>
11	fgfhgf	kchareo/o.html	si	• <a href="#">Mostrar</a> • <a href="#">Editar</a>	• <a href="#">Ver código</a>

**Flujo de eventos**

**Flujo básico Gestionar Nomencladores**

	Actor	Sistema
1.	Pulsa la opción "Sitios" del menú principal	
2.		Muestra una lista con los sitios registrados por el usuario y brinda la opción de: - Nuevo ( <b>Ver Sección 1</b> ). - Mostrar ( <b>Ver Sección 2</b> ). - Editar ( <b>Ver Sección 3</b> ). - Ver código - Atrás
3.	Selecciona una opción.	
4.		Muestra la vista referente a la opción que seleccionó.

5.		Termina CU.
<b>Flujos alternos</b>		
<b>Atrás</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa la opción "Atrás".	
2.		<b>Redirecciona a la portada de la aplicación.</b>
<b>No tiene sitios registrados</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Muestra el mensaje "No tiene sitios registrados, para agregar uno, pulsa la opción Nuevo".
2.	Pulsa la opción "Nuevo"	
3.		<b>Ver sección 1.</b>
<b>Sección 1: "Nuevo" (Prototipo - 1)</b>		
<b>Flujo básico Nuevo</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa la opción "Nuevo"	
2.		Muestra una vista con el formulario correspondiente.
3.	Rellena el formulario.	
4.	Pulsa la acción "Crear".	
5.		Valida que los datos sean correctos.
6.		Almacena la información.
7.		<b>Paso 2 del flujo básico</b> y muestra el mensaje: "!!Aviso!! Sitio añadido correctamente"
8.		Termina Sección 1
<b>Flujos alternos</b>		
<b>Datos Incorrectos</b>		
	<b>Actor</b>	<b>Sistema</b>

1.		Verifica que los datos son incorrectos.
2.		Selecciona los campos incorrectos y muestra un mensaje.
<b>Cancelar</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa opción "Cancelar"	
2.		<b>Paso 2 del flujo básico</b>
<b>Sección 2: "Mostrar" (Prototipo - 2)</b>		
<b>Flujo básico Listado de Roles</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa la opción "Mostrar".	
2.		Muestra una vista con los datos correspondientes al sitio seleccionado. Además permite hacer las acciones:  - Eliminar - Editar
3.		Termina Sección 2
<b>Flujos alternos</b>		
<b>Eliminar</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa la opción "Eliminar"	
2.		Elimina el sitio.
3.		<b>Paso 2 del flujo básico</b> y muestra el mensaje: "!!Aviso!! Sitio eliminado con éxito".
<b>Editar</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa la opción "Editar"	
2.		<b>Ver Sección 3.</b>
<b>Sección 3: "Editar" (Prototipo - 3)</b>		

<b>Flujo básico Editar</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa la opción "Editar".	
2.		Muestra una vista con el formulario y los datos del rol seleccionado. Además permite:  - Eliminar
3.	Modifica los datos.	
4.	Pulsa la opción "Actualizar".	
5.		Valida los datos entrados.
6.		Almacena la información.
7.		<b>Paso 2 del flujo básico</b> y muestra el mensaje: "!!Aviso!! Sitio actualizado con éxito".
8.		Termina Sección 3
<b>Flujos alternos</b>		
<b>4 Datos Incorrectos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Verifica que los datos son incorrectos.
2.		Selecciona los campos incorrectos y muestra un mensaje.
3.		<b>Ir al paso 2 de la Sección 3</b>
<b>Pulsa opción "Eliminar"</b>		
	<b>Actor</b>	<b>Sistema</b>
	Pulsa la opción "Eliminar".	
1.		Elimina el rol.
2.		<b>Paso 2 del flujo básico</b> y muestra el mensaje: "!!Aviso!! Rol eliminado con éxito".
<b>Atrás</b>		
	<b>Actor</b>	<b>Sistema</b>

1.	Pulsa la opción "Atrás"	
2.		<b>Paso 2 del flujo básico.</b>
<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	<b>Ver código de seguimiento Paso 2 del Flujo Básico.</b>
<b>Requisitos no funcionales</b>		
<b>Asuntos pendientes</b>		

Tabla 3: CU Gestionar sitio.

Si es necesario consultar las descripciones de los demás casos de uso, puede examinar el Anexo B.

## 2.6. Descripción del sistema propuesto

El sistema propuesto tiene como objetivo fundamental proporcionar una herramienta de Analítica Web a organizaciones nacionales, para apoyar el proceso de tomas de decisiones de las mismas. Debe ser capaz de brindar en tiempo real diferentes tipos de reportes en diferentes períodos de tiempo, los cuales se presentarán en forma de tablas y gráficos para su mejor comprensión. El mismo debe permitir el acceso a las personas autorizadas y garantizar la seguridad de la información que maneja.

### 2.6.1 Patrones de diseño implementados

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de *software*. En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de *software* que están sujetos a contextos similares. Se deben tener presente los siguientes elementos de un patrón: su nombre, el problema (cuando aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios) (Suárez, 2010).

Los Patrones Generales de *Software* para Asignar Responsabilidades (GRASP del inglés *General Responsibility Assignment Software Patterns*) tienen una importante utilidad en el diseño de una aplicación al igual que los *Gang-of-Four* o Pandilla de los Cuatro (GoF por sus siglas en inglés). Los empleados en la solución de la problemática planteada, se exponen a continuación:

#### Patrones GRASP:

**Experto** es un patrón para asignar una responsabilidad al experto en información: la clase que posee la información necesaria para cumplir con la responsabilidad. Con él no se pretende designar una idea oscura ni extraña; expresa simplemente la "intuición" de que los objetos hacen cosas relacionadas con la información que poseen (Larman, 2003).

*Symfony2* utiliza este patrón con la inclusión de *Doctrine* para el mapeo de base de datos. Se utiliza específicamente para crear una capa de abstracción en el modelo, encapsular toda la lógica de los datos y generar las clases entidades con todas las funcionalidades comunes (*GET*, *SET* y el constructor de la entidad); las clases de abstracción de datos poseen un grupo de funcionalidades que están relacionadas directamente con la entidad que representan y contienen la información necesaria de la tabla asociada.

En la solución implementada se pone de manifiesto este patrón en la clase *SitioController*, cuando se añade un nuevo sitio, pues el sistema debe generar el “código de seguimiento” para dicho sitio y lo hace mediante esta clase la cual posee la información del mismo.

**Creador** es el patrón para asignar a la clase B la responsabilidad de crear una instancia de la clase A, si se cumple una de las siguientes condiciones: B contiene A, B agrega A, B tiene los datos de inicialización de A, B registra A, B utiliza A muy de cerca (Larman, 2003).

Este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento.

Este patrón es utilizado principalmente para el trabajo con las entidades que se almacenan en la base de datos, un ejemplo de su utilización es cuando el usuario anónimo solicita la creación de una cuenta, la clase *UsuarioController* es la encargada de instanciar los métodos de la entidad Usuario, que luego se almacenará en la base de datos, debido a que obtiene todos los datos necesarios para la inicialización de los mismos.

**Bajo Acoplamiento** impulsa la asignación de responsabilidades de manera que su localización no incremente el acoplamiento hasta un nivel que lleve a los resultados negativos que puede producir un acoplamiento alto. No soporta el diseño de clases que son más independientes, lo que reduce el impacto del cambio. El mismo no se puede considerar de manera aislada a otros patrones como el **Experto** o el de **Alta Cohesión**, sino que necesita incluirse como uno de los diferentes principios de diseño que influyen en una elección, al asignar una responsabilidad (Larman, 2003).

El bajo acoplamiento se evidencia en el proyecto completo, ya que todas las clases controladoras heredan únicamente de la clase *Controller* para alcanzar un bajo acoplamiento de clases y no existe otra dependencia o herencia entre clases en la implementación.

**Alta Cohesión** este patrón es un principio a tener en mente durante todas las decisiones de diseño. Incrementa la claridad y facilita la comprensión del diseño. Se simplifican el mantenimiento, las mejoras y soporta a menudo bajo acoplamiento (Larman, 2003).

Se puede observar claramente en el sistema ya que cada clase controladora se ajusta a manejar solo las responsabilidades correspondientes a las entidades con las que se relaciona, además para cada página cliente existe una página servidora encargada de manejar sus solicitudes poniéndose de manifiesto el patrón de diseño alta cohesión.

**Controlador** es el encargado de asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. La mayor parte de los sistemas reciben eventos de entrada externa, en cualquiera de los casos que puedan existir, si se recurre a un diseño orientado a objetos, hay que elegir los controladores que manejen esos eventos de entrada (Larman, 2003).

Se pone de manifiesto en todo el sistema ya que cada uno de los eventos generados por el usuario es redirigido a una clase controladora para que realice las operaciones solicitadas.

## **Patrones GoF:**

**Observador** este patrón define, en sentido general, una dependencia de tipo uno-a-muchos entre objetos, de forma tal que cuando uno de los objetos cambia su estado, notifica su cambio a todos sus dependientes.

La solución muestra la utilización de este patrón en los objetos de tipo Usuario, ya que la clase entidad Usuario, presenta una relación uno-a-mucho con la entidad Sitio y de esta forma si se produce algún cambio en un objeto de esta última, se notificará a los objetos de tipo Sitio con los cuales tiene relación.

**Decorador** *Symfony2* utiliza este patrón en la creación de plantillas generales para que otras las utilicen. Este patrón se utiliza en la solución con la creación de la plantilla base.html.twig, la cual posee en su contenido el código que será general para todas las interfaces de la solución, evitando de esta forma, tener que copiar este código en cada plantilla de la misma (Larman, 2003).

## **2.6.2 Patrón arquitectónico implementado por *Symfony2***

*Symfony2* es un *framework* diseñado para optimizar, proporcionando varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web. *Symfony2* está fundamentado en un patrón clásico del diseño web conocido como arquitectura modelo vista controlador (MVC). El modelo representa la lógica de negocio, la vista transforma el modelo en una página web que permite al usuario interactuar con ella y el controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. En el sistema, el modelo es la representación específica de la información y gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, el controlador responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información.

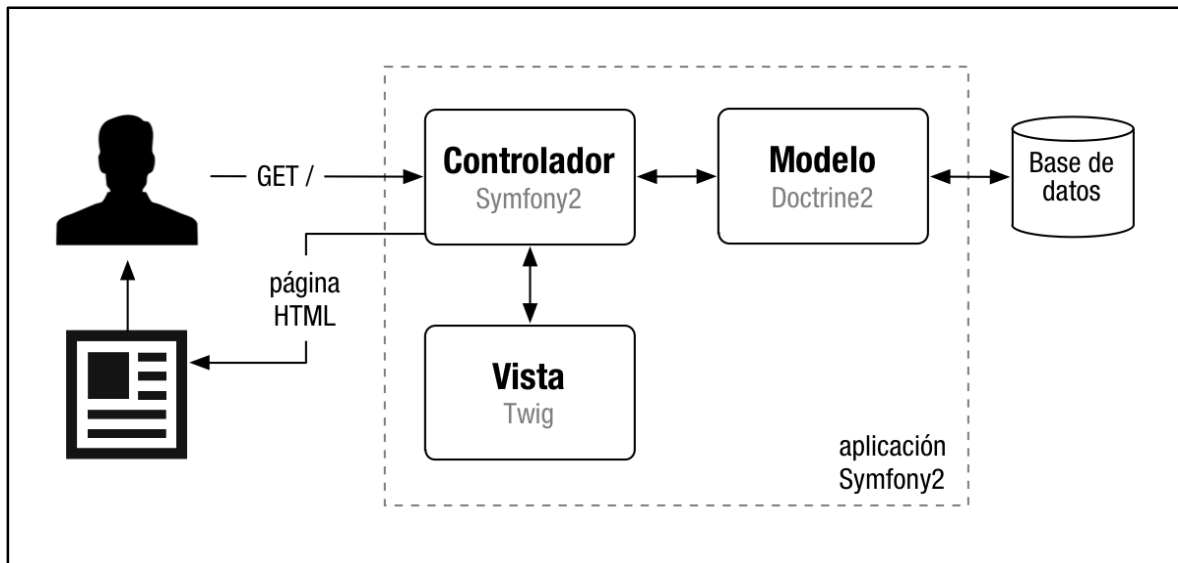


Figura 3<sup>7</sup>: Diagrama del patrón arquitectónico MVC.

"*Symfony2* no es un *framework* MVC. *Symfony2* sólo proporciona herramientas para la parte del Controlador y de la Vista. La parte del Modelo es responsabilidad del desarrollador". La separación de la vista y el modelo trae ventajas; es posible tener diferentes representaciones de la misma información, haciendo uso del mismo código dentro del modelo. Es posible además programar el código del modelo, abstrayéndose de la representación visual que se le dará a la información (Sebastián, 2010).

### 2.6.3 Estructura en *bundles* de la aplicación

Los *bundles* son la base de la nueva filosofía de trabajo de *Symfony2*. El código de tus aplicaciones y el propio código fuente de *Symfony2* se estructura mediante *bundles*. Técnicamente, un *bundle* es un directorio que contiene todo tipo de archivos dentro una estructura jerarquizada de directorios. Los *bundles* de las aplicaciones *Symfony2* suelen contener clases PHP y archivos web (*JavaScript*, CSS e imágenes). No obstante, no existe ninguna restricción sobre lo que puedes incluir dentro de un *bundle*. Tampoco existen límites técnicos sobre el tamaño que puede llegar a tener un *bundle* (Eguiluz, 2013).

Para el desarrollo de la solución al problema planteado se decide crear un proyecto de *Symfony2* que consta de tres *bundles*:

**BackEndBundle:** es el encargado de la gestión de los roles y usuarios de la aplicación.

<sup>7</sup> Tomado de: "Desarrollo ágil con *Symfony2*" por Javier Eguiluz publicado el 7-11-2013.



**FrontEndBundle:** es el encargado de las páginas estáticas del sitio, además de que en él se define la plantilla base la cual utilizarán todas las plantillas restantes de la aplicación.

**SitioBundle:** es el encargado de la gestión de los sitios registrados en la aplicación, así como toda la lógica asociada a los datos estadísticos relacionados a éstos.

## 2.6.4 Seguridad en el sistema

En el desarrollo de una aplicación web, la seguridad del sistema es uno de los aspectos más importantes, ya que es la garantía de la confidencialidad, integridad y disponibilidad de la información. Muchas aplicaciones web se ven afectadas en numerosas ocasiones por diferentes vulnerabilidades del propio código fuente o por descuido de los desarrolladores. En este sentido *Symfony2* implementa una serie de mecanismos para garantizar una aplicación segura, definiendo una lista de control de acceso, en el fichero *app/config/security.yml*.

Aparte de la seguridad que proporciona el *framework* utilizado, se estableció el mecanismo de autenticación al sistema basado en usuario y contraseña, además, para esta última se empleó un algoritmo de cifrado para almacenarla en la base de datos y se utilizó el modelo de control de acceso basado en roles, con estos procedimientos se asegura que sólo un usuario correctamente identificado pueda acceder al sistema, garantizando la confidencialidad e integridad de los datos, y estos usuarios solo podrán accionar sobre la información precisa que por el rol establecido en el sistema pueden controlar, también se estableció el protocolo *HTTPS* para asegurar el flujo de información entre el usuario y la aplicación.

## 2.7 Diagramas de clases del diseño

Los diagramas de clases del diseño permiten describir gráficamente las especificaciones de las clases del *software*. Muestran las clases (descripciones de objetos que comparten características comunes) que componen el sistema y cómo se relacionan entre sí (Pressman, 2005. ).

Las siguientes figuras muestran los Diagramas de Clases del Diseño con Estereotipos Web (DCDEW) de los caso de uso expuestos anteriormente en la especificación de casos de uso.

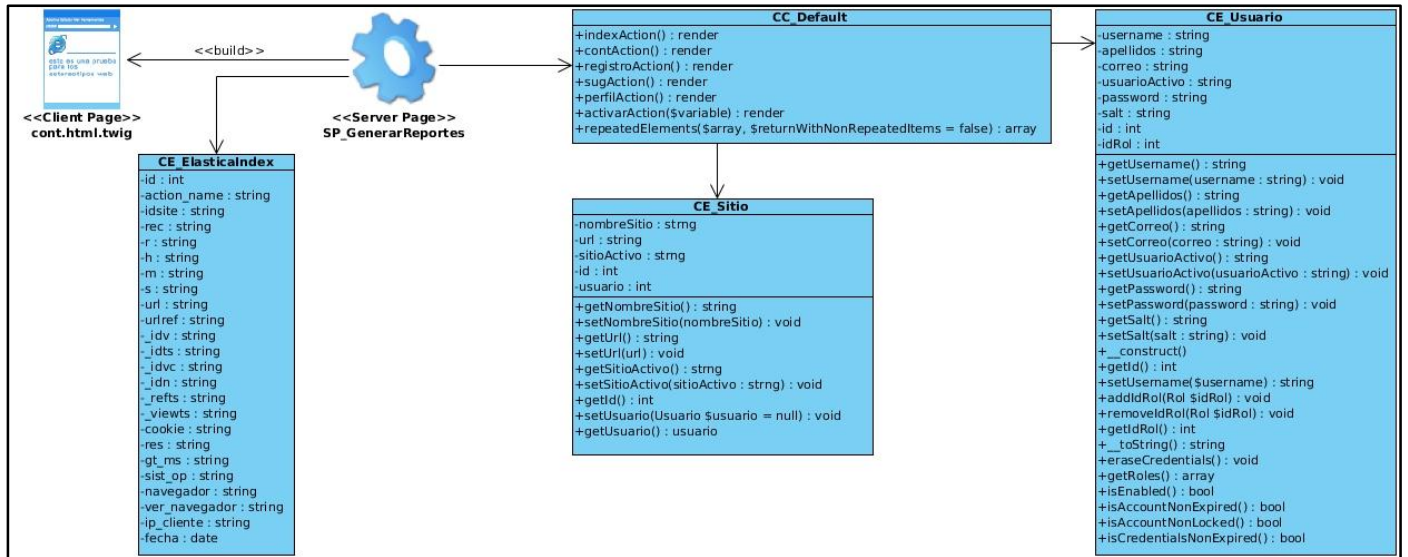


Figura 4: DCDEW para CU Generar reportes

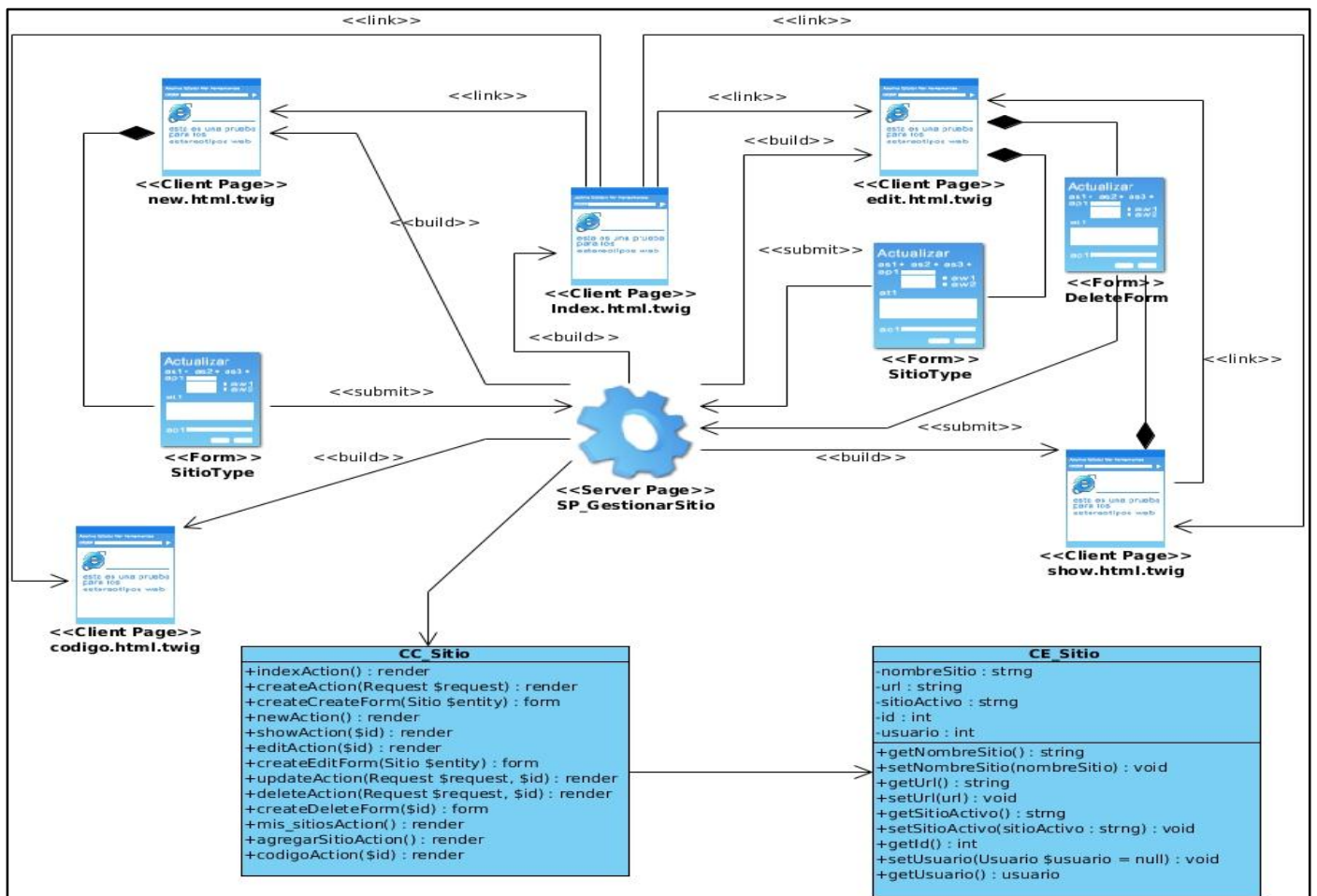


Figura 5: DCDEW para CU Gestionar sitio

En caso necesario de consultar los diagramas de clases del diseño de los demás casos de uso, puede examinar el Anexo C.

