

**Universidad de las Ciencias Informáticas**

**Facultad 10**



*Módulo de procesamiento de datos de Airesweb*

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

**Autores:** Dunia Virgen Ruz Góngora

Yordan Suarez Santiago

**Tutores:** Ing. Oscar Andrés Casas Machado

MSc. Graciela González Pérez

**CIUDAD DE LA HABANA, Junio del 2010**

**“Año del 51 Aniversario del triunfo de la Revolución”**

**Declaración de autoría:**

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

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

**Dunia Virgen Ruz Góngora**

\_\_\_\_\_

Firma del Autor

**Yordan Suarez Santiago**

\_\_\_\_\_

Firma del Autor

**Ing. Oscar Andrés Casas Machado**

\_\_\_\_\_

Firma del Tutor

**MSc. Graciela González Pérez**

\_\_\_\_\_

Firma del Tutor



*... Lo fundamental es que seamos capaces de hacer cada día algo que perfeccione lo que hicimos el día anterior...*

*Ernesto Che Guevara*

*Agradecemos a nuestros tutores Oscar y Graciela por su ayuda brindada para la elaboración de este trabajo. A la gente del proyecto Aires en especial a Coty y Adriano. A los profesores con los que hemos contado estos años y los cuales han contribuido a nuestra preparación. A mi compañer@ de tesis, por no asfixiarme y dejarme trabajar a mi paso.*

### **Dunia**

*Agradezco especialmente a mi madre por apoyarme en todo y no tronchar mis quimeras. A mi tía Marbelis que es otra madrecita para mí y siempre ha estado al pendiente. A franco mi esposo por apoyarme en todos estos años. A mi tía Yanelis y su familia quien sin su ayuda no hubiese podido terminar este trabajo. A mi padre adoptivo Aleji por saber ocupar tan bien el rol paterno. A mis hermanos Nayrobis y Danay, a mis sobrinas Dailenis y Yesica por ser la alegría de mi vida. Agradezco a todas las personas que conocí durante estos cinco años y que de una forma u otra me ayudaron, en especial a mis amigos de siempre: Yumi, la niña lila, tato, Julio, Ernesto Julio, José Ernesto, Hector y Ailyn por estar ahí siempre que los necesité. A mi grupo 10101 donde comencé a ver la universidad desde otro punto de vista, también a mi grupo 10306 donde conocí personas magnificas. En fin gracias a todas esas personitas!*

### **Yordan**

*Agradezco a mis padres por el apoyo de todos estos años y por estar siempre al pendiente de mí. A Osvaldo que más que un padrastro ha sabido comportarse como un padre al que admirar y respetar. A mi hermana Yanara por ser parte de mi alegría. A las personas que conocí durante este periodo y que se convirtieron en mis colegas con los cuales sé que puedo contar siempre en especial a Nelio y Abel. A la blizzard, por los amigos que me permitió conocer.*

### *Dunia*

*A mi madre por su cariño y la humilde educación que me dio, mima todos mis triunfos te los dedico.*

*A mis niñas Nana, Lali y Yesi, espero ser un ejemplo para ellas de que todo en la vida se puede.*

*A mi abuelito Eddy que aunque la vida no le alcanzó para ver el concreto de mis sacrificios aún me repican sus palabras 'Mija yo sé que lo vas a lograr'.*

*A franco mi esposo por apoyarme en todo este tiempo y no ponerme barreras.*

*A mi hermanito del alma, Nairobi, por superar la pérdida temprana de nuestro padre biológico y convertirse para mí en una figura paterna a tan temprana edad.*

*A mi bebito que la cigüeña ya me lo trae cuando llegue noviembre.*

### *Yordan*

*A mi madre...*

### Resumen

Internet se ha convertido en parte indispensable dentro de las líneas empresariales así como de las organizaciones formando parte de la sociedad actual. La toma de decisiones y su eficiencia influyen directamente en el establecimiento de metas y propósitos generales dentro de las entidades que son representadas en internet por sitios web. Los analizadores de registros web brindan a una determinada entidad la posibilidad de tener conocimiento de las acciones que se realizan en estos sitios así como lograr un acercamiento a los usuarios- clientes y establecer metas para satisfacer las necesidades de los mismos.

El Analizador Inteligente de Registros Web (Airesweb) es un software de este tipo que se desarrolla en la UCI, actualmente cuenta con una base de datos donde se almacena la información generada. Para optimizar el traslado de los datos desde un archivo de registro hacia la base de datos del sistema, este trabajo tiene como propósito la implementación de un módulo de procesamiento de datos para el sistema Airesweb. Con la intención de lograr dicho objetivo se hizo un estudio de los principales software de análisis de registro existentes, se seleccionaron los lenguajes y herramientas de programación a utilizar y se optó por la unión de dos metodologías ágiles Scrum-XP para guiar el ciclo de vida del producto. La selección acertada de los medios de trabajo y la metodología de desarrollo utilizada hicieron posible obtener un módulo eficiente que cumple con los requisitos establecidos y que está funcionalmente activo para ser incluido en futuras versiones de Airesweb.

**Palabras Claves:** Toma de decisiones, sitios web, analizadores de registros web, archivo de registro, Airesweb, módulo de procesamiento, internet.

---

Resumen.....	5
Introducción.....	1
<b>Capítulo 1: Fundamentación teórica.</b> .....	<b>5</b>
1.1    Trabajos similares .....	5
1.2    Ámbito Internacional.....	5
1.3    Ámbito Nacional .....	8
1.4    Módulo .....	8
1.4.1    Características de un módulo.....	8
1.4.2    Módulos de un software .....	9
1.4.3    Ventajas de un módulo.....	10
1.4.4    Clasificación de los módulos .....	11
1.4.5    Criterios de la programación modular.....	12
1.5    Tecnologías a utilizar .....	12
1.5.1    Lenguaje de programación.....	12
1.5.2    Gestores de base de datos .....	18
1.5.3    Lenguaje de modelado .....	21
1.6    Metodología de desarrollo.....	23
1.6.1    Programación Extrema (XP) .....	23
1.6.2    Scrum.....	26
1.6.3    Proceso Unificado de Desarrollo (RUP) .....	27
1.6.4    Proceso Unificado de Desarrollo (RUP) y Programación extrema (XP) .....	29
1.6.5    Scrum-XP (SXP).....	32
1.7    Herramientas .....	32
1.7.1    Herramienta de programación.....	32
1.7.2    Herramienta de modelado.....	33
1.8    Conclusiones .....	34
<b>Capítulo 2. Características del sistema</b> .....	<b>35</b>

---

2.1	Descripción del sistema .....	35
2.2	Solución propuesta.....	35
2.3	Lista de reserva del producto.....	35
2.4	Descripción de las Historias de Usuarios .....	37
2.5	Diagrama de componentes .....	47
2.6	Plan de releases.....	49
2.7	Clases del módulo.....	50
2.8	Diseño de la base de datos.....	51
2.9	Conclusiones .....	51
	<b>Capítulo 3. Validación de la solución propuesta.....</b>	<b>52</b>
3.1	Casos de pruebas .....	52
3.2	Resultados obtenidos.....	60
3.2.1	Tiempo de desarrollo.....	60
3.3	Conclusiones .....	61
	Conclusiones.....	62
	Recomendaciones.....	63
	Anexos.....	68
	Glosario de términos.....	74

### **Introducción**

Internet, conocido inicialmente como ARPANET, se ha convertido en un medio masivo de comunicación mundial. Las tendencias actuales y el auge del uso de Internet por parte de entidades e individuos así como las mejoras de comunicación y facilidades en aspectos como el comercio que esto ofrece a las personas repercuten en el desarrollo de la sociedad actual.

Diariamente en el mundo se realizan miles de búsquedas y transferencias de datos utilizando la red, para ello se emplean los sitios web que forman parte de esta, los mismos están constituidos a su vez por páginas web que se conectan entre ellas mediante hipervínculos y que tienen dentro de sus características principales la combinación de textos con imágenes[1].

En las últimas dos décadas y con el aumento del uso del internet las organizaciones y la mayoría de las grandes y pequeñas empresas cuentan con sitios web que les promocionan y representan las 24 horas del día, dicho comportamiento provoca un incremento en la competitividad y la productividad de los mismos. Ante esta situación la toma de decisiones por parte de los dirigentes o involucrados se hace cada vez más compleja, la necesidad de controlar las acciones realizadas en estos sitios web así como obtener información de su funcionamiento se agudiza influyendo en el desempeño y el mantenimiento dentro de la competencia con respecto a otras líneas empresariales[2].

Debido a esta situación las empresas y organizaciones tienen que enfrentarse a problemas que dificultan el interactuar con los usuarios, muchas de estas dificultades dependen de incorrectos diseños tanto técnicos como gráficos de las páginas web. El análisis de los registros web juega un papel importante para enfrentar dichos conflictos pues brinda a una entidad una serie de datos significativos que les posibilita medir adecuadamente el rendimiento de su presencia en Internet y si su accionar está correctamente ejecutado.

A solicitud del Centro de Información para la Prensa (CIPRE), centro con alcance nacional especializado sobre el periodismo que se dedica al tratamiento de la información generada por fuentes internas y externas a la organización, desde febrero del 2008 en la UCI se desarrolla un software de análisis de registros web que lleva por nombre Airesweb, sistema que realiza un análisis inteligente de los archivos de registros web. El mismo cuenta con una base de datos donde se almacena la información generada por el análisis de los registros web.