### 2.8 Diagramas de interacción

Los diagramas de interacción representan la forma en que un cliente u objeto se comunica entre sí en petición a un evento. No son sólo importantes para modelar los aspectos dinámicos de un sistema, sino también para construir sistemas ejecutables por medio de ingeniería directa e inversa (Pressman, 2005. ).

Los siguientes diagramas muestran la interacción entre las clases y los actores de los CU críticos del sistema.

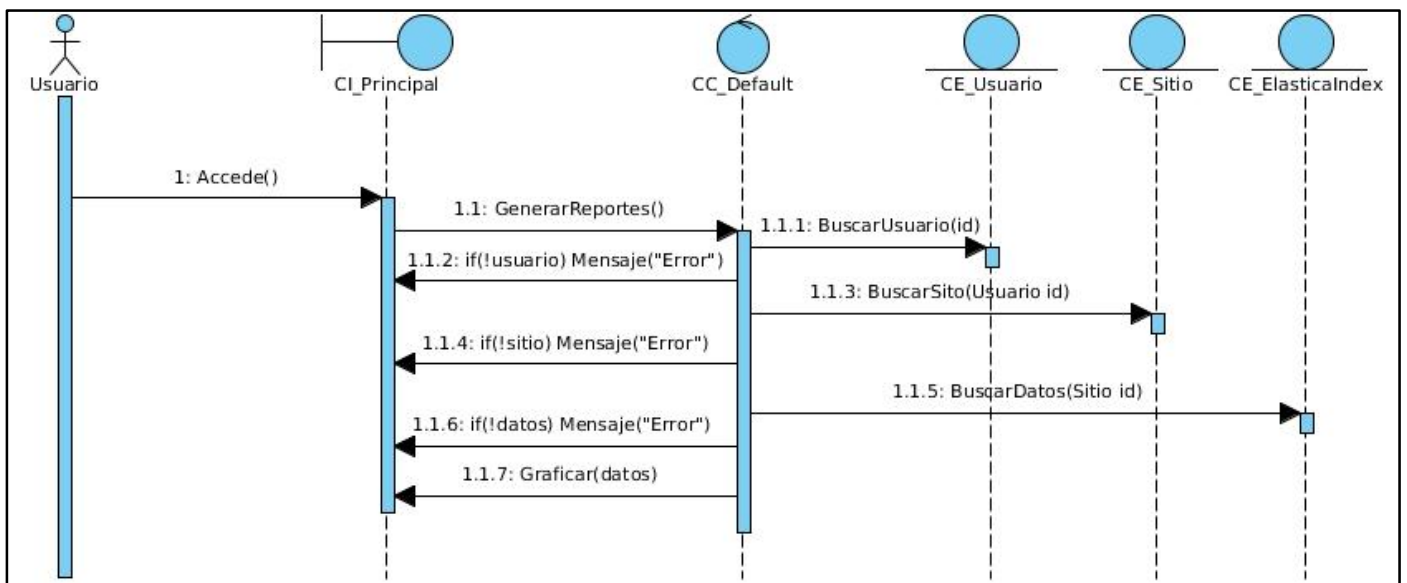


Figura 6: Diagrama de secuencia para CU Generar reportes

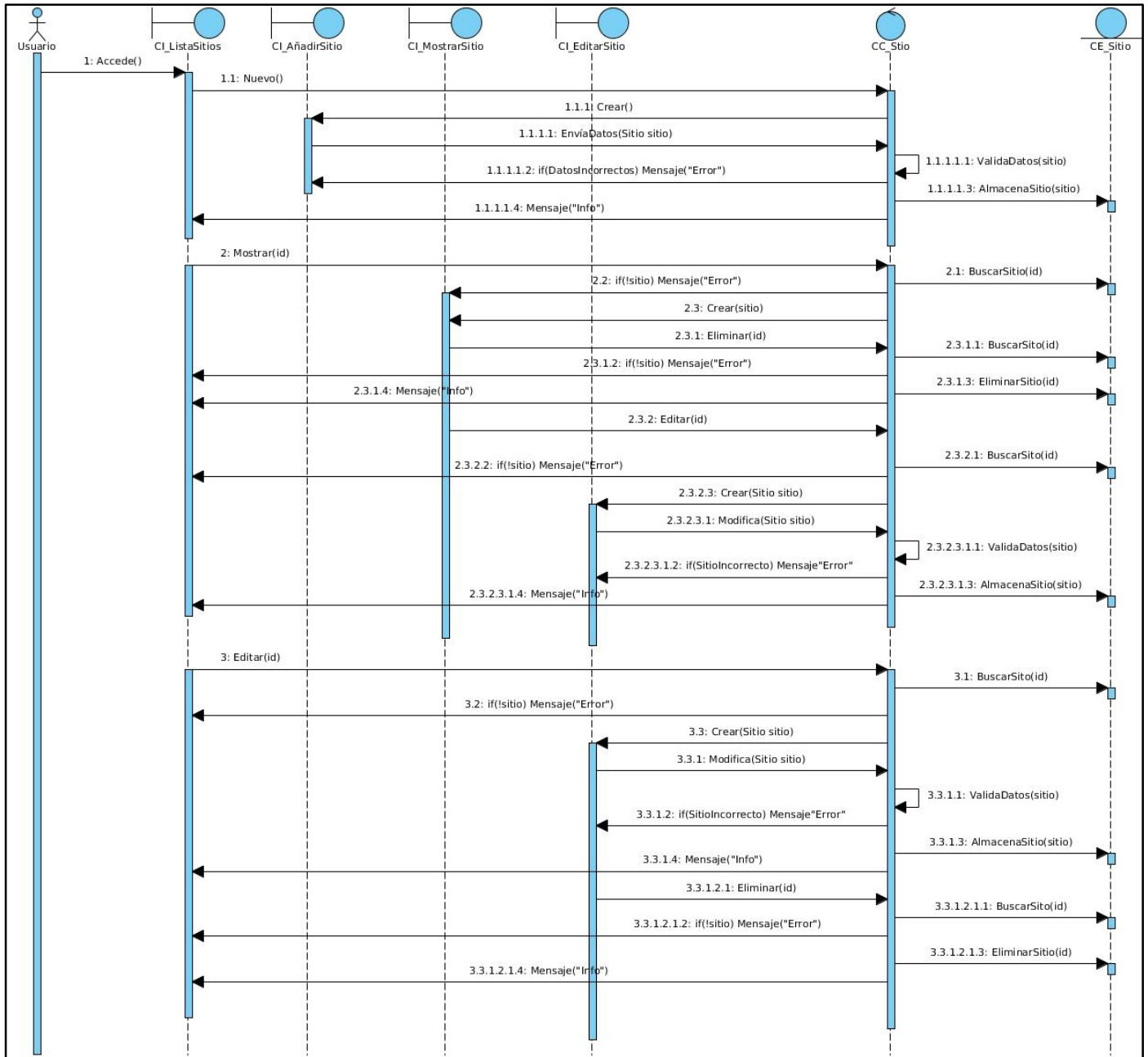


Figura 7: Diagrama de secuencia para CU Gestionar sitio

En caso necesario de consultar los diagramas de secuencia de los demás casos de uso, puede examinar el Anexo D.

### 2.9 Modelo de datos

Un modelo de base de datos es un tipo de modelo de datos que determina la estructura lógica de una base de datos y de manera fundamental determina el modo de almacenar, organizar y manipular los datos (Olga Pons, 2005).

A continuación se expondrá como quedará el modelo de datos de la solución propuesta. Teniendo en cuenta que la misma propone utilizar dos bases de datos diferentes, se dividió la explicación en dos figuras para su mejor comprensión

La Figura 8 muestra la estructura de la base de datos para los datos administrativos, la cual contiene un total de cuatro tablas: **rol**, **usuario**, **sitio** y **usuario\_rol** esta última representa la relación de mucho-a-mucho entre las tablas **rol** y **usuario**; en la figura también se puede apreciar las columnas o atributos de las mismas.

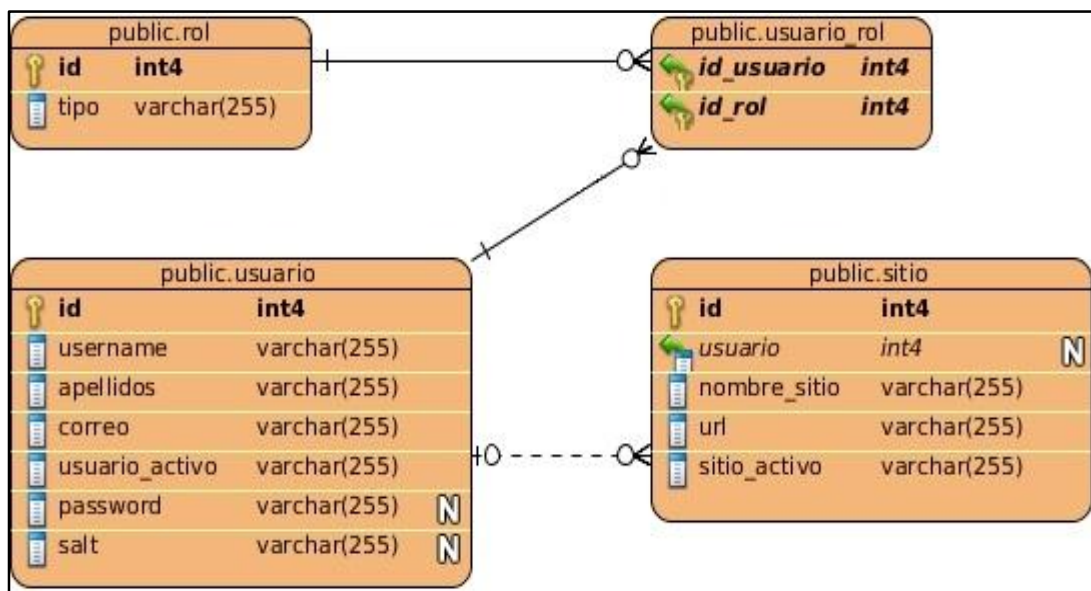


Figura 8: Modelo de datos para datos administrativos.

La Figura 9 muestra la estructura de la base de datos para los datos estadísticos. La misma cuenta con una sola tabla en la que se almacenarán todos los datos asociados a la interacción del usuario con un sitio determinado (la línea discontinua en la figura denota esta relación, aclarar que es solamente representativa), también se pueden apreciar los datos específicos que se recogen para el análisis.

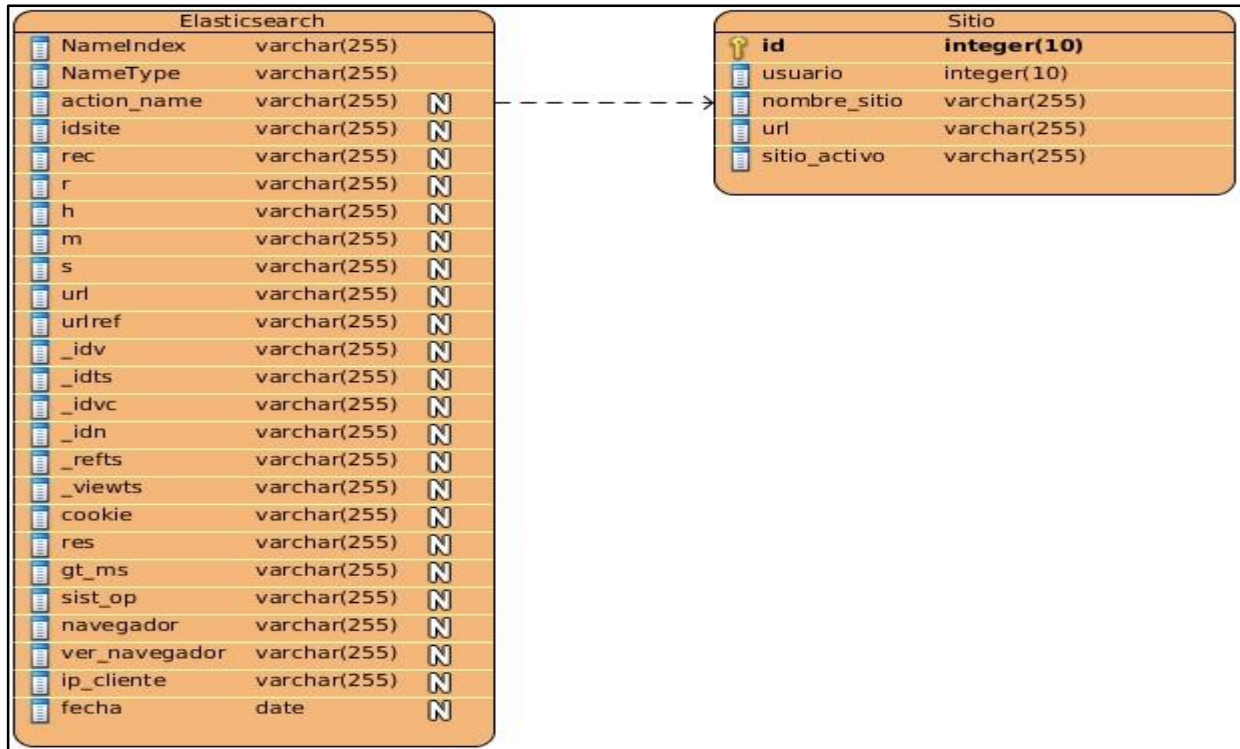


Figura 9: Modelo de datos para datos estadísticos.

## 2.10 Modelo de implementación

Este modelo representa la composición física de la implementación en términos de subsistemas de implementación. Describe cómo los elementos de diseño se implementan en componentes.

Se considera el artefacto más significativo del flujo de trabajo de Implementación, debido a la importancia que tiene para los desarrolladores comprender el funcionamiento del sistema desde el punto de vista de componentes y sus relaciones. Este modelo está conformado por los diagramas de componente y el de despliegue.

### 2.10.1 Diagrama de componente

Los diagramas de componente describen los elementos físicos del sistema y sus relaciones. Los componentes representan todos los tipos de elementos *software* que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes o bibliotecas cargadas dinámicamente. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente (Sommerville, 2005).

A continuación se muestra el diagrama de componente general del sistema:

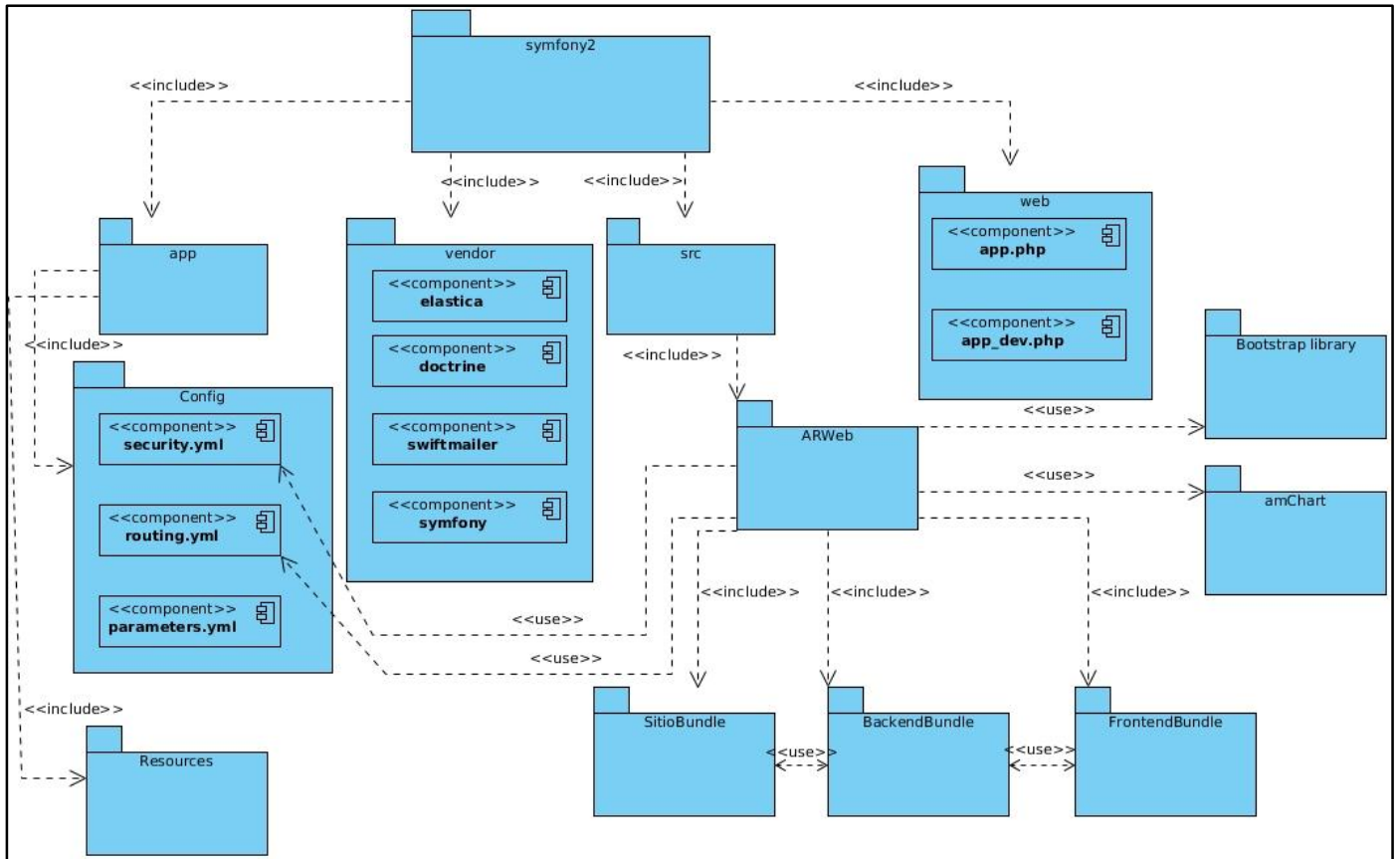


Figura 10: Diagrama de componente general.

En el Anexo E se pueden encontrar los diagramas de componentes referentes a los *bundles* desarrollados en la solución propuesta.

### 2.10.2 Diagrama de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Un nodo es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional, que generalmente tiene memoria y a menudo, capacidad de procesamiento. Los nodos se utilizan para modelar la topología del *hardware* sobre el que se ejecuta el sistema. Representan un procesador o un dispositivo sobre el que se pueden desplegar los componentes. La relación entre un nodo y el componente que despliega puede mostrarse con una relación de dependencia (Larman, 2003).

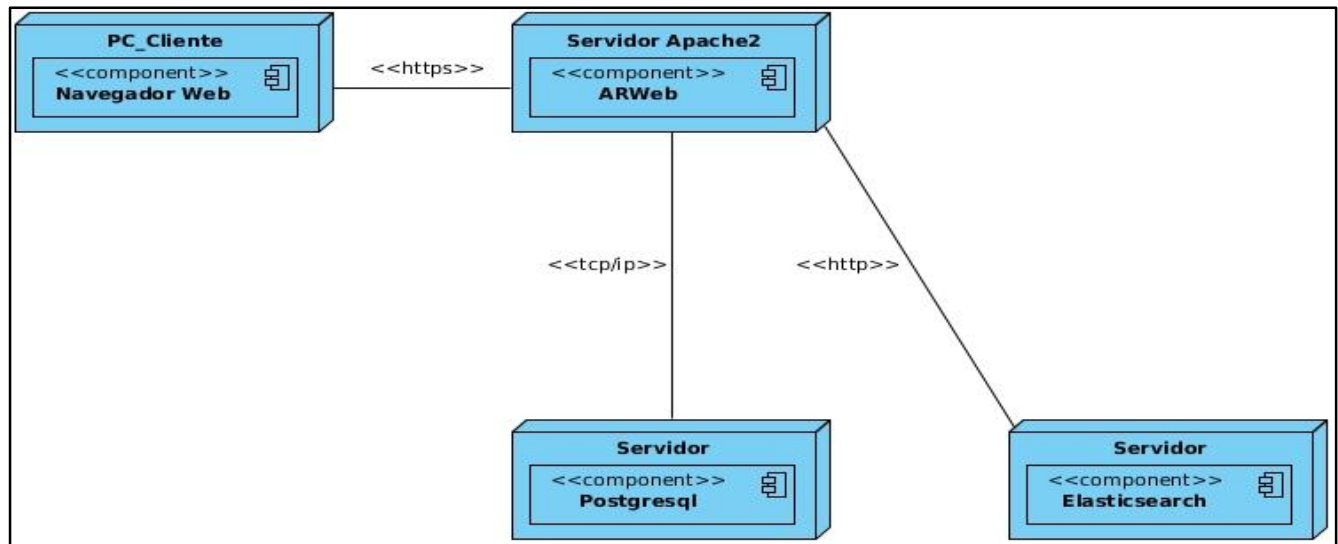


Figura 11: Diagrama de despliegue.