A partir de los planteamientos anteriores y luego de analizar la importancia de lograr una optimización en el procesamiento de los datos del sistema Airesweb se plantea el siguiente **problema a resolver**: ¿Cómo optimizar el traslado de los datos desde un archivo de registro hacia la base de datos del sistema Airesweb?

Lo que da lugar a la siguiente **idea a defender**: El desarrollo de un módulo procesador de datos gestionará de forma óptima el traslado de la información desde un archivo de registro hacia la base de datos del sistema Airesweb.

Teniendo en cuenta el problema y la idea a defender antes planteada, se define como **objeto de estudio** el procesamiento de grandes volúmenes de datos y el **campo de acción** está enmarcado específicamente en el procesamiento de grandes volúmenes de datos de archivos de registros.

El **objetivo general** del trabajo es: Desarrollar el módulo de procesamiento de datos para el sistema de análisis Airesweb, que recopile y almacene de forma óptima los datos en la base de datos del sistema.

Dicho objetivo a su vez está desglosado en tres **objetivos específicos** los cuales son:

1. Sistematizar antecedentes y estado actual de las aplicaciones de procesamiento de datos a nivel internacional, Cuba y la UCI.
2. Implementar las funcionalidades del módulo de procesamiento de datos del sistema Airesweb.
3. Probar las funcionalidades en un entorno real.

Para lograr los objetivos propuestos se precisan las siguientes **tareas investigativas**:

1. Sistematización y establecimiento de antecedentes de aplicaciones de procesamiento de datos en Cuba y la UCI.
2. Implementación de las funcionalidades del módulo de procesamiento de datos de Airesweb.
3. Realización de pruebas a las funcionalidades seleccionadas para comprobar su correcto funcionamiento.

Del trabajo se esperan los siguientes **aportes prácticos**:

- 1- Optimización de la eficiencia del procesamiento de los datos del sistema de Airesweb para contribuir a la toma de decisiones sobre la usabilidad de un sitio web.
- 2- La creación de una base de datos que mantendrá de forma persistente en el tiempo los datos del sistema Airesweb.
- 3- Divulgación de la utilización de las herramientas de análisis de registros web y el uso de la analítica web y del software libre en Cuba.

Para dar cumplimiento a las tareas investigativas se usaran los siguientes **métodos**:

El **Analítico - Sintético** se aplicó para entender el sistema Airesweb partiendo del análisis de las características que presenta y para formular conclusiones que apoyen el desarrollo del módulo a través de la síntesis de los conocimientos y resultados obtenidos.

El **Histórico - Lógico** permitió una mayor comprensión del estado actual de los sistema de análisis de registros enfatizando en el sistema Airesweb a partir del análisis de su trayectoria y las etapas principales por las que ha transitado teniendo esto en cuenta a la hora de desarrollar el módulo procesador.

La **Entrevista** ayudó en el establecimiento de un acuerdo entre los involucrados, clientes y equipo de trabajo, siendo esencial a la hora de definir las funcionalidades del sistema, identificando a la vez las funcionalidades particulares de cada cliente y las restricciones que se imponen.

La **Modelación** mediante el lenguaje unificado de modelado (UML) se utilizó para reflejar la estructura, relaciones internas y características de la solución a través de diagramas.

### **Estructura del contenido**

El presente trabajo estructura su contenido en tres capítulos enfocados a su adecuado desarrollo:

**Capítulo1. Fundamentación Teórica:** Este capítulo se centra en el estado del arte del objeto de estudio. En él se analiza la existencia de trabajos similares en el ámbito nacional e internacional y se presenta además una descripción de las herramientas utilizadas para el desarrollo del módulo.

**Capítulo 2. Características del sistema:** Capítulo en el cual se hace una explicación de la teoría y las técnicas utilizadas para la realización del sistema.

**Capítulo 3. Validación de la solución propuesta:** En este capítulo se describen las pruebas realizadas a las funcionalidades del módulo en un entorno real del sistema y los principales resultados obtenidos.

### Capítulo 1: Fundamentación teórica.

Los sistemas analizadores inteligentes de registros web son de gran ayuda principalmente para empresas y organizaciones que han modernizado su accionar ubicándose en Internet y se ven constantemente en la necesidad de tomar decisiones claves que influyen en su desempeño. Los analizadores inteligentes de registros web están compuestos por módulos. Con el objetivo de implementar un módulo para Airesweb que modifique, lea y analice ficheros se abordarán conceptos y herramientas a utilizar en dicha implementación.