## 2.11 Estándares de codificación

Un estándar es un modelo, norma, patrón, referencia o la especificación que regula la realización de ciertos procesos o la fabricación de componentes para garantizar la interoperabilidad. En la implementación de la aplicación web a desarrollar, se utilizarán varios estándares de codificación, que certificarán legibilidad y organización al código de la misma, simplificando esfuerzos a la hora de darle mantenimiento y seguimiento a la aplicación.

A continuación se especifican los estándares de codificación a utilizar en la construcción de la solución:

**Camel Case:** la notación "*Camel Case*" consiste en escribir frases o palabras compuestas eliminando los espacios intermedios y poniendo en mayúscula la primera letra de cada palabra (Lago, 2008).

**Lower Camel Case:** similar al *Camel Case* sólo que la primera letra de la primera palabra es también en minúscula.

**Upper Camel Case:** similar al *Camel Case* siendo la primera letra de la primera palabra en mayúscula.

El nombre de las clases debe ser declarado utilizando el estilo *Upper Camel Case* y cuando en un fichero se define el espacio de nombre (*namespace*) debe existir una línea en blanco a continuación de la declaración. Además cuando están presentes todas las declaraciones de uso se escriben a continuación de la definición del *namespace* y se debe utilizar la palabra reservada *use* una vez por cada declaración además debe concluir con una línea en blanco.



```

<?php

namespace airesWeb\SitioBundle\Controller;

use Symfony\Component\HttpFoundation\Request;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Elastica\Facet\DateHistogram;
use Elastica\Query;

class CodigoController extends Controller
{
    public function createindexAction()
    {
        //create new instance
        $elasticaClient = new \Elastica\Client();
        // Load index
        $elasticaIndex = $elasticaClient->getIndex('mi_indice');
        if (!$elasticaIndex->exists())
        {
            // Create the index new
            $elasticaIndex->create(array(), true);
            return $elasticaIndex;
        }

        return $elasticaIndex;
    }
}

```

Figura 12: Nombre de las clases, namespace y use.

El nombre de las funciones debe ser declarado utilizando el estilo *Lower Camel Case*.

```

class CodigoController extends Controller
{
    public function createindexAction()
    {
        {...15 lines }
    }

    public function createtypeAction(\Elastica\Index $elasticaIndex)
    {
        {...46 lines }
    }

    public function addAction(Request $request)
    {
        {...85 lines }
    }

    public function detalleAction()
    {
        {...46 lines }
    }

    public function pruebaAction()
    {

```

Figura 13: Nombre las funciones en *Lower Camel Case*.

Los atributos de las clases deben ser privados, además de que no debe existir más de un atributo por línea y deben utilizar el estilo *Lower Camel Case*.

```
private $id;

/**
 * @var string
 *
 * @ORM\Column(name="username", type="string", nullable=false)
 * @Assert\NotBlank(message = "Por favor, escriba su nombre")
 */
private $username;

/**
 * @var string
 *
 * @ORM\Column(name="apellidos", type="string", nullable=false)
 * @Assert\NotBlank(message = "Por favor, escriba sus apellidos")
 */
private $apellidos;

/**
 * @var string
 * @ORM\Column(name="correo", type="string", nullable=false)
 * @Assert\Email(checkMX=true, message = "No es válida la dirección de correo")
 */
private $correo;

/**
```

Figura 14: Atributos en Lower Camel Case.

## 2.12 Conclusiones

La modelación de los diferentes diagramas de clases del diseño, secuencia, modelo de la base de datos, despliegue y casos de uso, junto con el eficiente levantamiento de los requisitos, ayudaron al equipo de desarrollo a entender con más facilidad el problema planteado. Los patrones de diseño escogidos, propiciaron una arquitectura sólida y robusta al sistema. Como resultado de la implementación y de las facilidades brindadas por el diseño se obtuvo un sistema funcional, completamente operativo. En el cual se aplicaron diferentes mecanismos de seguridad con el fin de garantizar confiabilidad e integridad a los servicios prestados por el mismo. También se especificaron los estándares de codificación utilizados en el proceso de implementación, lo que facilitó la lectura, comprensión y mantenimiento del código fuente generado.

## Capítulo 3: Validación y Prueba del Sistema

### 3.1. Introducción

Las pruebas son un conjunto de actividades que se planean con anticipación y se realizan de manera sistemática. Por tanto, se debe definir una plantilla para las pruebas de software (un conjunto de pasos en que se puedan incluir técnicas y métodos específicos del diseño de casos de pruebas) (Pressman, 2005).

Las pruebas se realizan con el objetivo de encontrar errores cometidos al realizar el diseño y construcción de un producto de *software*. A la aplicación ARWeb se le hicieron pruebas de carga y estrés, de seguridad y funcionales, a continuación se detallan los principales resultados de la aplicación de las mismas.

### 3.2. Pruebas funcionales

Las pruebas funcionales están basadas en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el *software*. Las mismas se hacen mediante el diseño de modelos de prueba que buscan evaluar cada una de las opciones con las que cuenta el paquete informático. Son pruebas específicas, concretas y exhaustivas para comprobar y validar que el *software* hace lo que debe y sobre todo, lo que se ha especificado (Adame, 2009).

Se ejecutaron este tipo de prueba para todos los casos de uso con diferentes entradas del usuario, con el objetivo de determinar que los resultados obtenidos fueran los esperados bajo cualquier situación, y así, dar por cumplidos los requerimientos del cliente. A continuación se presenta una muestra de los casos de pruebas que se elaboraron para cada caso de uso:

#### Descripción de las variables

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	nombre	Campo de texto	No	Se debe especificar al menos una palabra.
2	url	Campo de texto	No	Debe ser una dirección url válida

Tabla 4: Descripción de las variables.

#### SC Nuevo

Escenario	Descripción	Variable: nombre	Variable: url	Respuesta del sistema	Flujo central
EC 1.1	El usuario introduce todos los	V	V	Almacena la información y	El usuario llena todos los campos de manera

Insertar Sitio correctamente	valores de manera correcta en el sistema.			muestra el mensaje: "¡Aviso! Sitio añadido correctamente"	correcta y da clic en el botón "Crear".
EC 1.2 Campo(s) incorrecto(s)	El usuario ingresa datos incorrectos.	I	I	Selecciona el campo(s) incorrecto(s) y muestra el mensaje: "¡Error! No se ha podido añadir el sitio, error en las entradas".	El usuario llena los campos de manera incorrecta y da clic en el botón "Crear".
		I	V		
		V	I		

Tabla 5: Prueba funcional para el CU gestionar sitio, sección: Nuevo.

Las celdas de la tabla contienen V, I, o N/A; V indica válido, I indica inválido, y N/A que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante.

Estas pruebas fueron realizadas a la aplicación, arrojando como resultado un total de 11 no conformidades agrupadas en 4 tipos: Error-Idioma, Redacción-Aplicación, Ortografía-Aplicación y Funcionalidad. La ejecución de estas pruebas se distribuyó en 2 iteraciones, en una primera iteración donde se encontraron las 11 no conformidades y en una segunda en la cual no se detectaron (Ver figura 15).

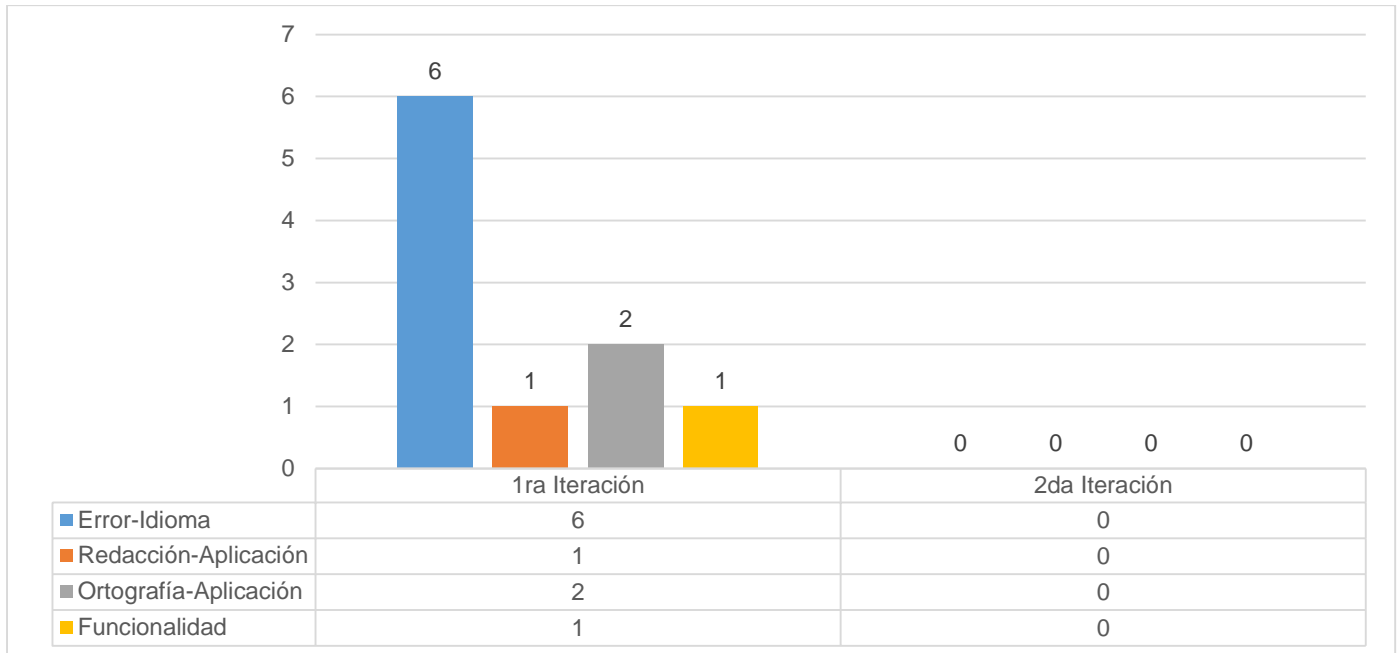


Figura 15: Resultados de las pruebas funcionales.

### 3.3. Pruebas de carga y estrés

Estas pruebas se realizan para medir la respuesta de las aplicaciones a los distintos volúmenes de carga esperados (cantidad de usuarios y/o peticiones) y cómo responde el mismo frente a dicha carga. Para ello se utilizó la herramienta *jMeter*, con el objetivo de realizar mediciones exactas, revisar el comportamiento funcional y medir el rendimiento de la misma. La realización de dichas pruebas fueron divididas en varios *tests* de 50, 100 y 150 hilos cada uno, los cuales simulan la cantidad de usuarios que acceden a las funcionalidades concurrentemente. Las mismas se realizaron sobre todas las funcionalidades de la aplicación, permitiendo comprobar su correcto desempeño dentro de la misma, a continuación se presentan los resultados para un grupo de funcionalidades específicas escogidas por el equipo de desarrollo, teniendo en cuenta su complejidad e importancia dentro del sistema. Es de válida aclaración que las pruebas fueron realizadas en un servidor de bajas prestaciones con características: sistema operativo Linux-Ubuntu 12.04, microprocesador *Celeron* a 2.50 GHz y memoria RAM de 1GB, considerablemente inferior a los descritos en los requisitos.

Funcionalidad	Hilos	# Muestras	Min	Max	% Error	Rendimiento
Añadir Sitio.	50	200	554	5999	0.00	18.6/seg
	100	400	724	10689	0.00	21.4/seg

	150	600	706	20170	0.00	20.5/seg
Solicitar Cuenta.	50	100	233	6462	0.00	10.3/seg
	100	200	1358	9206	0.00	13.7/seg
	150	300	153	39294	0.00	7.2/seg
Generar Reportes.	50	1400	0	4336	0.00	212.8/seg
	100	2800	0	8095	0.00	208.3/seg
	150	4200	1	12614	0.00	209.8/seg
Buscar Estadísticas de Sitio.	50	50	114	1528	0.00	21.8/seg
	100	100	348	3742	0.00	21.7/seg
	150	150	290	6079	0.00	21.2/seg

Tabla 6: Resultados de las pruebas con jMeter.

**Hilos:** Total de usuarios concurrentes.

**# Muestras:** Total de peticiones realizadas al servidor.

**Min:** Tiempo (mili-segundos) mínimo de respuesta de la aplicación.

**Max:** Tiempo (mili-segundos) máximo de respuesta de la aplicación.

**% Error:** Porcentaje de error en la respuesta de la aplicación al total de peticiones.

**Rendimiento:** Tiempo (segundos) de respuesta de la aplicación por cada petición.

### 3.4. Pruebas de seguridad

Las pruebas de seguridad comprueban que los mecanismos de protección integrados en el sistema realmente lo protejan de irrupciones inapropiadas. En ellas se debe intentar conseguir las claves de acceso por cualquier medio, deben producir a propósito errores del sistema para ser corregidos luego por los desarrolladores del sistema (Quality, 2011).

Para garantizar la seguridad del sistema se realizaron pruebas de seguridad mediante una lista de chequeo ajustable a aplicaciones web. El objetivo general de esta lista es evaluar a través de indicadores la seguridad de las aplicaciones en un primer nivel, establecido por las especialistas del grupo de seguridad del Departamento de Pruebas de Software (DEPSW) de la UCI, dichas pruebas cuentan con 15 indicadores separados en 4 tipos de pruebas. Los resultados de las mismas se exponen a continuación.

Forma de Uso de la Lista de Chequeo:

**Peso:** Define si el indicador a evaluar es crítico o no.

**Evaluación (Eval):** Es la forma de evaluar el indicador en cuestión. El mismo se evalúa de 1 en caso de mal y 0 en caso que elemento revisado no presente errores.

**Cantidad de elementos afectados:** Especifica la cantidad de errores encontrados sobre el mismo indicador.

**Comentario:** Especifica los señalamientos o sugerencias que quiera incluir la persona que aplica la lista de chequeo.

**Indicadores a Evaluar:** Descripción del parámetro a evaluar.

**N.P. (No Procede):** Se usa para especificar que el indicador a evaluar no se puede aplicar en ese caso.

Estructura de la lista de chequeo:

Pruebas de Autorización					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
Crítico	Puede un usuario estándar (no administrador) modificar sus privilegios en la aplicación.	0		0	
Crítico	Puede un usuario estándar (no administrador) modificar los privilegios de otro usuario.	0		0	

Tabla 7: Lista de chequeo prueba de autorización.

Pruebas de Gestión de Sesiones					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
Crítico	Copiar la URL de la aplicación estando	0		0	

	autenticado, cerrar el navegador y volver a abrirlo, pegar la dirección copiada anteriormente, la aplicación permite que el usuario conserve su estado de autenticado..				
Crítico	Al cerrar la sesión de un usuario y dar clic en el botón del navegador "Atrás" la aplicación vuelve entrar a la sesión autenticada.	0		0	

Tabla 8: Lista de chequeo prueba de gestión de sesión.

Comprobación del Sistema de Autenticación					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
Crítico	Se cierra la sesión del usuario después de un tiempo X (establecido por la aplicación) estando sin usar la misma.	0		0	10 minutos de inactividad.
Crítico	Se bloquea la cuenta del usuario después de un número X (establecidos por la aplicación) de intento de login fallidos por el usuario. De ser así definir la cantidad de		NP		



	intentos en la columna Comentarios.				
Medio	Se muestran diferentes mensajes de error al usuario “intentar autenticarse con un correo correcto y una contraseña incorrecta” y “al autenticarse con un correo incorrecto y la misma contraseña del intento anterior”.	1		1	
Crítico	El mensaje devuelto al intentar autenticarse con un correo correcto y contraseña incorrecta, es el mismo al intento con correo y contraseña incorrecta, si es el mismo intento verificar el tiempo de respuesta, pues podría apreciarse una media ligeramente diferente en el tiempo de respuesta, que igualmente puede usarse esta información para descubrir cuentas válidas.	0		0	El tiempo de respuesta es similar en ambos casos.
Crítico	El campo correo de la	0		0	

	autenticación al sistema tiene el auto completamiento activado (guarda los correos que se autentican). Para ver esto se debe autenticar al sistema, luego salir y poner la primera letra del correo autenticado para ver si muestra el nombre de usuario completo.				
Crítico	El sistema protege el envío de los datos mediante protocolo seguro ( <i>https</i> ).	0		0	
Crítico	El sistema usa algún certificado.	0		0	

Tabla 9: Lista de chequeo comprobación del sistema de autenticación.

Validación de Datos					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
Crítico	Se enmascaran datos confiables cuando se visualicen en la aplicación (Por ejemplo: Contraseñas)	0		0	
Medio	La aplicación solamente	1		1	Permite cualquier tipo de contraseña con la

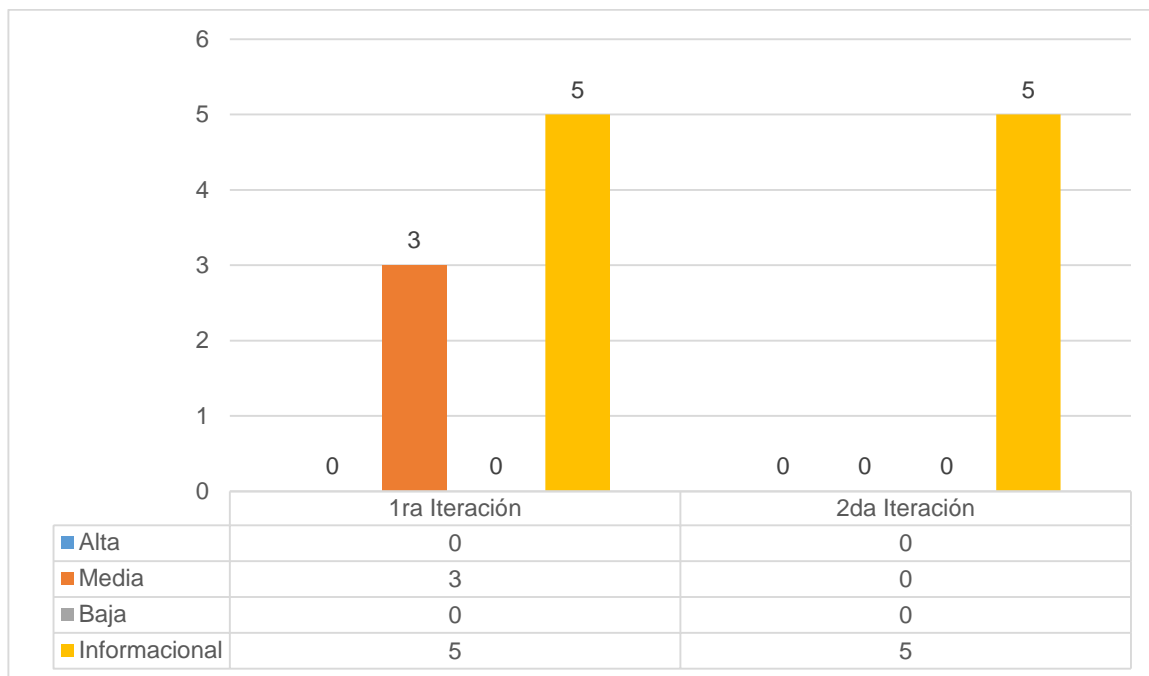
	permite contraseñas alfanuméricas, que incluyan caracteres especiales y que tengan seis caracteres mínimos de longitud.				restricción de 6 caracteres como mínimo.
Crítico	La aplicación permite la funcionalidad de cambio de contraseña únicamente a usuarios autenticados validando la antigua contraseña, la nueva contraseña y la respuesta a la pregunta de seguridad (opcional en dependencia del proyecto).	1		1	No valida la contraseña anterior.
Crítico	El sistema muestra algún mensaje indebido, al colocar en la barra de dirección o en campos de entrada los caracteres: comillas simples	0		0	

**Tabla 10: Lista de chequeo validación de datos.**

La aplicación de la lista de chequeo arrojó un total de tres no conformidades: una dentro de la clasificación “Comprobación del Sistema de Autenticación” y dos dentro de “Validación de Datos”; determinándose que la relacionada con el tipo de contraseñas permitidas por la aplicación y los diferentes mensajes de error mostrados por la aplicación en respuesta a una autenticación fallida no proceden, por otra parte se eliminó la restante.

También para evaluar la seguridad del sistema se utilizó la herramienta *Acunetix Web Vulnerability Scanner 8.0*, la cual es una herramienta que es capaz de escanear sitios web en busca de posibles fallos de seguridad que puedan poner en peligro la integridad de la página publicada en internet. Esta aplicación ejecuta una serie de pruebas, totalmente configurables por el usuario, para identificar las vulnerabilidades tanto en la programación de la página como en la configuración del servidor (Marco de desarrollo de la Junta de Andalucía. *Acunetix Web Vulnerability Scanner* (2014)).

Esta herramienta agrupa las vulnerabilidades encontradas en cuatro categorías: Alta, Media, Baja e Informativa. Se realizaron un total de dos iteraciones de pruebas, la primera arrojó un total de tres vulnerabilidades de tipo Media, y cinco de tipo Informativa, estas últimas son referentes al autocompletamiento de los campos “Correo” y “Contraseña” en los formularios de *login* con que cuenta la aplicación, las cuales se determinaron que no procedían pues se implementó dentro del sistema la opción “Recuérdame”, para brindar mayor facilidad a los usuarios del mismo. En una segunda iteración solo se detectaron las cinco vulnerabilidades de tipo Informativa (Ver Figura 12).



**Figura 16: Resultados de las pruebas de seguridad con *Acunetix*.**

### 3.5. Conclusiones

La realización de pruebas funcionales, de seguridad y de carga y estrés permitieron la detección de errores y problemas para su posterior solución, además brindaron la posibilidad de monitorear el funcionamiento de

la aplicación en diferentes entornos de ejecución y carga, con el objetivo de determinar posibles mejoras al rendimiento y capacidad de la misma.

## **Conclusiones**

El presente trabajo dio cumplimiento a cada uno de los objetivos trazados, pudiéndose destacar de manera general, las conclusiones siguientes:

1. El estudio de las herramientas de analítica web en el ámbito nacional y extranjero, las tendencias actuales de las mismas y las exigencias del cliente, permitieron elegir las herramientas, tecnologías, los lenguajes y metodología para guiar el proceso de desarrollo.
2. Los artefactos y actividades generados, basados en la metodología seleccionada, así como las entrevistas realizadas al cliente permitieron un mejor dominio del problema planteado, con vistas a la implementación del sistema.
3. Como resultado de la implementación y dándole cumplimiento al objetivo general de la investigación se obtuvo un Analizador de Registros Web (ARWeb), para monitorizar en tiempo real las acciones en los sitios web cubanos y apoyar la toma de decisiones en organizaciones nacionales.
4. La aplicación de disímiles pruebas permitió validar el correcto funcionamiento de la aplicación desarrollada, simulando ambientes reales en diferentes entornos de ejecución.

## **Recomendaciones**

Al término de esta investigación se recomienda:

1. Incluir funcionalidades de envío automático de reportes por correo.
2. Construir un sistema de detección de anomalías (mucho, poco tráfico, muchos errores).
3. Permitir que el usuario defina reglas y que sea notificado cuando las mismas se cumplan (sistema de alertas personalizado).

## Bibliografía referenciada

**Adame, Javier Venegas. 2009.** *AUTOMATIZACIÓN DE TESTING*. Barcelona : s.n., 2009.

**Ana P. Rodriguez, Josué Polanco, Darwin Hernández. 2013.** *Open UP* . 2013.

**Bowen, Rich. 2007.** *What's New in Apache Web Server 2.2?* 2007.

**Byteflair. 2013.** Byteflair. [En línea] 2013. [Citado el: 15 de Mayo de 2014.]  
<http://byteflair.com/2013/07/comenzando-con-elasticsearch/>.

**Carrasco, Nicolas. 2014.** Prezi. [En línea] 2014. [Citado el: 22 de Febrero de 2014.]  
<http://prezi.com/eunnrsm1pfba/copy-of-informatica-forense/>.

**Chavez, Miguel Angel. 2012.** *AIRESWEB: SISTEMA DE REPORTES Y ANÁLISIS INTELIGENTE DE REGISTROS WEB*. 2012.

**Eguiluz, Javier. 2013.** *Desarrollo web ágil con Symfony2*. Madrid : s.n., 2013.

**Fabien Potencier, Ryan Weaver. 2014.** *Symfony 2.3, el libro oficial*. 2014.

**Ganazhapa, Ing. Jenny Alexandra Pujos. 2012.** *EL SITIO WEB Y SU INCIDENCIA EN LA DIFUSIÓN Y PROMOCIÓN DEL CENTRO EDUCATIVO ALBERT EINSTEIN UBICADA EN LA PARROQUIA CIUDAD NUEVA DEL CANTÓN PÍLLARO PROVINCIA DE TUNGURAHUA*. 2012.

**Garcia, Gines Bravo. 2010.** *PHP 5 A traves de ejemplos*. 2010.

**Gaita, Alvaro Martínez. 2009.** desarrolloweb.com. [En línea] 2009. [Citado el: 22 de Marzo de 2014.]  
[http://www.desarrolloweb.com/de\\_interes/chartbeat-trafico-web-tiempo-real-3176.html](http://www.desarrolloweb.com/de_interes/chartbeat-trafico-web-tiempo-real-3176.html).

**Genveta:dev. 2012.** Bootstrap, framework de twiter. [En línea] 16 de Junio de 2012. [Citado el: 4 de Mayo de 2014.] <http://www.genbetadev.com/frameworks/bootstrap>.

**Gonsález, Eduardo Sánchez. 2014.** EstrategasWeb. [En línea] Marzo de 2014. [Citado el: 23 de Mayo de 2014.] <http://estrategasweb.com/wp-content/uploads/2014/03/Introducci%C3%B3n-Anal%C3%ADtica-Web-Parla2.pdf>.

**Google.** Sistemas viables. [En línea] [Citado el: 3 de Marzo de 2014.]  
<https://www.google.com/cu/#q=Google+Analytics+ayuda+a+dise%C3%B1ar+anuncios+m%C3%A1s+orientados+a+mejorar+las+iniciativas+de+marketing+y+a+crear+sitios+web+que+generen+m%C3%A1s+conversiones>.

**Groussard, Thierry. 2010.** *Recursos Informáticos Java Enterprise Edition*. Barcelona : s.n., 2010.

**Hipertextual. 2013.** Bitelia. [En línea] 28 de Mayo de 2013. [Citado el: 10 de Febrero de 2014.]  
<http://bitelia.com/2013/05/entendiendo-html5-guia-para-principiantes>.



**2014.** Junta de Andalucía. [En línea] 2014. [Citado el: 11 de Mayo de 2014.]  
<http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/656>.

**Kaushik, Avinash. 2011.** *Analítica Web 2.0.* s.l. : Gestión 2000, 2011.

**Lago, Diego. 2008.** Guía de estilode programación. [En línea] 2008. [Citado el: 4 de Mayo de 2014.]  
<http://did.cexto.com>.

**Larman, Craig. 2003.** *UML y patrones Tomo II.* s.l. : PEARSON, 2003.

—. **2003.** *UML y Patrones. 2da Edición.* s.l. : Prentice Hall, 2003. ISBN: 84-205-3438-2.

**LibrosWeb. 2014.** LibrosWeb. [En línea] 2014. [Citado el: 16 de Febrero de 2014.]  
[http://librosweb.es/javascript/capitulo\\_1/breve\\_historia.html](http://librosweb.es/javascript/capitulo_1/breve_historia.html).

**Lima, Fernando. 2014.** SlidePlayer. [En línea] 2014. [Citado el: 7 de Febrero de 2014.]  
<http://slideplayer.es/slide/143422/>.

**Maldonado, Sergio. 2012.** *Análítica web. Medir para triunfar.* s.l. : ESIC, 2012.

**Miller, Stephan A. 2012.** *Piwik Web Analytics Essentials.* 2012.

**Ojeda, Antonio Navajas. 2012.** *Guía Completa de CSS3.* s.l. : Autoedición, 2012.

**Olga Pons, Nicolas Marin. 2005.** *Introducción a las bases de datos: el modelo relacional.* s.l. : Paraninfo, 2005.

**Oracle. 2013.** NetBeans. [En línea] Oracle, 2013. [Citado el: 6 de Mayo de 2014.]  
[https://netbeans.org/index\\_es.html](https://netbeans.org/index_es.html).

**Paéz, Pablo G. 2010.** El mayor portal de gerencia. [En línea] 2010. [Citado el: 3 de Abril de 2014.]  
<http://www.elmayorportaldegerencia.com/Publicaciones/%5BPD%5D%20Publicaciones%20-%20Metodologia%20para%20una%20toma%20de%20decisiones%20efectiva.pdf>.

**Peter Eisentraut, Bernd Helmle. 2008.** *PostgreSQL-Administration.* 2008.

**2009.** Posicionamiento y Marketing en Buscadores. [En línea] 2009. [Citado el: 4 de Febrero de 2014.]  
[http://pos-y-marketing-en-buscadores-2013.blogspot.com/2013\\_04\\_01\\_archive.html](http://pos-y-marketing-en-buscadores-2013.blogspot.com/2013_04_01_archive.html).

**Pressman, Roger. 2005.** *Ingeniería del Software. Un enfoque práctico. Sexta Edición.* s.l. : INTERAMERICANA DE MEXICO, 2005. ISBN 9789701054734.

—. **2005.** . *Un enfoque práctico a la Ingeniería de Software.* 2005. .

**Quality, V&V. 2011.** Pruebas de seguridad. [En línea] 2011. [Citado el: 24 de Mayo de 2014.]  
[http://www.vvquality.com/w1/index.php?option=com\\_content&view=article&id=79&Itemid=151](http://www.vvquality.com/w1/index.php?option=com_content&view=article&id=79&Itemid=151).

**Sebastián, J. 2010.** *Modelo Vista Controlador- Definición y Características.* 2010.

**Softbull. 2012.** Softbull. [En línea] TelFo Networks S.L., 2012. [Citado el: 5 de Abril de 2014.] <http://amcharts.softbull.com/>.

**Sommerville, Ian. 2005.** *Ingeniería de Software*. 2005.

**Suárez, Pablo Javier Kabytes. 2010.** Excelentes plugins para crear gráficas con jquery. [En línea] 8 de Octubre de 2010. [Citado el: 30 de enero de 2014.] <http://www.kabytes.com/programacion/4-excelentes-plugins-para-crear-graficas-con-jquery/>.

**Vea, Andreu. 2013.** *Cómo creamos Internet*. s.l. : Península, 2013.

**Zambrano, Renán Alexi Briones. 2013.** *Metodología de Investigación*. 2013.

## Bibliografía consultada

- Muñoz, Gemma y Elósegui, Tristán. 2011.** *El Arte de Medir. Manual de Analítica Web.* s.l. : Profit, 2011. ISBN: 9788492956647.
- Adame, Javier Venegas. 2009.** *AUTOMATIZACIÓN DE TESTING.* Barcelona : s.n., 2009.
- Ana P. Rodriguez, Josué Polanco, Darwin Hernández. 2013.** *Open UP .* 2013.
- ASSETS. [En línea] <http://www.assets.co.cu/assets.asp>.
- Bowen, Rich. 2007.** *What's New in Apache Web Server 2.2?* 2007.
- Byteflair. 2013.** Byteflair. [En línea] 2013. [Citado el: 15 de Mayo de 2014.] <http://byteflair.com/2013/07/comenzando-con-elasticsearch/>.
- Carrasco, Nicolas. 2014.** Prezi. [En línea] 2014. [Citado el: 22 de Febrero de 2014.] <http://prezi.com/eunnrsm1pfba/copy-of-informatica-forense/>.
- Celaya, Javier. 2008.** *La empresa en la web 2.0 .* s.l. : Gestión 2000, 2008.
- Chavez, Miguel Angel. 2012.** *AIRESWEB: SISTEMA DE REPORTES Y ANÁLISIS INTELIGENTE DE REGISTROS WEB.* 2012.
- Clifton, Brian. 2012.** *Advanced Web Metrics with Google Analytics.* 2012. ISBN-10: 1118168445.
- Cuervo, Víctor. 2012.** aulambra.com. *Tecnologías de Internet.* [En línea] 28 de febrero de 2012. [Citado el: 16 de enero de 2014.] <http://www.aulambra.com/novedades/programacion/dojo-1-7/>.
- Eguiluz, Javier. 2013.** *Desarrollo web ágil con Symfony2.* Madrid : s.n., 2013.
- . 2008. *Introducción a JavaScript.* 2008.
- Eguiluz, Javier y Potencier, Fabian. 2011.** *Desarrollo web ágil con Symfony 2.* 2011.
- Fabien Potencier, Ryan Weaver. 2014.** *Symfony 2.3, el libro oficial.* 2014.
- Fowler, M. 2003.** *Web Server Patterns. Model View Controller, Patterns of Enterprise Application Architecture.* 2003.
- G. Figeroa, Robert, J. Solis, Camilo y A. Cabrera, Armando. 2008.** *Metodologías tradicionales vs. metodologías ágiles.* 2008.
- Ganazhapa, Ing. Jenny Alexandra Pujos. 2012.** *EL SITIO WEB Y SU INCIDENCIA EN LA DIFUSIÓN Y PROMOCIÓN DEL CENTRO EDUCATIVO ALBERT EINSTEIN UBICADA EN LA PARROQUIA CIUDAD NUEVA DEL CANTÓN PÍLLARO PROVINCIA DE TUNGURAHUA.* 2012.
- García, Gines Bravo. 2010.** *PHP 5 A traves de ejemplos.* 2010.

- Gaita, Alvaro Martínez. 2009.** desarrolloweb.com. [En línea] 2009. [Citado el: 22 de Marzo de 2014.] [http://www.desarrolloweb.com/de\\_interes/chartbeat-traffic-web-tiempo-real-3176.html](http://www.desarrolloweb.com/de_interes/chartbeat-traffic-web-tiempo-real-3176.html).
- Genveta:dev. 2012.** Bootstrap, framework de twitter. [En línea] 16 de Junio de 2012. [Citado el: 4 de Mayo de 2014.] <http://www.genbetadev.com/frameworks/bootstrap>.
- Godoy Jiménez, José Manuel. 2002.** Diseño de proyectos de software en código abierto. [En línea] 2002. <http://www.ibiblio.org/pub/linux/docs/LuCaS/Tutoriales/doc-dise%F1o-software/doc-dise%F1o-software-parte-1.pdf>.
- Gonsález, Eduardo Sánchez. 2014.** EstrategasWeb. [En línea] Marzo de 2014. [Citado el: 23 de Mayo de 2014.] <http://estrategasweb.com/wp-content/uploads/2014/03/Introducci%C3%B3n-Anal%C3%ADtica-Web-Parla2.pdf>.
- Google.** Sistemas viables. [En línea] [Citado el: 3 de Marzo de 2014.] <https://www.google.com/cu/#q=Google+Analytics+ayuda+a+dise%C3%B1ar+anuncios+m%C3%A1s+orientados%2C+a+mejorar+las+iniciativas+de+marketing+y+a+crear+sitios+web+que+generen+m%C3%A1s+conversiones>.
- Groussard, Thierry. 2010.** *Recursos Informáticos Java Enterprise Edition*. Barcelona : s.n., 2010.
- Gutiérrez, Javier J.** Departamento de Lenguajes y Sistemas Informáticos. *Departamento de Lenguajes y Sistemas Informáticos*. [En línea] [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf).
- Herrera, Daniel Ernesto Navarro. 2011.** LaWebera.es. ¿Cómo darle un uso apropiado a CSS3? [En línea] 31 de octubre de 2011. [Citado el: 2013 de diciembre de 19.] <http://www.lawebera.es/maquetacion-web/darle-uso-apropiado-css3.php#..>
- . 2012. LaWebera.es. *MooTools vs otros frameworks. Comparativa*. [En línea] 14 de febrero de 2012. [Citado el: 15 de enero de 2014.] <http://www.lawebera.es/mootools/mootools-vs-otros-frameworks-comparativa.php>.
- Hipertextual. 2013.** Bitelia. [En línea] 28 de Mayo de 2013. [Citado el: 10 de Febrero de 2014.] <http://bitelia.com/2013/05/entendiendo-html5-guia-para-principiantes>.
- Jacobson, Booch y J, Rumbaugh. 2000.** *El Proceso Unificado de Desarrollo de Software*. s.l. : Addison Wesley, 2000.
2014. Junta de Andalucía. [En línea] 2014. [Citado el: 11 de Mayo de 2014.] <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/656>.
- Kaushik, Avinash. 2011.** *Analítica Web 2.0*. s.l. : Gestión 2000, 2011.
- . 2011. *Analítica Web 2.0*. s.l. : Gestión 2000, 2011. ISBN: 9788498750959.
- Kotler, Philip. 2008.** *Principios de Marketing*. s.l. : Pearson Educación, 2008.

**Lago, Diego. 2008.** Guía de estilode programación. [En línea] 2008. [Citado el: 4 de Mayo de 2014.] <http://did.cexto.com>.

**Larman, Craig. 2003.** *UML y patrones Tomo II*. s.l. : PEARSON, 2003.

—. **2003.** *UML y Patrones. 2da Edición*. s.l. : Prentice Hall, 2003. ISBN: 84-205-3438-2.

**LibrosWeb. 2014.** LibrosWeb. [En línea] 2014. [Citado el: 16 de Febrero de 2014.] [http://librosweb.es/javascript/capitulo\\_1/breve\\_historia.html](http://librosweb.es/javascript/capitulo_1/breve_historia.html).

**Lima, Fernando. 2014.** SlidePlayer. [En línea] 2014. [Citado el: 7 de Febrero de 2014.] <http://slideplayer.es/slide/143422/>.

**Maldonado, Sergio. 2012.** *Análítica web. Medir para triunfar*. s.l. : ESIC, 2012.

**Marco de desarrollo de la Junta de Andalucía. 2014.** Acunetix Web Vulnerability Scanner. [En línea] 2014. [Citado el: 24 de enero de 2014.] <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/656..>

**Marley, Jimi. 2011.** Programacion. [En línea] 2011. [http://www.programacion.com/articulo/%20por\\_que\\_elegir\\_php\\_143](http://www.programacion.com/articulo/%20por_que_elegir_php_143).

**Medina, Lic Mónica Sánchez.** Elaboración de un sitio web. [En línea] [Citado el: 21 de Marzo de 2014.] <http://cb06tic.wikispaces.com/file/view/GUIA%20DE%20ELABORACION%20DE%20UN%20SITIO%20WEB.pdf/440498206/GUIA%20DE%20ELABORACION%20DE%20UN%20SITIO%20WEB.pdf>.

**Miller, Stephan A. 2012.** *Piwik Web Analytics Essentials*. 2012.

**2013.** Mootools. *a compact javascript framework*. [En línea] 2013. [Citado el: 15 de enero de 2014.] <http://mootools.net/>.

**Morgado, Jose Maria Acuña. 2013.** Reeditor.com. *Librerías javascript de gran utilidad para tu desarrollo web*. [En línea] 9 de octubre de 2013. [Citado el: 13 de enero de 2014.] <http://www.reeditor.com/columna/11084/17/informatica/librerias/javascript/gran/utilidad/tu/desarrollo/web>.

**Ojeda, Antonio Navajas. 2012.** *Guía Completa de CSS3*. s.l. : Autoedición, 2012.

**Olga Pons, Nicolas Marin. 2005.** *Introducción a las bases de datos: el modelo relacional*. s.l. : Paraninfo, 2005.

**Oracle. 2013.** NetBeans. [En línea] Oracle, 2013. [Citado el: 6 de Mayo de 2014.] [https://netbeans.org/index\\_es.html](https://netbeans.org/index_es.html).

**Paéz, Pablo G. 2010.** El mayor portal de gerencia. [En línea] 2010. [Citado el: 3 de Abril de 2014.] <http://www.elmayorportaldegerencia.com/Publicaciones/%5BPD%5D%20Publicaciones%20-%20Metodologia%20para%20una%20toma%20de%20decisiones%20efectiva.pdf>.

**Peter Eisentraut, Bernd Helmle. 2008.** *PostgreSQL-Administration*. 2008.

**2009.** Posicionamiento y Marketing en Buscadores. [En línea] 2009. [Citado el: 4 de Febrero de 2014.] [http://pos-y-marketing-en-buscadores-2013.blogspot.com/2013\\_04\\_01\\_archive.html](http://pos-y-marketing-en-buscadores-2013.blogspot.com/2013_04_01_archive.html).

**Potencier, Fabien.** Symphony. *Oficial Site*. [En línea] <http://symfony.com/>.

**Pressman, Roger. 2005.** *Ingeniería del Software. Un enfoque práctico. Sexta Edición.* s.l. : INTERAMERICANA DE MEXICO, 2005. ISBN 9789701054734.

—. **2005.** . *Un enfoque práctico a la Ingeniería de Software.* 2005. .

*Proceso de Desarrollo de Software Metodología DAC.* **Mendez, Ing. Aleli Sanchez.**

**Quality, V&V. 2011.** Pruebas de seguridad. [En línea] 2011. [Citado el: 24 de Mayo de 2014.] [http://www.vvquality.com/w1/index.php?option=com\\_content&view=article&id=79&Itemid=151](http://www.vvquality.com/w1/index.php?option=com_content&view=article&id=79&Itemid=151).