#### 1.1 Trabajos similares

Un analizador de registros web es un software que obtiene información a partir de los registros web. Airesweb entra en este tipo de software, y frente a la carencia de un módulo que gestione de forma óptima el traslado de la información desde un archivo de registro hacia la base de datos del sistema, se estudiarán softwares similares en el ámbito nacional e internacional que ayuden a la implementación de dicho módulo.

#### 1.2 Ámbito Internacional

Actualmente en el mundo existen potentes sistemas de análisis de registros web dentro de ellos se encuentran los sistemas de análisis Analog, AWStats, Webalizer, Webtrends y WebLog Expert de los antes mencionados se hace una referencia a continuación:

##### **Analog [3]**

Analog es un software libre que muestra los patrones de uso de un servidor web y dentro de sus características principales se destacan: considerado rápido, configurable, contiene informes en 32 idiomas y funciona en cualquier sistema operativo.

Analog es un programa para medir el uso de un servidor web. Muestra las páginas más populares, los países desde los cuales es visitado el sitio monitorizado y todo tipo de información útil a saber en una página web. Está escrito en C estándar, por lo que debe compilar en cualquier máquina con un compilador de C.

### **AWStats [4]**

El Advanced Web Statistics o AWStats por sus siglas en inglés es un software libre distribuido bajo la licencia GNU General Public License (GPL) y se considera como una herramienta completa que genera estadísticas gráficas avanzadas para sitios web. Este analizador de registro funciona como un CGI o desde línea de comandos y muestra, en pocas páginas, toda la información que los archivos de registros contienen. Es capaz de analizar los archivos de registro de servidores importantes, como el registro log de Apache, WebStar, IIS (formato de registro de W3C) y otras web, proxy, WAP, servidores de correo y algunos servidores FTP.

AWStats dentro de las informaciones que muestra se encuentran:

- Número de visitas, y el número de visitantes únicos.
- Duración de las visitas y últimas visitas.
- Los usuarios autenticados, y la última visita autenticada.
- Lista de servidores, últimas visitas y la lista sin resolver de direcciones IP.
- Tipo de archivo.
- Sistemas operativos utilizados (páginas, hits, KB por cada sistema operativo, OS detectados 35).
- Navegadores utilizados.
- Visitas de robots (319 robots detectados).
- Los motores de búsqueda, palabras clave utilizadas para encontrar el sitio.
- Errores de HTTP.
- Otros informes personalizados basados en la URL.
- Número de veces que el sitio es "añadido a los marcadores de favoritos".

Awstats, cuyo código se encuentra programado en Perl, puede ser empleado en la mayoría de los sistemas operativos.

### **Webtrends [5]**

Webtrends presenta licencia privativa, lo cual provoca que su código no pueda ser reutilizado por terceras personas. Webtrends es la solución líder en el mercado de la analítica web que mide y obtiene información sobre las necesidades de sus clientes, respuestas y perfiles de navegación que son muy útiles para tomar decisiones estratégicas como:

- Lanzamiento de nuevos productos o servicios.
- Optimización de las relaciones con sus clientes.

### **Webalizer [6]**

Webalizer es un software de fácil configuración. Está escrito en C lo que aumenta su rapidez y es portátil. Maneja estándar en formato archivo de registro común (CLF) los registros del servidor, el servidor proxy Squid formato nativo, y formatos de registro extendido W3C.

Distribuido bajo la Licencia Pública GNU, por lo que su código fuente está disponible, así como las distribuciones binarias para algunas de las plataformas más populares.

### **WebLog Expert [7]**

WebLog Expert es un rápido y potente analizador de registro de servidores. Produce reportes fáciles de leer en formato HTML que incluyen información en texto y gráficos.

WebLog Expert puede analizar los registros de Apache y los servidores web Internet Information Server (IIS). Incluso puede leer registros comprimidos a GZ y ZIP por lo que no se tiene que descomprimir manualmente. Su interfaz es intuitiva y presenta un asistente para mejorar la interacción con el sitio.

WebLog Expert dentro de los informes que muestra se encuentran:

- Las estadísticas de las páginas, archivos, imágenes, directorios, las consultas, el tiempo vistas, las páginas de entrada, páginas de salida, los abandonos, las rutas a través del sitio, los tipos de archivos virtuales y los dominios.
- Información sobre los visitantes: los ejércitos, dominios de nivel superior, países, estados, ciudades, organizaciones, los usuarios autenticados, resoluciones de pantalla y profundidad de color.

- Sitios de referencia, las direcciones URL, los motores de búsqueda (incluida la información acerca de las frases de búsqueda y palabras clave).
- Información acerca de los errores: los tipos de error, información detallada del error.