**Rafal Kuc, Marek Rogozinski. 2013.** . *Mastering Elasticsearch server.* s.l. : PACKT PUBLISHING,, 2013. . ISBN-10: 178328143X..

**2007.** RibosoMatic. [En línea] 17 de noviembre de 2007. [Citado el: 16 de enero de 2014.] <http://www.ribosomatic.com/articulos/top-librerias-ajax-dhtml-y-javascript/>.

**Salinas Caro, Patricio.** Tutorial de UML. [En línea] <http://www.dcc.uchile.cl/~psalinas/uml/>.

**Sebastián, J. 2010.** *Modelo Vista Controlador- Definición y Características.* 2010.

**Silva, Mauricio Samy.** *JQuery. A Biblioteca do Programador JavaScript.* Brasil : s.n.

Sistema de Gestión de la Nueva Universidad. *Sistema de Informacion Docente de la Educacion Cubana.* [En línea] <http://sigenu.mes.edu.cu:8080/dmmes/pages/info/aboutUsWelcome.faces>.

**Softbull. 2012.** Softbull. [En línea] TelFo Networks S.L., 2012. [Citado el: 5 de Abril de 2014.] <http://amcharts.softbull.com/>.

**Sommerville, Ian. 2005.** *Ingeniería de Software.* 2005.

**Suárez, Pablo Javier Kabytes. 2010.** Excelentes plugins para crear gráficas con jquery. [En línea] 8 de Octubre de 2010. [Citado el: 30 de enero de 2014.] <http://www.kabytes.com/programacion/4-excelentes-plugins-para-crear-graficas-con-jquery/>.

**The Apache Software Foundation.** Apache. [En línea] <http://httpd.apache.org/>.

**The jQuery Foundation.** JQuery. *write less, do more.* [En línea] [Citado el: 15 de enero de 2014.] <http://jquery.com/>.

*Toma de decisiones empresariales.* **Fincowsky, Enrique Benjamín Franklin. 2011.** 11, 2011, Vol. 6.

**Vea, Andreu. 2013.** *Cómo creamos Internet.* s.l. : Península, 2013.

Visual Paradigm. *Oficial Site*. [En línea] <http://www.visual-paradigm.com/product/vpuml>.

**Zambrano, Renán Alexi Briones. 2013.** *Metodología de Investigación*. 2013.

## Anexos

### Anexo A

Cuestionario.

Preguntas	Respuestas
¿Cómo recolectan los datos las herramientas de Analítica Web en la actualidad?	
¿Cree que es la más idónea? ¿Alguna recomendación?	
¿Le satisface la cantidad de datos recolectados?	
¿Le satisface los reportes mostrados? ¿Alguna recomendación?	
¿Alguna preferencia de colores en cuanto a la interfaz?	
¿Alguna preferencia sobre librerías para graficar?	
¿Alguna preferencia sobre la estructura del sistema en lo referente a la parte visual?	
¿Alguna preferencia sobre herramientas para el desarrollo?	
¿Cantidad de usuarios estimados que utilizaran la aplicación?	

### Anexo B

Descripción de CU.

CU Buscar estadísticas de sitio

<b>Objetivo</b>	Buscar estadísticas por distintos criterios de búsqueda.
<b>Actores</b>	Usuario, Administrador
<b>Resumen</b>	El CU se inicializa cuando el actor desea ver las visitas y el origen de las mismas en un intervalo de tiempo.
<b>Complejidad</b>	Alta.
<b>Prioridad</b>	Crítico.



<b>Precondiciones</b>	<p>El usuario ha sido autenticado.</p> <p>El usuario ha registrado al menos un sitio.</p> <p>Al menos un sitio de los registrados ha sido visitado.</p>
<b>Postcondiciones</b>	Se generaron los reportes "Visitas" y "Origen de las visitas".

**Prototipo**

The screenshot shows a search form with the following elements:

- Sitios 1:** A dropdown menu containing the text "dfgdfdfg".
- Intervalos 2:** Radio buttons for "Todo", "Años", "Meses", "Semanas", "Días", and "Horas".
- Fecha Inicial 3:** A date field showing "2014-05-01" and a "Cambiar" button.
- Fecha Final 4:** A date field showing "2014-05-25" and a "Cambiar" button.
- Buscar 5:** A "Buscar" button.

A calendar for May 2014 is open, with the 25th highlighted. Below the form, the text "Visitas Generales" and "JS chart by amCharts" are visible.

**Flujo de eventos**

**Flujo básico Buscar estadísticas**

	Actor	Sistema
1.	Accede a la portada de la aplicación.	
2.		<p>Muestra un formulario que brinda la posibilidad de establecer los criterio de búsqueda siguientes:</p> <ul style="list-style-type: none"> <li>- Sitio(<b>Prototipo 1</b>).</li> <li>- Intervalo(<b>Prototipo 2</b>).</li> <li>- Fecha inicial(<b>Prototipo 3</b>).</li> <li>- Fecha final(<b>Prototipo 4</b>).</li> </ul>

3.	Establece la combinación deseada.	
4.	Pulsa la opción: "Buscar" ( <b>Prototipo 5</b> ).	
5.		Valida el criterio de búsqueda seleccionado.
6.		Redirecciona a la vista: "Visitas detalladas".
7.		Genera el reporte "Visitas".
8.		Genera el reporte "Origen de las visitas".
9.		Termina CU

**Flujos alternos**

**1 Error de fecha**

	Actor	Sistema
1.		Verifica que la combinación de fechas está incorrecta.
2.		Muestra el mensaje: "Verificar combinación de fechas".

**2 Datos**

	Actor	Sistema
1.		Verifica que en la combinación de fechas no existen visitas registradas.
2.		Redirecciona a la vista: "Visitas detalladas" y muestra el mensaje: "No hay visitas registradas".

<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	
<b>Requisitos no funcionales</b>		
<b>Asuntos pendientes</b>		

**CU Activar cuenta**

<b>Objetivo</b>	Activar la cuenta dentro de la aplicación.
<b>Actores</b>	Anónimo.
<b>Resumen</b>	El CU se inicializa cuando el actor (en lo adelante usuario) decide activar su cuenta

<b>Complejidad</b>	Baja.	
<b>Prioridad</b>	Secundario.	
<b>Precondiciones</b>	El usuario ha solicitado una cuenta. El sistema ha enviado al correo del usuario un link relacionado con la solicitud.	
<b>Postcondiciones</b>	Se activó la cuenta referente al usuario en cuestión.	
<b>Prototipo</b>		
<p>Se ha activado con éxito la cuenta a nombre de: carlos estrada , ahora puede acceder al sitio. Muchas gracias por utilizar nuestros servicios.</p> <hr/> <p>© 2014 - ARWeb <a href="#">Ayuda</a> <a href="#">Contacto</a> <a href="#">Privacidad</a> <a href="#">Sobre Nosotros</a></p>		
<b>Flujo de eventos</b>		
<b>Flujo básico Activar cuenta</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Accede a la aplicación mediante el link enviado por la misma.	
2.		Activa la cuenta.
3.		Muestra la vista: "Cuenta activa".
4.		Termina el CU.
<b>Flujos alternos</b>		
<b>1 Error dato</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Verifica que la variable en el link no corresponde a ningún usuario.

2.	Muestra el mensaje: "No tiene cuenta registrada en la aplicación para activar".	
<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	Activar cuenta en el CU Solicitar cuenta.
<b>Requisitos funcionales</b>	<b>no</b>	
<b>Asuntos pendientes</b>		

CU Gestionar Rol

<b>Objetivo</b>	Gestionar Rol.
<b>Actores</b>	Administrador.
<b>Resumen</b>	El CU se inicializa cuando el actor (en lo adelante usuario) desea gestionar rol.
<b>Complejidad</b>	Alta.
<b>Prioridad</b>	Crítico.
<b>Precondiciones</b>	El usuario ha sido autenticado en el sistema.
<b>Postcondiciones</b>	Se realizaron satisfactoriamente cualquier acción sobre el rol.

Prototipo

Id	Tipo	Acción
1	ROLE_ADMIN	<ul style="list-style-type: none"> <li>• Ver 2</li> <li>• Editar 3</li> </ul>
5	ROLE_USUARIO	<ul style="list-style-type: none"> <li>• Ver</li> <li>• Editar</li> </ul>

Flujo de eventos

<b>Flujo básico Gestionar Rol</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa la opción "Gestionar" del menú principal y escoge la opción "Rol".	
2.		Muestra una lista de los roles existentes en forma de tabla, con las siguientes opciones: <ul style="list-style-type: none"> <li>➤ Nuevo (<b>Ver Sección 1</b>).</li> <li>➤ Ver (<b>Ver Sección 2</b>).</li> <li>➤ Editar (<b>Ver Sección 3</b>).</li> </ul>
3.	Selecciona una opción.	
4.		Muestra la vista referente a la opción que seleccionó.
5.		Termina CU
<b>Flujos alternos</b>		
<b>Atrás</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa la opción "Atrás".	
2.		<b>Paso 7 del Flujo Básico del CU Autenticar Usuario.</b>
<b>Sección 1: "Nuevo" (Prototipo - 1)</b>		
<b>Flujo básico Nuevo</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa la opción "Nuevo"	
2.		Muestra una vista con un formulario con el siguiente campo: <ul style="list-style-type: none"> <li>- Tipo</li> </ul>
3.	Rellena el formulario.	
4.	Pulsa la acción "Crear".	
5.		Valida que los datos sean correctos
6.		Almacena la información.

7.		<b>Paso 2 del flujo básico</b> y muestra el mensaje: “!!Aviso!! Rol añadido correctamente”
8.		Termina Sección 1
<b>Flujos alternos</b>		
<b>Datos Incorrectos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Verifica que los datos son incorrectos.
2.		Selecciona el campo incorrectos y muestra un mensaje.
<b>Cancelar</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa opción “Cancelar”	
2.		<b>Paso 2 del flujo básico</b>
<b>Sección 2: “Ver” (Prototipo - 2)</b>		
<b>Flujo básico Listado de Roles</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa la opción “Ver”.	
2.		Muestra una vista con los datos correspondientes al rol seleccionado. Además permite hacer las acciones:  - Eliminar - Editar
3.		Termina Sección 2
<b>Flujos alternos</b>		
<b>Eliminar</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa la opción “Eliminar”	
2.		Elimina el rol.
		<b>Paso 2 del flujo básico</b> y muestra el mensaje: “!!Aviso!! Rol eliminado”

		con éxito”.
<b>Editar</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa la opción “Editar”	
2.		<b>Ver Sección 3.</b>
<b>Sección 3: “Editar” (Prototipo - 3)</b>		
<b>Flujo básico Editar</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa la opción “Editar”.	
2.		Muestra una vista con el formulario y los datos del rol seleccionado. Además permite:  - Eliminar
3.	Modifica los datos.	
4.	Pulsa la opción “Actualizar”.	
5.		Valida los datos entrados.
6.		Almacena la información, redirecciona a la vista de “Gestionar Nomenclador Rol” y muestra el mensaje “!!Aviso!! Rol modificado correctamente”
7.		Termina Sección 3
<b>Flujos alternos</b>		
<b>4 Datos Incorrectos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Verifica que los datos son incorrectos.
2.		Selecciona los campos incorrectos y muestra un mensaje.
3.		<b>Ir al paso 2 de la Sección 3</b>
<b>Pulsa opción “Eliminar”</b>		
	<b>Actor</b>	<b>Sistema</b>

1.		Elimina el rol.
2.		<b>Paso 2 del flujo básico</b> y muestra el mensaje: “!!Aviso!! Rol eliminado con éxito”.
<b>Atrás</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa la opción “Atrás”	
2.		<b>Paso 2 del flujo básico.</b>
<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	
<b>Requisitos no funcionales</b>		
<b>Asuntos pendientes</b>		

CU Gestionar Usuario

<b>Objetivo</b>	Gestionar Usuario
<b>Actores</b>	Administrador
<b>Resumen</b>	El CU se inicializa cuando el actor (en lo adelante usuario) desea gestionar algún usuario.
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Crítico
<b>Precondiciones</b>	El usuario está autenticado en el sistema.
<b>Postcondiciones</b>	Se realizan satisfactoriamente cualquier acción sobre el o los usuarios
<b>Prototipo</b>	



Id	Nombre	Apellidos	Correo	Usuario Activo	Acciones
21	Carlos	martinez	clmartinez@uci.cu	no	<ul style="list-style-type: none"> <li>• Ver <b>2</b></li> <li>• Editar <b>3</b></li> </ul>
2	rosali	asfdasd	rrtejeda@estudiantes.uci.cu	si	<ul style="list-style-type: none"> <li>• Ver</li> <li>• Editar</li> </ul>
22	veguita	asdsad	mvega@estudiantes.uci.cu	no	<ul style="list-style-type: none"> <li>• Ver</li> <li>• Editar</li> </ul>
5	dasdq	swerwerew	cmestrada@estudiantes.uci.cu	si	<ul style="list-style-type: none"> <li>• Ver</li> <li>• Editar</li> </ul>

**1**

**Flujo de eventos**

**Flujo básico Gestionar Usuario**

	Actor	Sistema
1.	Pulsa la opción "Gestionar" del menú principal y posteriormente la opción "Usuario"	
2.		Muestra una lista de los usuario existentes en forma de tabla, con las siguientes opciones: - Nuevo ( <b>Ver Sección 1</b> ). - Ver ( <b>Ver Sección 2</b> ). - Editar ( <b>Ver Sección 3</b> ).
3.	Selecciona una opción.	
4.		Muestra la vista referente a la opción que seleccionó.
5.		Termina el CU.

**Flujos alternos**

Atrás		
	Actor	Sistema
1.	Pulsa la opción "Atrás"	
2.		<b>Paso 7 del Flujo Básico del CU Autenticar Usuario.</b>
Sección 1: "Nuevo" (Prototipo - 1)		
Flujo básico Nuevo		
	Actor	Sistema
1.	Pulsa la opción "Nuevo"	
2.		Muestra en la vista un formulario con los siguientes campos: <ul style="list-style-type: none"> <li>- Nombre</li> <li>- Apellidos</li> <li>- Correo</li> <li>- Contraseña</li> <li>- Tipo</li> <li>- Usuario activo</li> </ul>
3.	Rellena el formulario.	
4.	Pulsa la acción "Crear".	
5.		Valida que los datos sean correctos
6.		Almacena la información.
7.		<b>Paso 2 flujo básico</b> y muestra el mensaje: "!!Aviso!! Usuario añadido correctamente".
8.		Termina Sección 1
Flujos alternos		
Datos Incorrectos		
	Actor	Sistema
1.		Verifica que los datos son incorrectos.

2.		Selecciona el campo incorrecto y muestra un mensaje.
3.		<b>Ir al paso 2 Sección - 1</b>
<b>Cancelar</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa opción "Atrás"	
2.		<b>Paso 2 del flujo básico del CU "Gestionar Usuario"</b>
<b>Sección 2: "Ver"</b>		
<b>Flujo básico Ver</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa la opción "Ver".	
2.		Muestra una vista con los datos correspondientes al usuario seleccionado. Además permite hacer las acciones siguientes:  - Eliminar - Editar
3.		Termina Sección 2
<b>Flujos alternos</b>		
<b>Eliminar</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa la opción "Eliminar"	
2.		Elimina el usuario, redirecciona a la vista de "Gestionar Usuario" mostrando el mensaje !!Aviso!! Usuario eliminado con éxito"
<b>Editar</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa opción "Editar"	
2.		<b>Ver Sección 3</b>
<b>Sección 3: "Editar" (Prototipo - 3)</b>		
<b>Flujo básico Editar</b>		

	Actor	Sistema
1.	Pulsa la opción "Editar".	
2.		Muestra una vista con el formulario y los datos del rol seleccionado. Además permite: - Eliminar
3.	Modifica los datos.	
4.	Pulsa opción "Actualizar".	
5.		Valida los datos entrados.
6.		Almacena la información.
7.		<b>Paso 2 flujo básico</b> y muestra el mensaje: "!!Aviso!! Usuario modificado correctamente".
8.		Termina Sección 3
<b>Flujos alternos</b>		
Datos incorrectos		
	Actor	Sistema
1.		Verifica que los datos son incorrectos.
2.		Selecciona los campos incorrectos y muestra un mensaje.
		<b>Ir al paso 2 de la Sección 3.</b>
Eliminar		
	Actor	Sistema
1.	Pulsa la opción "Eliminar"	
2.		<b>Paso 2 flujo básico</b> y muestra el mensaje "!!Aviso!! Usuario eliminado con éxito".
<b>Relaciones</b>		<b>CU Incluidos</b>
		<b>CU Extendidos</b>
<b>Requisitos no funcionales</b>		

<b>Asuntos pendientes</b>	
---------------------------	--

CU Ver código de seguimiento

<b>Objetivo</b>	Ver código de seguimiento.
<b>Actores</b>	Administrador, Usuario
<b>Resumen</b>	El CU se inicializa cuando el actor (en lo adelante usuario) desea ver el código de seguimiento.
<b>Complejidad</b>	Baja.
<b>Prioridad</b>	Crítico.
<b>Precondiciones</b>	El usuario ha sido autenticado en el sistema. El usuario ha registrado al menos un sitio.
<b>Postcondiciones</b>	Se mostró el código de seguimiento para el sitio escogido.

**Prototipo**

Id	Nombre del sitio	Url	Sitio activo	Acciones	Código
25	manuel	http://10.53.8.210/sigecom/web/app.php	no	<ul style="list-style-type: none"> <li>• <a href="#">Mostrar</a></li> <li>• <a href="#">Editar</a></li> </ul>	<ul style="list-style-type: none"> <li>• <a href="#">Ver código</a></li> </ul>

**Flujo de eventos**

**Flujo básico Generar Gráficas**

	Actor	Sistema
1.	Pulsa la opción "Ver código" ( <b>Prototipo 1</b> ).	
2.		Redirecciona a la vista "Código de seguimiento".
3.		Muestra el código de seguimiento referente al sitio.
4.		Termina CU.

<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	Ver código de seguimiento en el CU Gestionar sitio.

<b>Requisitos no funcionales</b>	
<b>Asuntos pendientes</b>	

CU Solicitar Cuenta

<b>Objetivo</b>	Crear una cuenta en la aplicación.
<b>Actores</b>	Anónimo.
<b>Resumen</b>	El CU se inicializa cuando el actor (en lo adelante usuario) desea crear una cuenta en la aplicación.
<b>Complejidad</b>	Media.
<b>Prioridad</b>	Crítico.
<b>Precondiciones</b>	El usuario accede a la dirección de la aplicación web.
<b>Postcondiciones</b>	Se crea la cuenta.

**Prototipo**

The screenshot shows a registration form with the following elements:

- Three input fields for "Nombre", "Apellidos", and "Correo" arranged horizontally.
- An input field for "Contraseña" (Password).
- An input field for "Confirmar Contraseña" (Confirm Password).
- A "Registrarme" button with a small icon labeled "1" next to it.
- A "Cancelar" button at the bottom left.

**Flujo de eventos**

**Flujo básico Solicitar cuenta**

	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa la opción "Registrarse" del menú	

	principal.	
2.		Muestra en la vista un formulario con los siguientes campos: - Nombre - Apellidos - Correo - Contraseña
3.	Llena los campos con los datos correspondientes.	
4.	Pulsa la opción "Registrarme" ( <b>Prototipo 1</b> ).	
5.		Valida los datos.
6.		Envía una notificación al correo del usuario con un link a la aplicación.
7.		Redirecciona a la vista "Información de registro".
8.		Termina CU.

**Flujos alternos**

**1 Datos incorrectos**

	Actor	Sistema
1.		Señala los campos erróneos y muestra el mensaje: "Campos incorrectos".

<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	Activar cuenta: Paso 6 del flujo básico.
<b>Requisitos no funcionales</b>		
<b>Asuntos pendientes</b>		

CU Autenticar Usuario

<b>Objetivo</b>	Autenticar usuarios en el sistema.
<b>Actores</b>	Usuario.

<b>Resumen</b>	El CU se inicializa cuando el actor desea acceder al sistema.
<b>Complejidad</b>	Baja.
<b>Prioridad</b>	Crítico.
<b>Precondiciones</b>	El usuario accede a la dirección de la aplicación web.
<b>Postcondiciones</b>	Se accede al sistema según el Rol que posea el usuario.

**Prototipo**

**Flujo de eventos**

**Flujo básico Autenticar Usuario**

<b>Actor</b>	<b>Sistema</b>
--------------	----------------



1.	Escribe la dirección de la aplicación web en el navegador o accede mediante un link.	
2.		Muestra un formulario de autenticación con los siguientes campos: - Correo ( <b>Prototipo - 2</b> ). - Contraseña ( <b>Prototipo - 3</b> ).
3.	Introduce los datos.	
4.	Pulsa la opción "Entrar" ( <b>Prototipo - 1</b> )	
5.		Verifica que los datos introducidos son correctos.
6.		Autentica en la aplicación.
7.		Redirecciona a la vista referente el rol que posea.
8.		Termina CU
<b>Flujos alternos</b>		
<b>1 Campos vacíos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Verifica que los campos están vacíos
2.		Se redirecciona a otra página donde se muestra un formulario con los mismos campos y el mensaje: "!!Error!! Por favor verifique sus credenciales".
<b>2 Usuario o Contraseña incorrectos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Verifica que los datos son incorrectos.
2.		Se redirecciona a otra página donde se muestra un formulario con los mismos campos y el mensaje: "!!Error!! Por favor verifique sus credenciales".
<b>3 Usuario con la cuenta desactivada o no existente</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Verifica que la cuenta tiene el estado desactiva

2.	Se redirecciona a otra página donde se muestra un formulario con los mismos campos y el mensaje: "Su cuenta no está activa o no existe."	
<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	Solicitar cuenta: Paso 1 del Flujo Básico.
<b>Requisitos no funcionales</b>		
<b>Asuntos pendientes</b>		

CU Editar Cuenta

<b>Objetivo</b>	Editar la cuenta personal del usuario.
<b>Actores</b>	Usuario.
<b>Resumen</b>	El CU se inicializa cuando el usuario desea modificar algún dato de su cuenta personal.
<b>Complejidad</b>	Baja.
<b>Prioridad</b>	Secundario.
<b>Precondiciones</b>	El usuario está autenticado.
<b>Postcondiciones</b>	Se modifican los datos de la cuenta.
<b>Prototipo</b>	

**Flujo de eventos**

**Flujo básico Editar Cuenta**

	Actor	Sistema
1.	Pulsa la opción "Mi Cuenta" del menú principal y después escoge la opción "Ver Perfil" .	
2.		Muestra una vista con los datos de la cuenta en un formulario, distribuidos en los siguientes campos: - Nombre - Apellidos - Correo - Contraseña
3.	Modifica los datos deseados.	
4.		
5.	Pulsa la opción "Actualizar" <b>(Prototipo 1)</b> .	
6.		Valida los datos.
7.		Actualiza la información de la cuenta y muestra el mensaje: "!!Aviso!! Los datos de tu perfil han sido modificados satisfactoriamente".

8.		Termina CU.
<b>Flujos alternos</b>		
<b>1 Campos vacíos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Verifica que los campos están vacíos
2.		Señala los campos en blanco y muestra el mensaje: "Rellene este campo".
<b>2 Datos incorrectos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Verifica que los datos son incorrectos.
2.		Señala los campos incorrectos y muestra el mensaje: "Entrada incorrecta".
<b>3 Atrás</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa la opción "Atrás".	
2.		<b>Paso 7 del Flujo Básico del CU Autenticar Usuario.</b>
<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	
<b>Requisitos no funcionales</b>		
<b>Asuntos pendientes</b>		

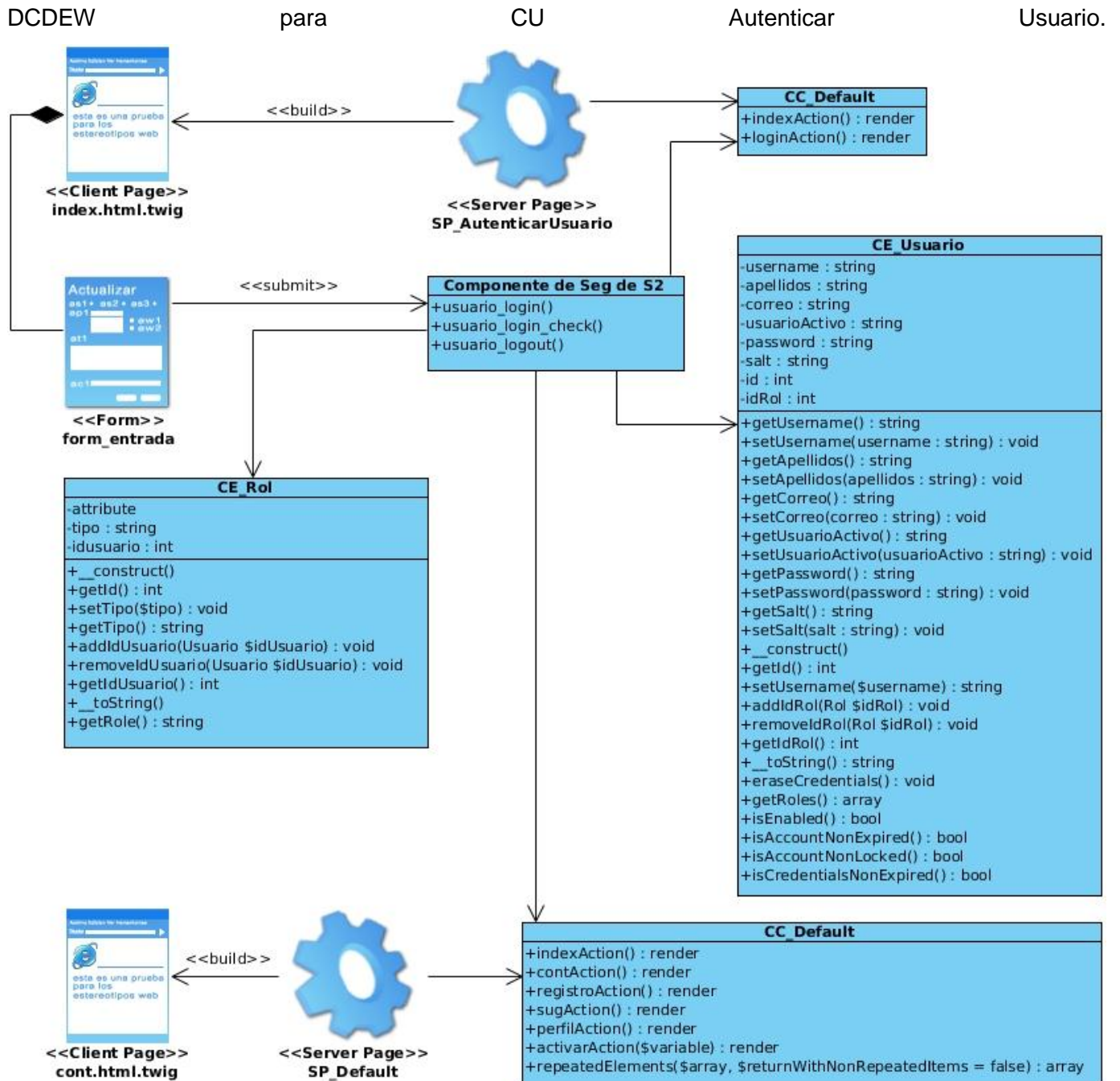
CU Cerrar sesión

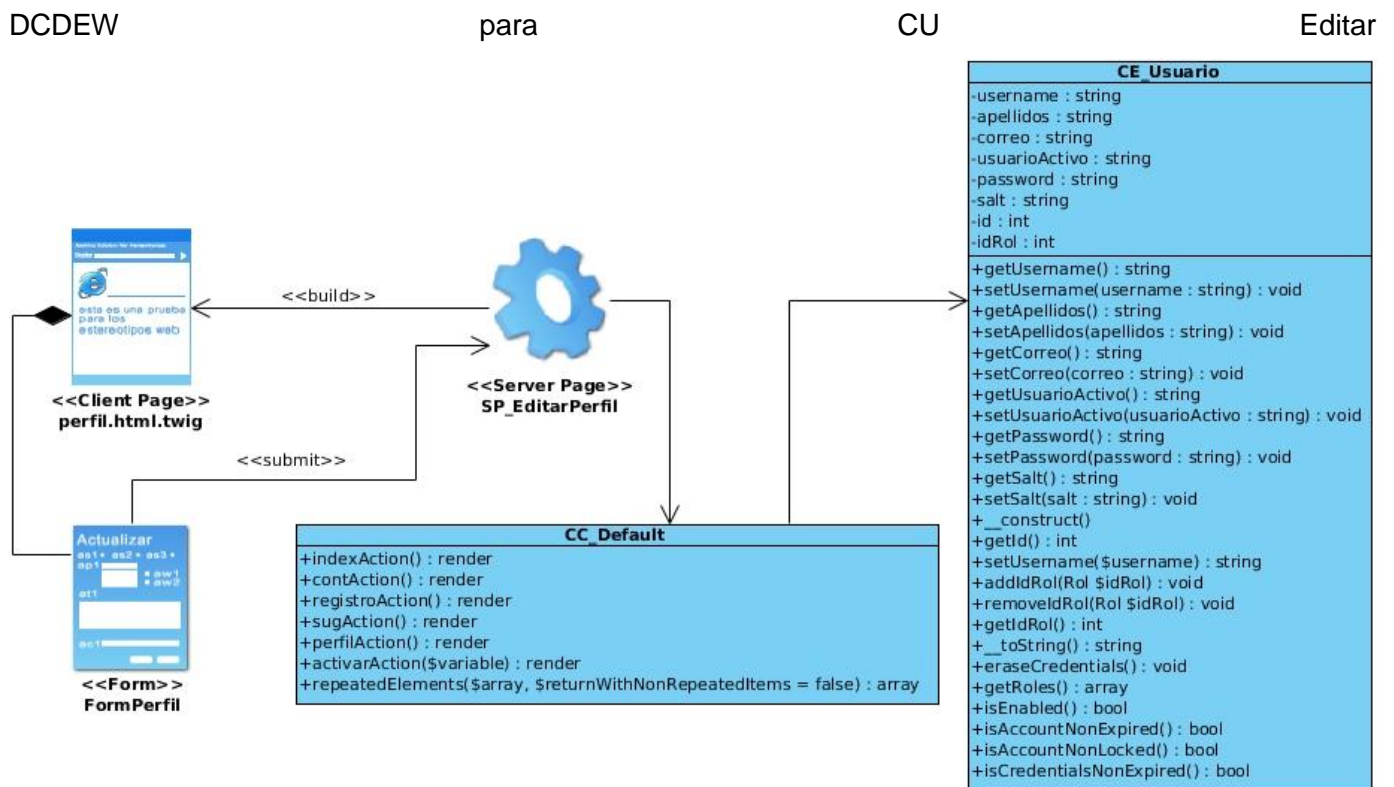
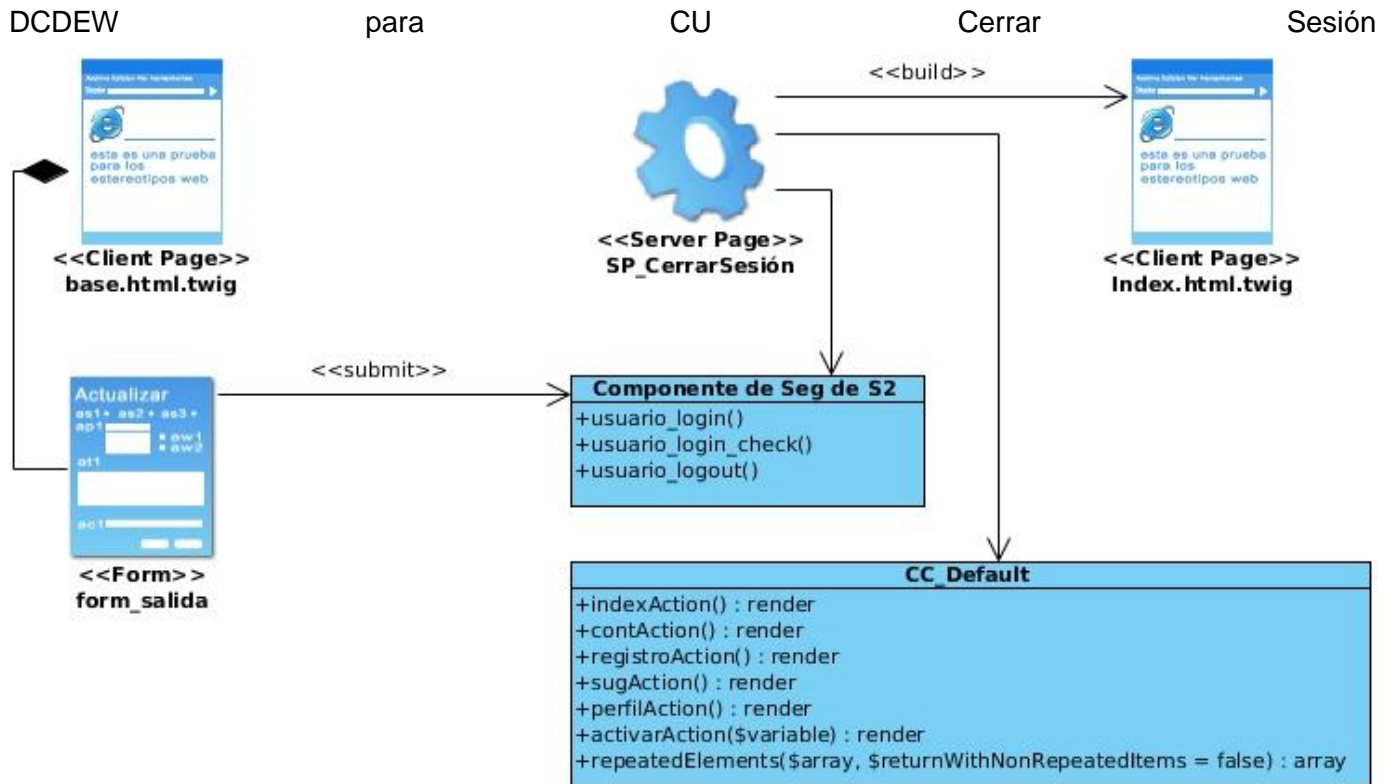
<b>Objetivo</b>	Desloguear al usuario.
<b>Actores</b>	Usuario, Administrador
<b>Resumen</b>	El CU se inicializa cuando el actor (en lo adelante usuario) decide cerrar su sesión.
<b>Complejidad</b>	Baja.

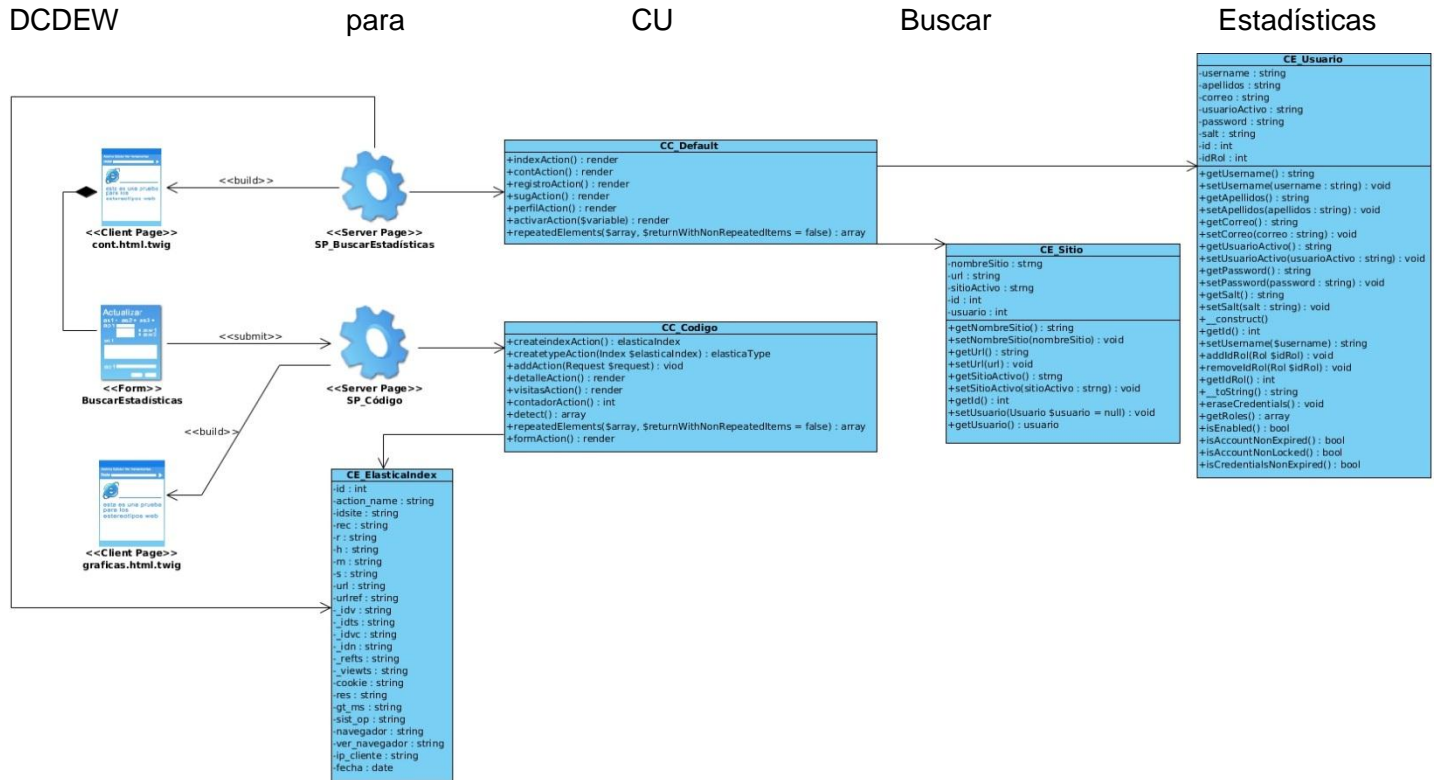
<b>Prioridad</b>	Secundario.	
<b>Precondiciones</b>	El usuario ha sido autenticado.	
<b>Postcondiciones</b>	Se deslogueó al usuario.	
<b>Prototipo</b>		
		
<b>Flujo de eventos</b>		
<b>Flujo básico Activar cuenta</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pulsa la opción "Mi Cuenta" del menú principal y escoge la opción "Salir" <b>(Prototipo 1)</b> .	
2.		Desloguea al usuario.
3.		Redirecciona a la portada.
4.		Termina el CU.
<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	
<b>Requisitos funcionales</b>	<b>no</b>	
<b>Asuntos pendientes</b>		

**Anexo C**

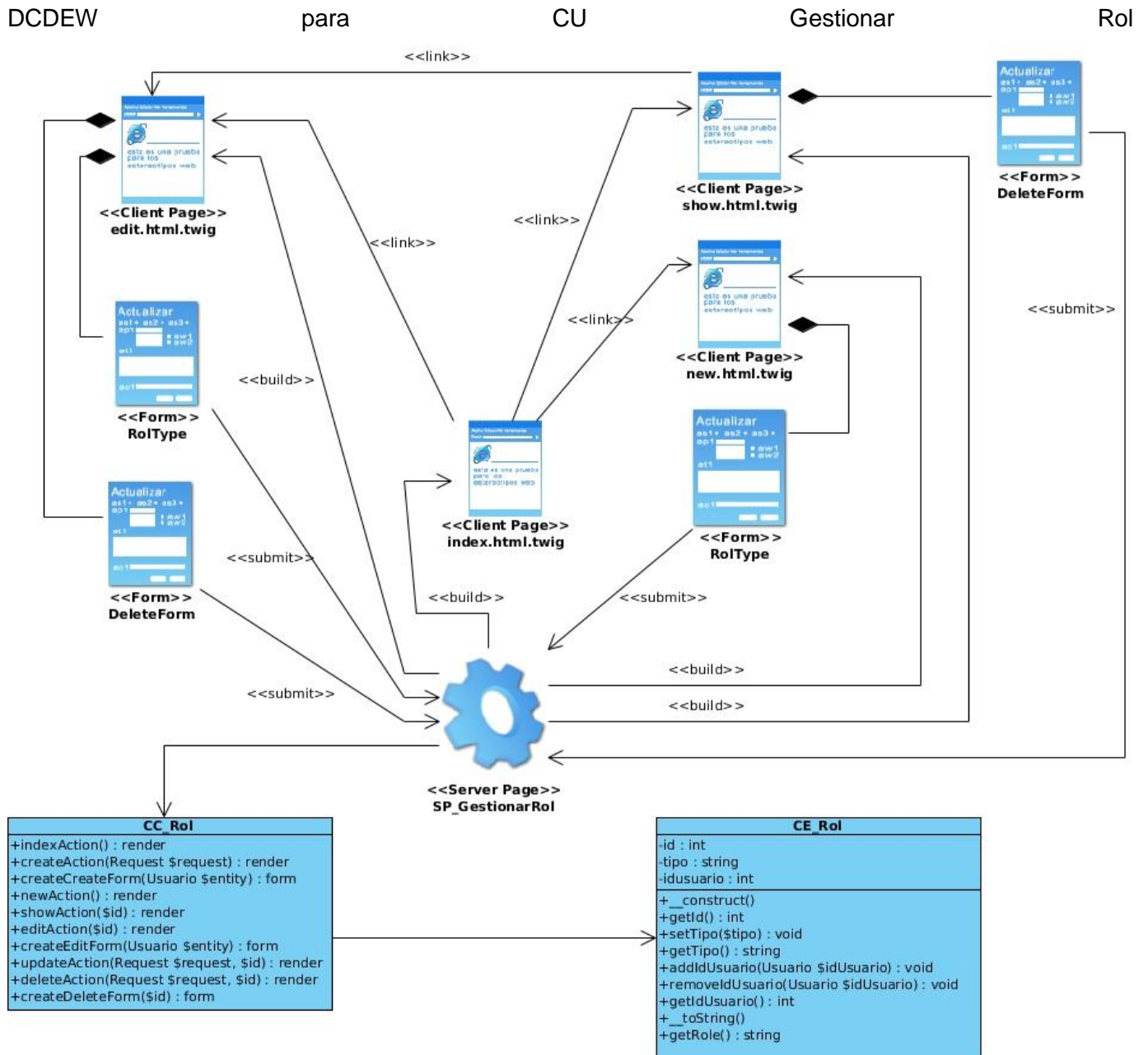
Diagramas de clases del diseño con estereotipos web.