### 1.3 **Ámbito Nacional**

A pesar del desarrollo que ha adquirido Cuba en la informática y los pequeños avances efectuados en cuanto al conocimiento de la analítica web no se tiene conocimientos sobre otra aplicación nacional, exceptuando el proyecto Airesweb, que apoyen a las organizaciones y entidades cubanas en la toma de decisiones. El sistema Airesweb actualmente está en desarrollo y perfeccionamiento y constituye la primera solución de su tipo en el país. Para optimizar el traslado de información desde los registros web hacia la base de datos se hace necesario la implementación de un módulo procesador de datos.

La versión Alfa liberada de Airesweb fue programada utilizando como lenguaje de programación perl, esto hacia muy lento el procesamiento de los datos es por ello que con el fin de implementar y desarrollar eficientemente el módulo de procesamiento de datos de Airesweb para una futura versión 1.0 surge la necesidad de seleccionar herramientas y conceptos previos que aumenten el conocimiento y apoyen dicho fin.

### 1.4 **Módulo [8]**

Un módulo es aquél que está constituido por una o varias instrucciones físicamente contiguas y lógicamente encadenadas, las cuales se pueden referenciar mediante un nombre y pueden ser llamadas desde diferentes puntos de un programa.

Un módulo puede ser:

- Una subrutina (o procedimiento)
- Una función
- Un programa

#### 1.4.1 **Características de un módulo [9]**

- Los módulos deben tener la máxima cohesión y el mínimo acoplamiento. Es decir, deben tener la

máxima independencia entre ellos.

- La salida del módulo debe ser función de la entrada, pero no de ningún estado interno. En la creación de los módulos deben cumplirse tres aspectos básicos: descripción, rendimiento y diseño.
- En la descripción se definen las funciones y objetivos del programa. Para obtener el máximo rendimiento se ha de comprobar que el programa realice el proceso aprovechando al máximo todos los recursos de los que dispone. En cuanto al diseño, se debe comprobar la estructura que sigue el módulo, así como la estructura de los datos y la forma de comunicaciones entre los diversos y diferentes módulos.
- Conforme se extiende el tamaño de los algoritmos, se hace más difícil su revisión, actualización y/o corrección.
- Una política común para solventar este problema consiste en la modularización. Esto significa que el algoritmo se fragmenta en partes llamadas módulos. En realidad, es un método de diseño que tiende a dividir el problema, de forma lógica, en partes perfectamente diferenciadas que pueden ser analizadas, programadas y puestas a punto independiente.
- Realmente la programación modular es un intento para diseñar programas, de forma tal que cualquier función lógica pueda ser intercambiada sin afectar a otras partes del programa.

### 1.4.2 Módulos de un software [10]

Un módulo de software encapsula las funciones relacionadas en un programa conjunto. Se puede encapsular las funciones de:

- Comunicarse con la base de datos
- Mantener un búfer
- Control de un socket de red
- Controlar el acceso a un objeto o conjunto de objetos

Al escribir un módulo de un software es importante atender a lo que mantiene relacionada a las funciones dentro del módulo.

Los módulos suelen tener múltiples funciones y sus propios datos privados. Sólo las funciones del módulo pueden modificar los datos privados del mismo. El módulo puede ser construido como un objeto de tal manera que varios objetos puedan estar simultáneamente usando el mismo módulo.

Los módulos de un software deben utilizar el principio orientado a objeto, el encapsulamiento. Esto significa que sólo las variables que forman parte de las interfaces de módulos pueden ser vistos por fuera de los módulos o funciones.

### **Módulo Interfaces**

Los módulos están divididos en una aplicación privada y una interfaz pública. La implementación se oculta de los módulos y funciones externos. Para los módulos deben crearse interfaces. Estas interfaces son utilizadas por otros módulos o funciones para comunicarse con dicho módulo.

### **Encabezados**

El encabezado define las funciones en el módulo. Las cabeceras se encuentran normalmente en los archivos de cabecera separada. Las mejores prácticas para los encabezados y de los módulos:

- Crear un encabezado para cada código fuente del un módulo en específico.
- Definir las funciones y datos en los módulos que se agrupan en secciones públicas o privadas. Las funciones públicas y los datos se utilizará como interfaz para el módulo. La interfaz es utilizada por otros módulos.
- No permita que las jerarquías de anidado o recursivo incluya archivos de cabecera.
- Crear un archivo de encabezado común que contenga todas las definiciones y las macros que se utilizan normalmente por todos los módulos.
- Trate de mantener los módulos dentro de 400 líneas de código.

Usando estas reglas, un cambio en la interfaz de un módulo no debe requerir cambiar el programa completo.

### **1.4.3 Ventajas de un módulo [11]**

Dividir la programación en módulos trae muchas ventajas para los programadores:

- Reutilizando código: La posibilidad de reutilizar un módulo es una de las utilidades más destacadas.
- Mayor comprensión: Una división inteligente produce una mayor comprensión de las fuentes del programa, ello conlleva también una gran ventaja, pues permite que otros programadores entiendan mejor el código además de facilitarle al programador la depuración del mismo.

- **Compartir proyectos:** A la hora de realizar programas grandes es realmente difícil que una persona sola pueda hacer todo el trabajo. Para solucionar esto, hay que considerar la posibilidad de planificar una programación modular, de manera que varios programadores puedan compartir un proyecto programando distintos módulos cada uno, que después serán unidos.
- **Rápida recompilación:** Antes de disponer del fichero ejecutable se dispone de un fichero objeto. Pues para enlazar los módulos estos deben estar con sus respectivos ficheros objetos. Si se guarda los ficheros objetos de cada módulo, y se tiene que cambiar código en alguno de los módulos, solo tendrá que recompilar dicho módulo, lo que ahorra considerablemente tiempo al programador.

### 1.4.4 Clasificación de los módulos [12]

Los módulos pueden ser clasificados en:

- Funciones
- Procedimientos

Estas funciones y procedimientos se escriben solamente una vez, pero pueden ser referenciados (invocados) en diferentes puntos de un programa, tantas veces como se requiera de modo que se puede evitar la duplicación innecesaria del código.

*Funciones:*

- Las funciones son bloques de instrucciones que tienen por objeto el alcanzar un resultado que sustituirá a la función en el punto de invocación (las funciones devuelven un resultado).
- Cada función se evoca utilizando su nombre en una expresión con los argumentos actuales o reales encerrados entre paréntesis.
- Para hacer una referencia a una función se invoca mediante un nombre y en caso de existir, una lista de parámetros actuales necesarios (argumentos). Los argumentos deben coincidir en cantidad, tipo y orden con los de la función que fue definida. La función devuelve un valor único.
- Las funciones a que se hace referencia, se conocen como funciones de usuario puesto que son definidas por él mismo y permiten su uso en forma idéntica a las funciones estándares. Para coordinar e iniciar el procesamiento, se utiliza un módulo principal que es colocado al final del algoritmo.

#### **1.4.5 Criterios de la programación modular [13]**

Con independencia de las técnicas, los requisitos que debe cumplir la programación modular son:

**Establecimiento de un organigrama modular:** Se realiza mediante bloques, en el que cada bloque corresponde a un módulo y muestra gráficamente la comunicación entre el módulo principal y los secundarios.

**Descripción del módulo principal:** Debe ser claro y conciso, reflejando los puntos fundamentales del programa.

**Descripción de los módulos básicos o secundarios:** Los módulos básicos deben resolver partes bien definidas del problema. Solo pueden tener un punto de entrada y un punto de salida. Si un módulo es complejo de resolver, conviene se subdivida en submódulos. Ningún módulo puede ser llamado desde distintos puntos del módulo principal.

**Normas de la programación:** Dependerán del análisis de cada problema y de las normas generales o particulares que haya recibido el programador.

### **1.5 Tecnologías a utilizar**

De acuerdo a la Real Academia Española, la tecnología es el conjunto de teorías y técnicas que permiten el aprovechamiento práctico del conocimiento científico mientras que el término de tecnología informática es considerado aquel que permite el procesamiento de información por medios artificiales y que incluye todo lo relacionado con las computadoras[14].

Un punto importante en el desarrollo de cualquier software es la selección correcta de las tecnologías a utilizar, pues con ello se estaría asegurando a largo plazo una mejor producción, una disminución del tiempo de trabajo y un producto final con calidad. Para dicha selección es importante tener en cuenta puntos como la disponibilidad de las herramientas, sus licencias y grado de dificultad a la hora del equipo de trabajo efectuar la interacción con estas.

#### **1.5.1 Lenguaje de programación**

Los lenguajes de programación fueron creados por el hombre con el fin de modelar y hacer en las computadoras programas que simulen la realidad en la que se desarrolla la sociedad. Estos lenguajes han

sido clasificados en bajo y alto nivel. El primer grupo consta de instrucciones simples y son muy entendibles al lenguaje de las máquinas mientras que el segundo grupo tiene un alto grado de abstracción y simula el desarrollo de los humanos.

Actualmente existen muchos lenguajes de programación de alto nivel dentro de los más conocidos se encuentran:

- Java
- C#
- C/C++

### 1.5.1.1 Breve sinopsis de Java [15]

El lenguaje Java fue creado por Sun Microsystems Inc. en un proceso por etapas que arranca en 1990, año en el que Sun creó un grupo de trabajo, liderado por James Gosling, para desarrollar un sistema para controlar electrodomésticos e incluso PDAs o Asistentes Personales (pequeños ordenadores) que, además, permitiera la conexión a redes de ordenadores. Se pretendía crear un hardware polivalente, con un Sistema Operativo eficiente (SunOS) y un lenguaje de desarrollo denominado Oak (roble), el precursor de Java. El proyecto finalizó en 1992 y resultó un completo fracaso debido al excesivo coste del producto, con relación a alternativas similares, tras lo cual el grupo se disolvió.