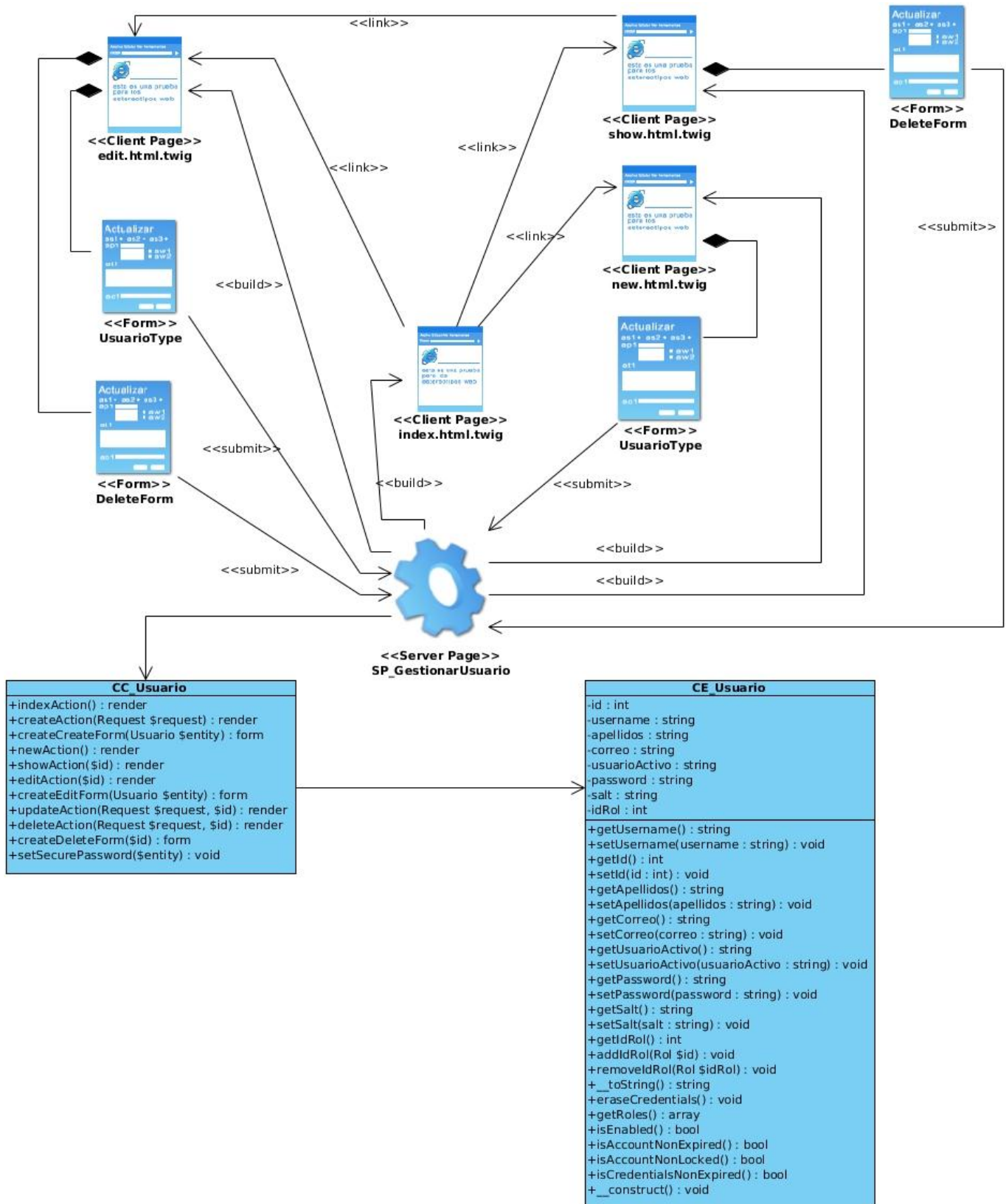
DCDEW

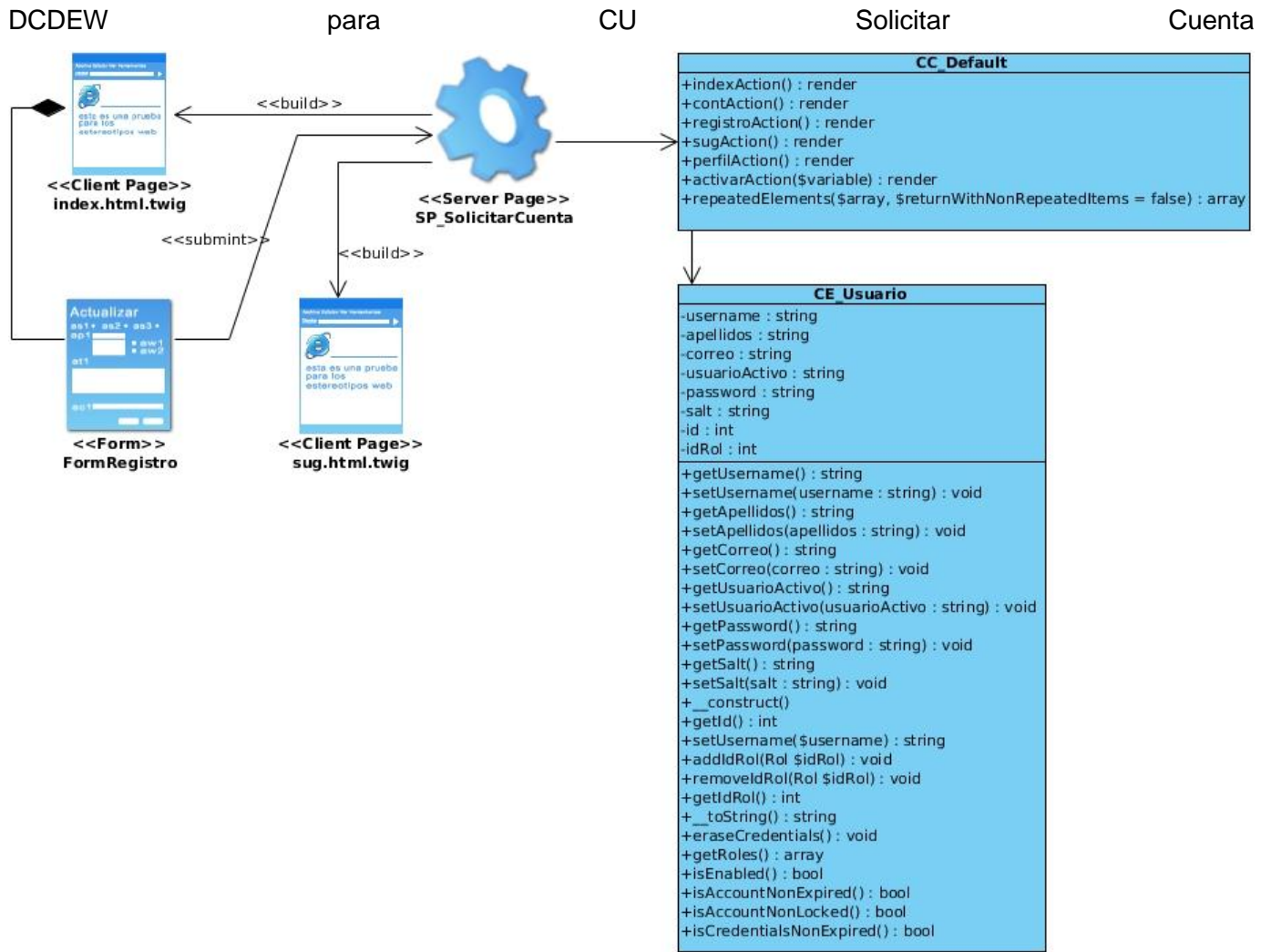
para

CU

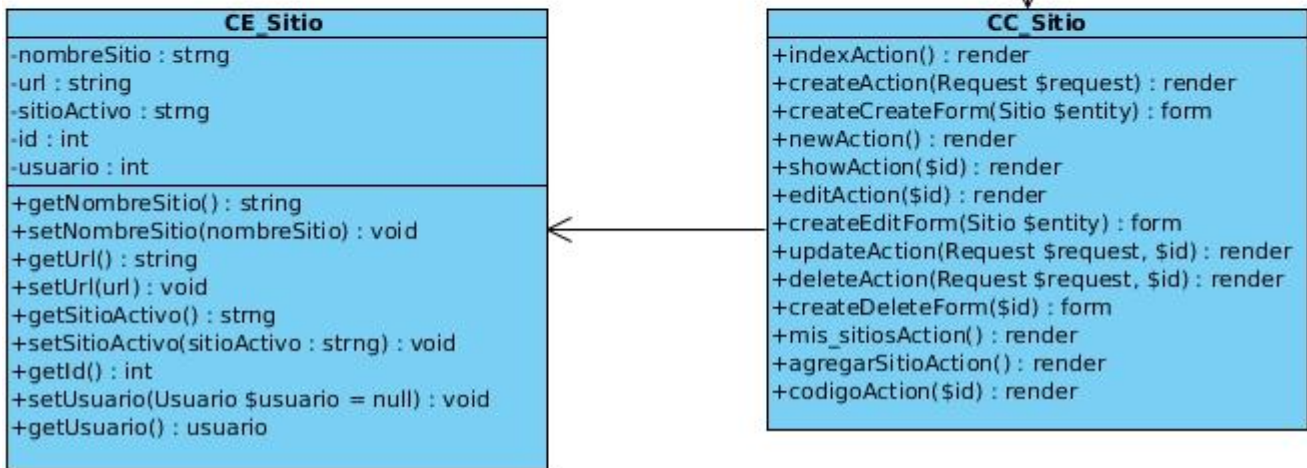
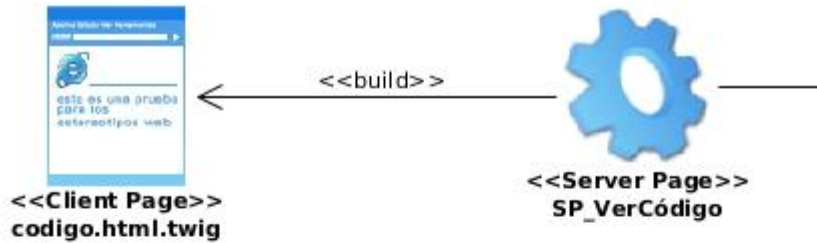
Gestionar

Usuario





DCDEW para CU Ver Código



### Anexo D

Diagramas de Secuencia

Diagrama de Secuencia para CU Activar Cuenta

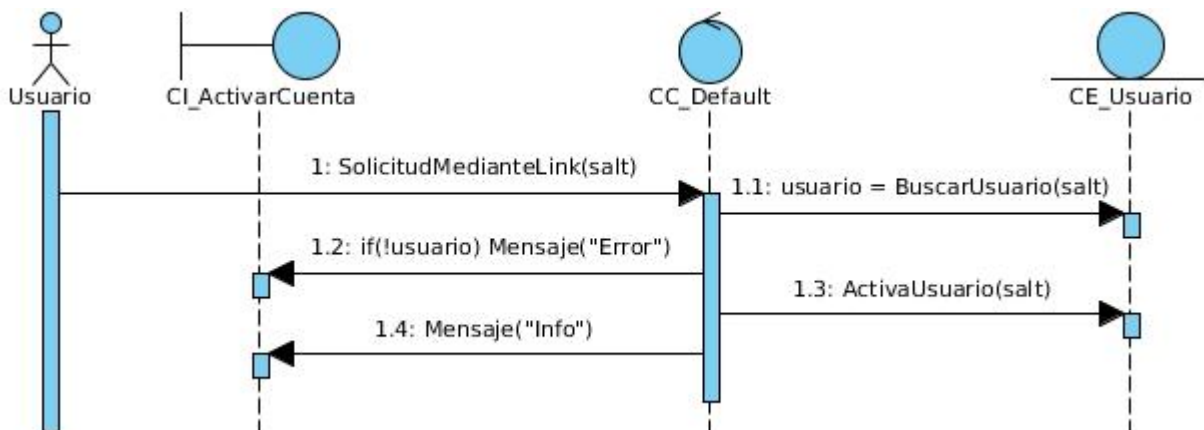


Diagrama de Secuencia para CU Autenticar Usuario

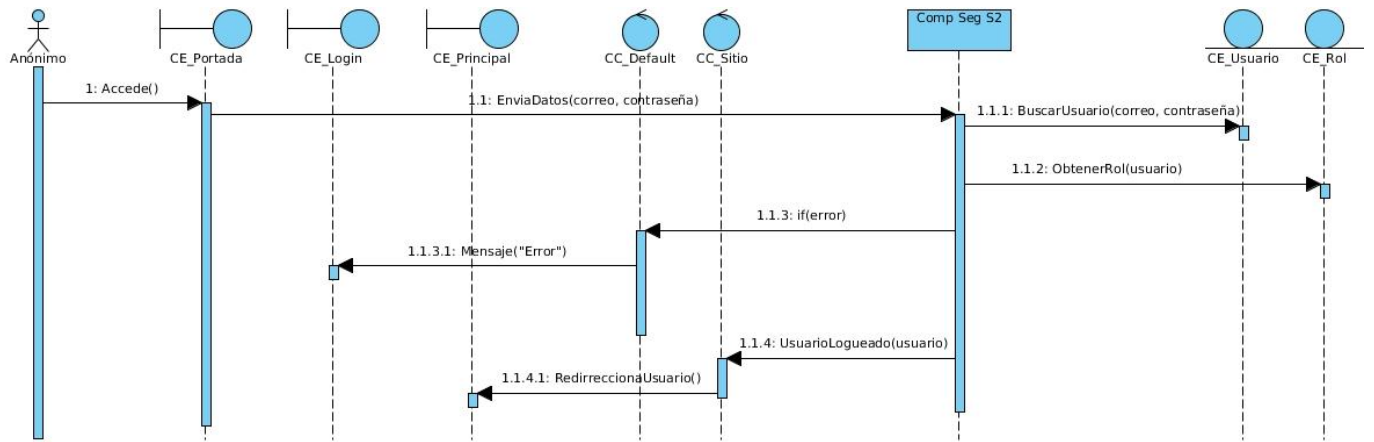
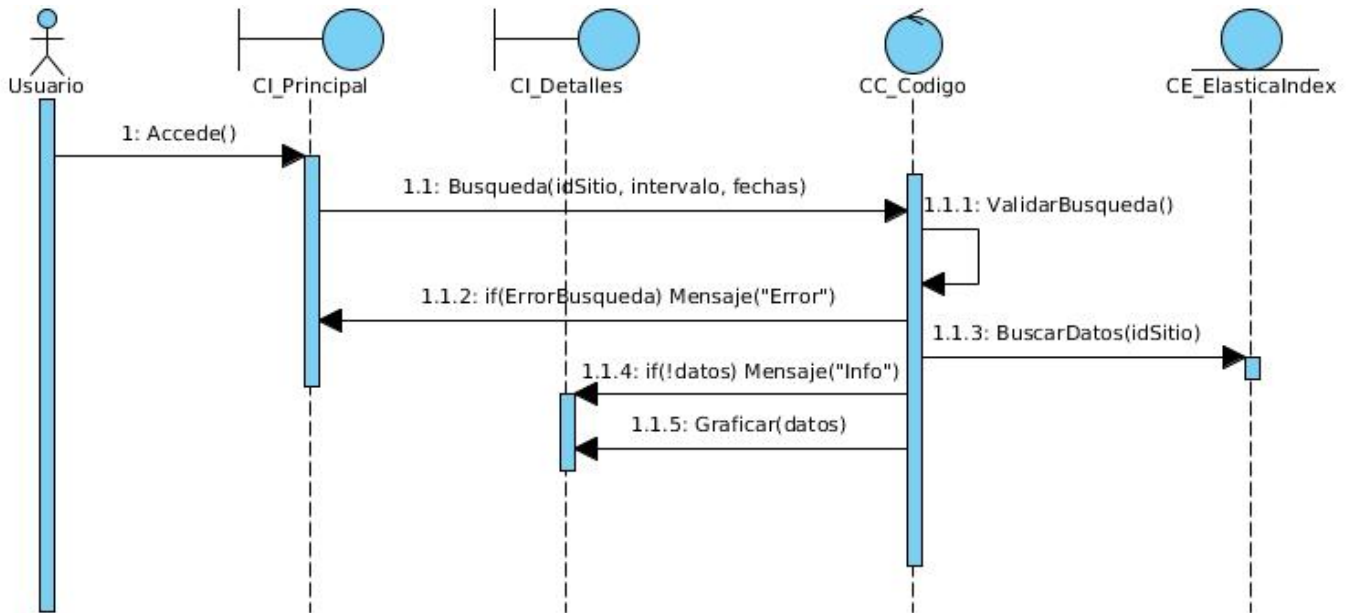


Diagrama de Secuencia para CU Buscar Estadísticas



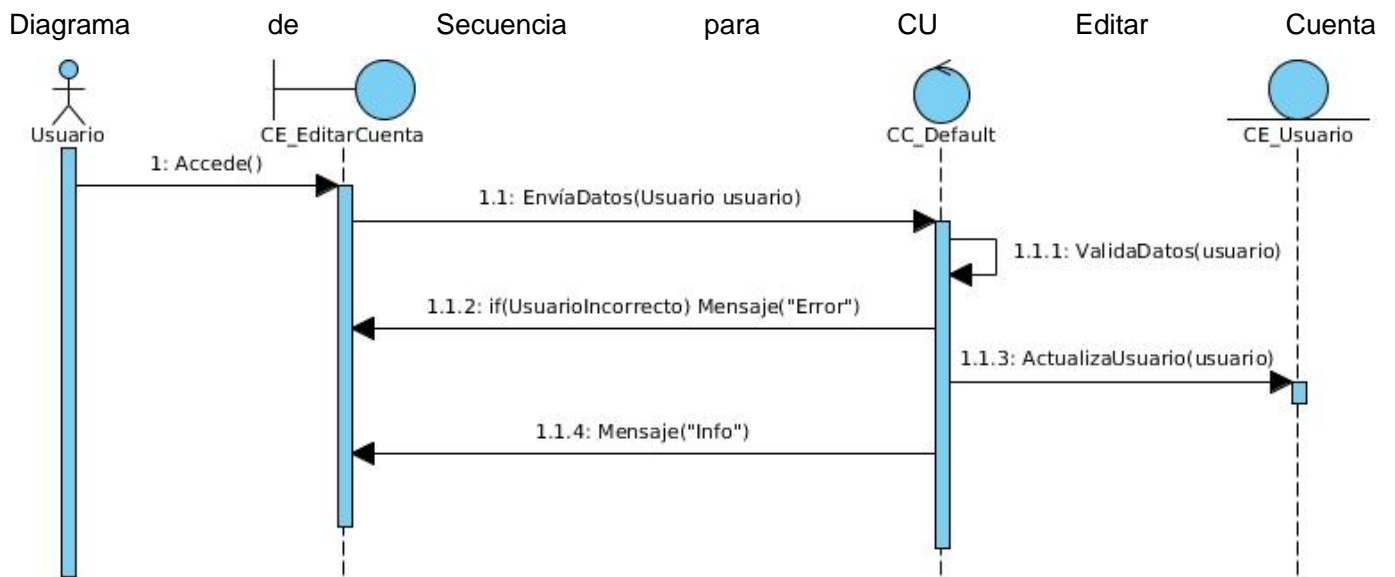
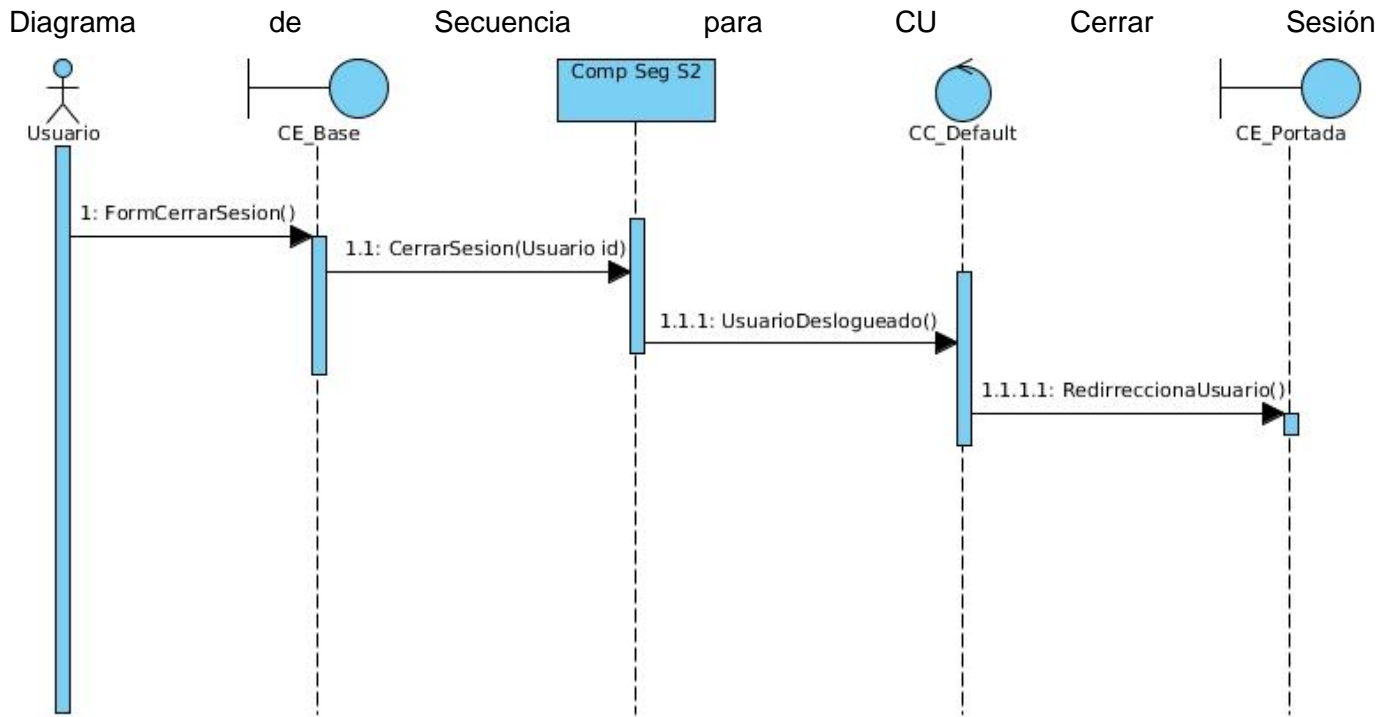