Por entonces aparece Mosaic y la World Wide Web. Después de la disolución del grupo de trabajo, únicamente quedaba del proyecto el lenguaje Oak. Gracias a una acertada decisión de distribuir libremente el lenguaje por la Red de Redes y al auge y la facilidad de acceso a Internet, propiciado por la WWW, el lenguaje se popularizó y se consiguió que una gran cantidad de programadores lo depurasen y terminasen de perfilar la forma y usos del mismo. A partir de este momento, el lenguaje se difunde a una velocidad vertiginosa, añadiéndosele numerosas clases y funcionalidad para TCP/IP. El nombre del lenguaje tuvo que ser cambiado ya que existía otro llamado Oak.

El nombre “Java” surgió en una de las sesiones de “brainstorming” celebradas por el equipo de desarrollo del lenguaje. Buscaban un nombre que evocara la esencia de la tecnología (viveza, animación, rapidez, interactividad...). Java fue elegido de entre muchísimas propuestas. De esta forma, Sun lanzó las primeras

versiones de Java a principios de 1995.

### 1.5.1.2 Breve sinopsis de C# [16]

C# (leído en inglés "C Sharp" y en español "C Almohadilla") es el nuevo lenguaje de propósito general diseñado por Microsoft para su plataforma .NET.

Aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que programar usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes ya que C# carece de elementos heredados innecesarios en .NET. Por esta razón, se suele decir que C# es el lenguaje nativo de .NET.

### Características del lenguaje C#

Las características más relevantes de este lenguaje:

- **Es autocontenido.** Un programa en C# no necesita de ficheros adicionales al propio código fuente, como los ficheros de cabecera (.h) de C++, lo que simplifica la arquitectura de los proyectos software desarrollados con C++.
- **Es homogéneo.** El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina en la que se compile (no ocurre lo que en C++), lo que facilita la portabilidad del código.
- **Es actual.** C# incorpora en el propio lenguaje elementos que se han demostrado ser muy útiles para el desarrollo de aplicaciones como el tipo básico decimal que representa valores decimales con 128 bits.
- **Está orientado a objetos.** C# soporta todas las características propias del paradigma de la programación orientada a objetos: *encapsulación*, *herencia* y *polimorfismo*.
- **Delega la gestión de memoria.** Como todo lenguaje de .NET, la gestión de la memoria se realiza automáticamente ya que tiene a su disposición el recolector de basura del CLR.
- **Proporciona seguridad con los tipos de datos.** C# no admiten ni funciones ni variables globales sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que

reduce problemas por conflictos de nombres y facilita la legibilidad del código.

- **Proporciona instrucciones seguras.** En C# se han impuesto una serie de restricciones para usar las instrucciones de control más comunes.
- **Facilita el mantenimiento (es "versionable").** C# incluye una política de versionado que permite crear nuevas versiones de tipos sin temor a que la introducción de nuevos miembros provoquen errores difíciles de detectar en tipos hijos previamente desarrollados y ya extendidos con miembros de igual nombre a los recién introducidos.
- **Apuesta por la compatibilidad.** C# mantiene una sintaxis muy similar a C++ o Java que permite, bajo ciertas condiciones, incluir directamente en código escrito en C# fragmentos de código escrito en estos lenguajes.

### 1.5.1.3 Breve sinopsis de C [17]

El lenguaje C es creado entre los años 1972-1973 por un equipo de programadores de los antiguos laboratorios Bell de AT&T. Dennis Ritchie diseñó e implementó el primer compilador de lenguaje C en un computador PDP-11. El lenguaje C se basó en dos lenguajes (prácticamente desaparecidos): "BCPL", escrito por Martin Richards, y "B", escrito por Ken Thompson en 1970 para el primer sistema UNIX en un PDP-7.

El lenguaje C originalmente "oficial" fue el "K&R C". Hacia 1988-1989, el American National Standards Institute (ANSI) adoptó una versión mejorada de C, conocida hasta hoy como "ANSI C" o C89. La versión ANSI C contiene una importante revisión a la sintaxis, especialmente para el uso de funciones, así como la estandarización de las librerías del sistema.

La última revisión del lenguaje C es conocida como "C99" y no está soportada en forma completa por la mayoría de compiladores, por lo cual gran parte de programadores siguen empleando sólo lo que proporciona la versión "C89".