Diagrama de Secuencia para CU Gestionar Rol

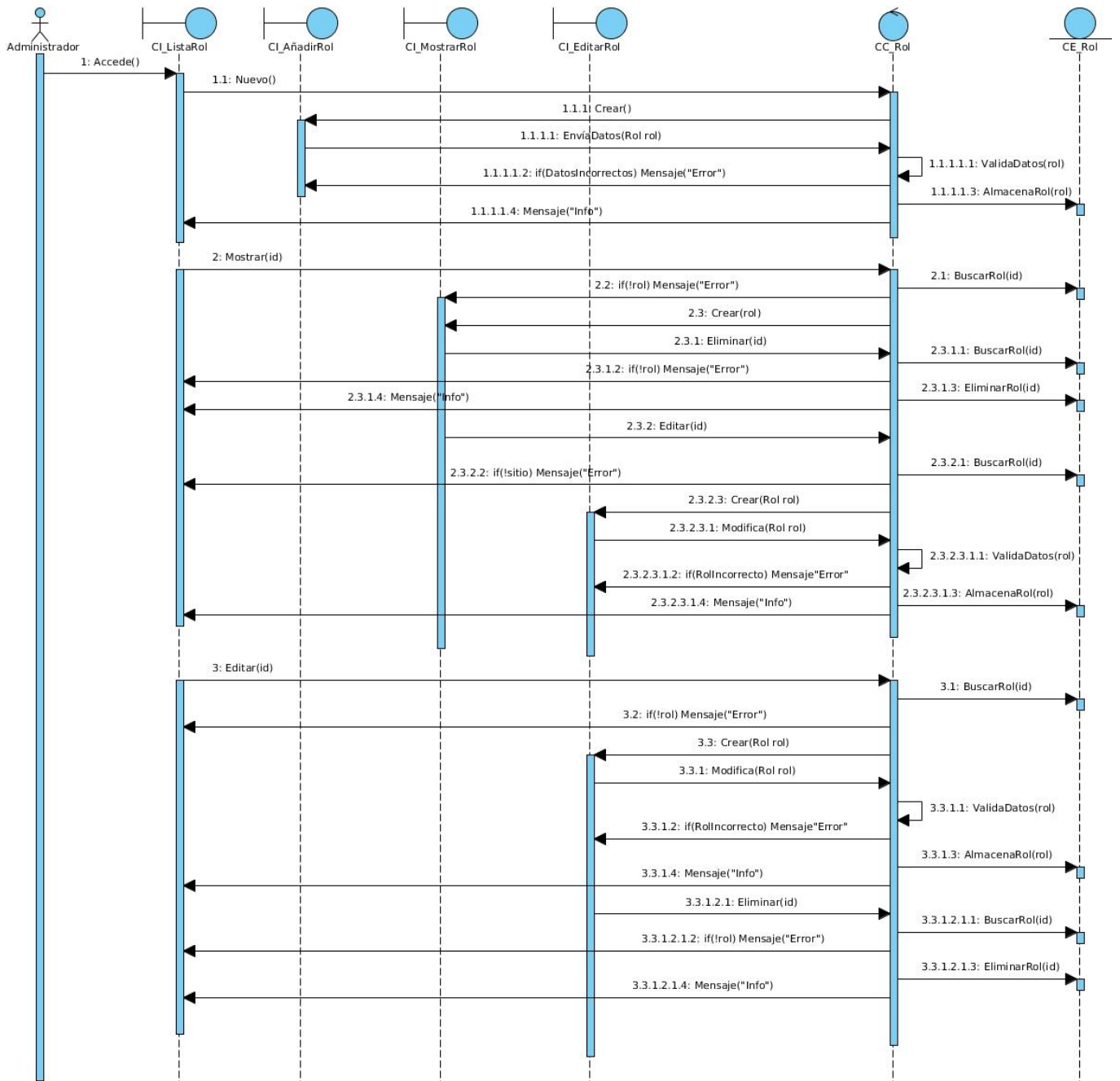
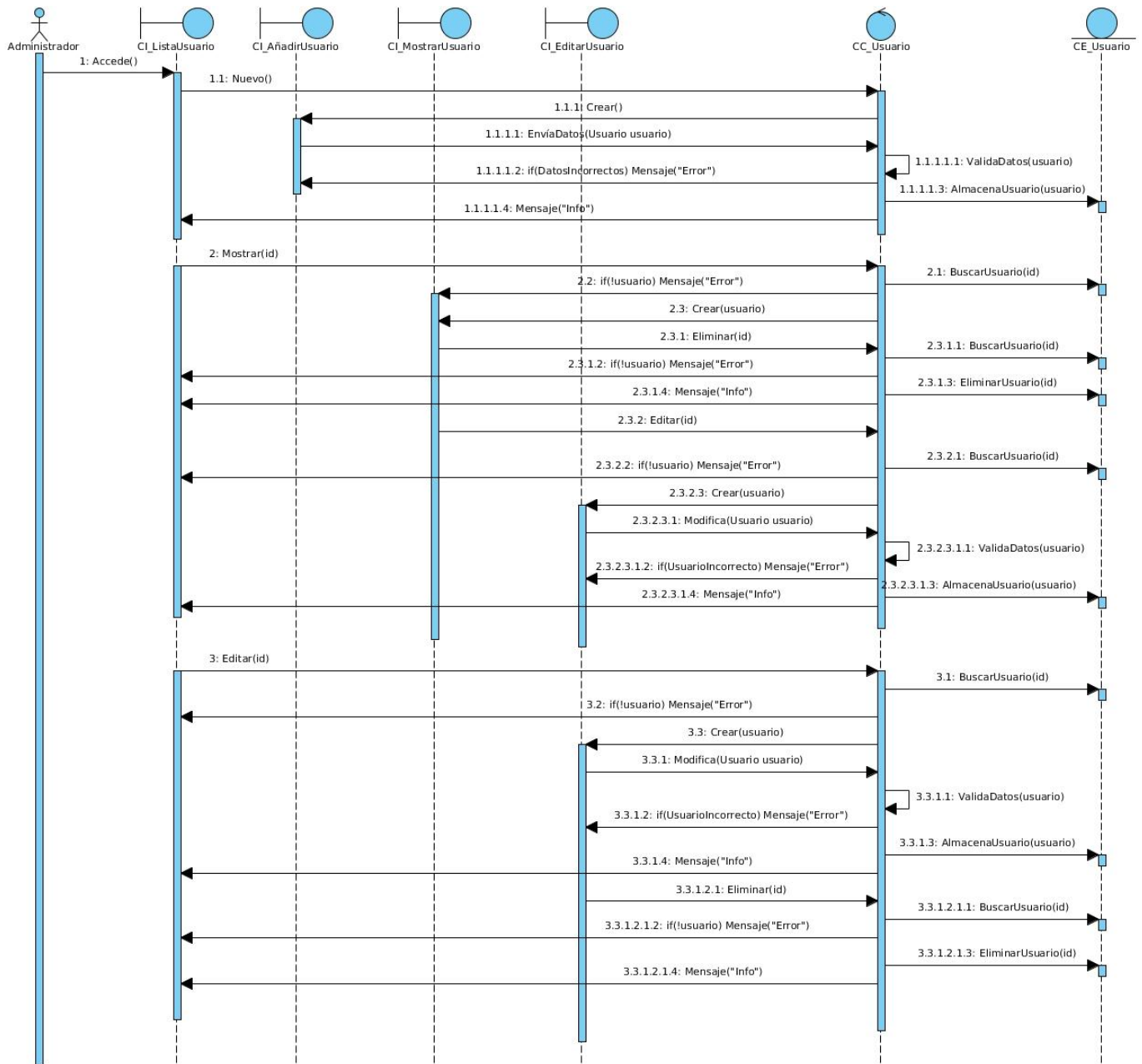


Diagrama de Secuencia para CU Gestionar Usuario





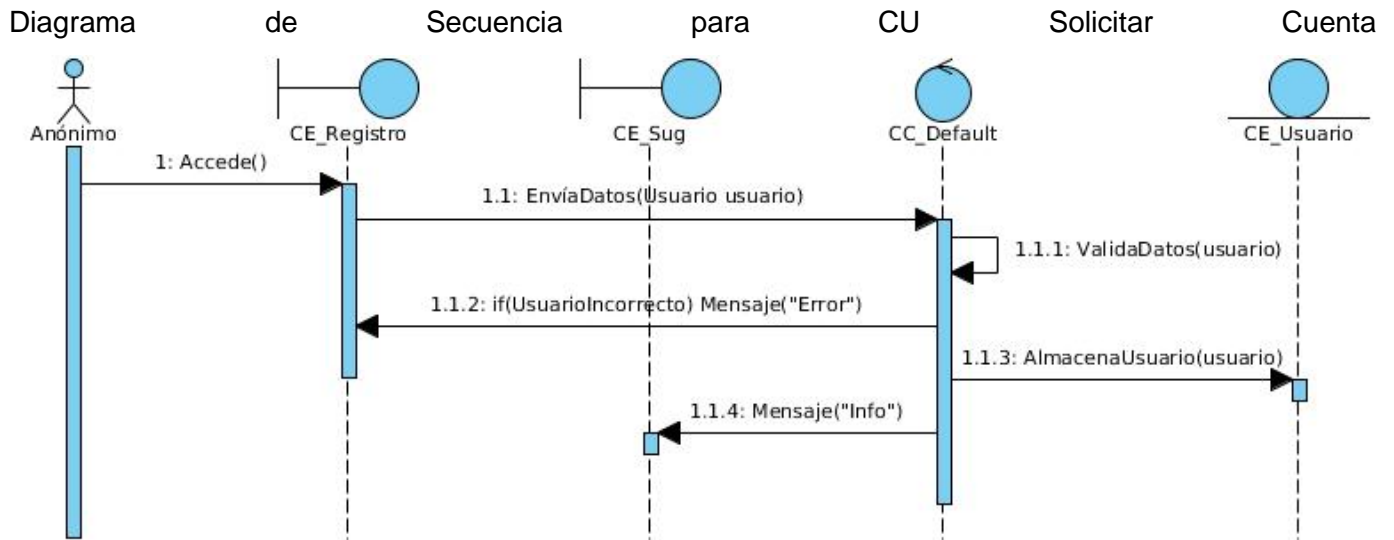
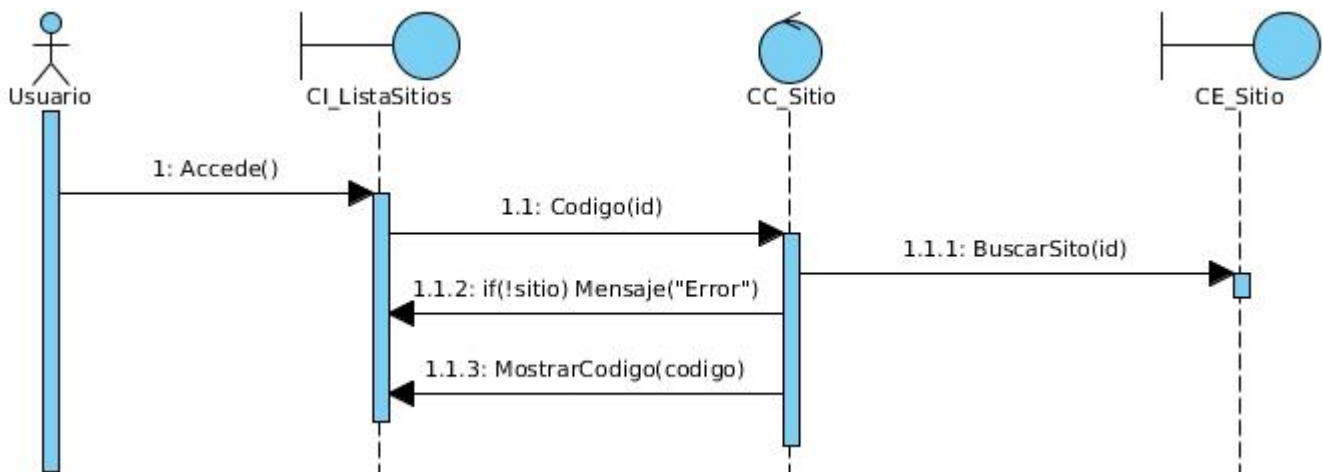


Diagrama de Secuencia para CU Ver Código



**Anexo E**

Diagramas de componentes.

Diagrama de Componente del *Bundle: BackEndBundle*

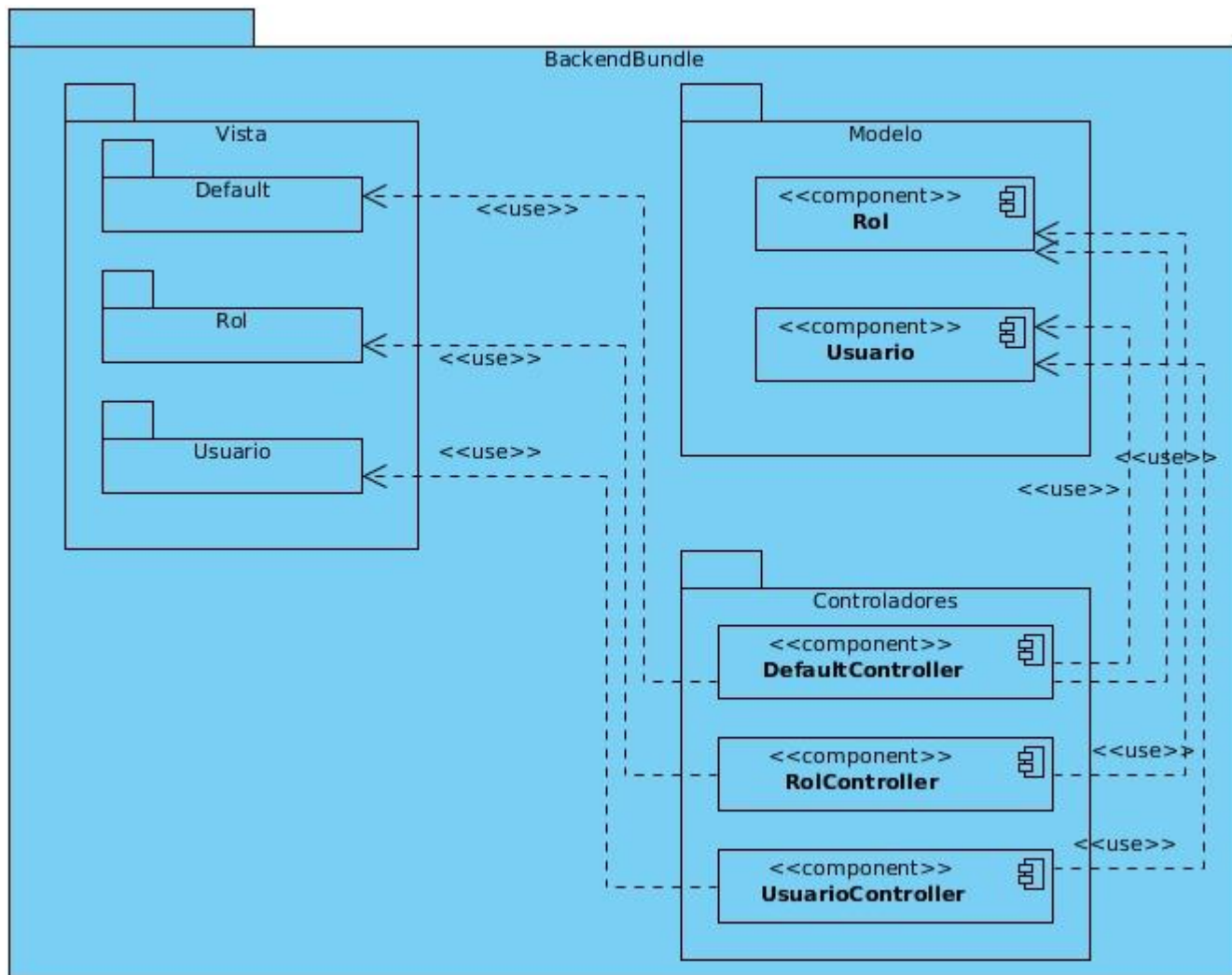


Diagrama de Componente del Bundle: *FrontEndBundle*

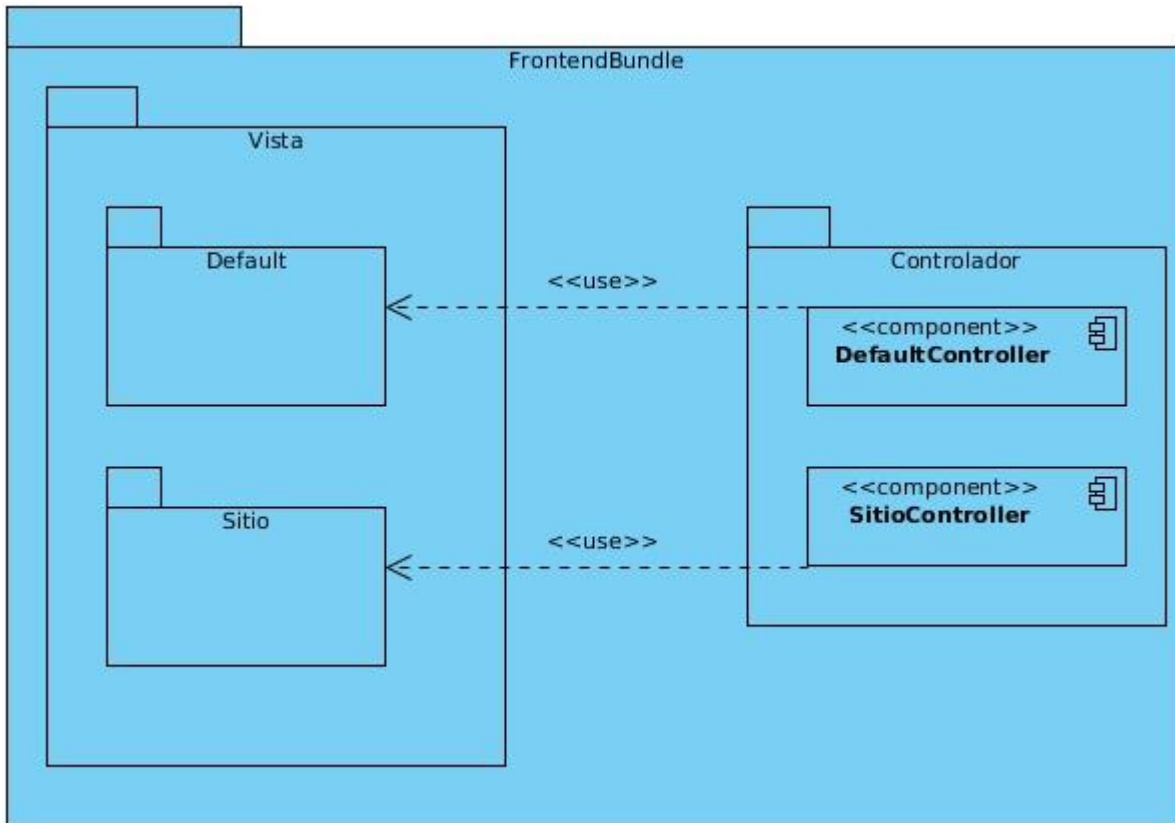


Diagrama de Componente del Bundle: *SitioBundle*