### Características generales de C

El lenguaje C se conoce como un lenguaje de medio nivel, pues podríamos situarlo entre los lenguajes de bajo nivel o de máquina (ensamblador) y los de alto nivel como el PASCAL, por ejemplo. Un lenguaje de

medio nivel nos ofrece un conjunto básico de sentencias de control y de manipulación de datos que nos permitirá construir sentencias y estructuras de nivel más alto. En la actualidad existen numerosos compiladores de C, todos ellos con sus peculiaridades, sin embargo, prácticamente todos son compatibles con el C normalizado por ANSI, el ANSIC. Entre las características del C podemos citar:

- 32 palabras clave (BASIC128; TPASCAL48).
- Eficiencia de ejecución del código generado (entre los lenguajes de bajo y alto nivel).
- Portabilidad.
- No impone tantas restricciones como los lenguajes de alto nivel, dando más libertad al programador. C es un lenguaje estructurado como PASCAL, y como tal, un programa en C cumple todas las características de la programación estructurada. De hecho, en C todo el código se estructura en funciones totalmente independientes entre sí.

### 1.5.1.4 Sinopsis de C++ [18]

En 1983 el lenguaje se rebautiza como C++ y en 1985 Stroustrup publica la primera edición del libro "The C++ Programming Language" que sirvió de estándar informal y texto de referencia. Posteriormente el lenguaje fue estandarizado (ISO C++) y paralelamente son publicadas la segunda y tercera ediciones de "The C++ Programming Language" de modo tal que reflejan estos cambios.

Desde sus inicios, C++ intentó ser un lenguaje que incluye completamente al lenguaje C pero al mismo tiempo incorpora muchas características sofisticadas no incluidas en aquél, tales como: POO, excepciones, sobrecarga de operadores, templates o plantillas.

Dentro de las características que tiene C++ se encuentran:

- Es un lenguaje independiente, aunque se conserva la compatibilidad con C.
- Es un lenguaje fuertemente tipado.
- Soporta multitarea mediante clases.

### 1.5.1.5 Lenguajes seleccionados: C/C++

Luego de realizar un estudio preliminar de algunos lenguajes de programación los investigadores decidieron utilizar los lenguajes en conjuntos C/C++ pues presentan mayor cantidad de librerías además

de una serie de características que a continuación serán expuestas en la tabla 1.1 teniendo en cuenta los siguientes criterios:

1. Expresividad: Facilidad del lenguaje para expresar los algoritmos
2. Bien Definido: Consistencia y falta de ambigüedad
3. Tipos y estructuras de datos
4. Modularidad: Permitir el desarrollo de componentes independientemente
5. Facilidades de entrada-salida: Soporte para interacción con el entorno
6. Transportabilidad/Portabilidad
7. Eficiencia/Performance
8. Pedagogía: Facilidad de aprendizaje y enseñanza
9. Generalidad: Aplicabilidad, Uso

**Tabla 1.1 Resumen [19]**

Características	C	C++
Expresividad	Regular	Muy Buena a Excesiva
Bien Definido	Regular	Muy Buena
Tipos y estructuras de datos	Deficiente	Muy Buena
Modularidad	Regular	Muy Buena
Facilidad de entrada/salida	Buena	Buena
Transportabilidad/ Portabilidad	Buena	Buena

Eficiencia/Perfomance	Excelente	Excelente
Pedagogía	Regular	Regular
Generalidad	Buena	Muy Buena
Estandarización	Buena	Buena
Evolución	Estable	Estable
Soporte de Librería	Bueno	Muy Bueno

### **1.5.2 Gestores de base de datos [20]**

Un gestor de base de datos o sistema de gestión de base de datos (SGBD o DBMS) es un software que permite introducir, organizar y recuperar la información de las bases de datos; en definitiva, administrarlas. Existen distintos tipos de gestores de bases de datos: relacional, jerárquico, red, etc. El modelo relacional es el utilizado por casi todos los gestores de bases de datos para PC's.

El modelo relacional (SGBDR) es un software que almacena los datos en forma de tablas.

Existen actualmente SGBD libres y comerciales, dentro de los libres se encuentran PostgreSQL y MySQL, mientras que en el otro grupo se encuentran Oracle, SQL-Server etc.

#### **1.5.2.1 MySQL**

Es un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca. MySQL es un software de código abierto, licenciado bajo la GPL de la GNU.

El lenguaje de programación que utiliza MySQL es Structured Query Language (SQL) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales[21].

### Características principales

Inicialmente, MySQL carecía de algunos elementos esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de esto, atrajo a los desarrolladores de páginas web con contenido dinámico, debido a su simplicidad, de tal manera que los elementos faltantes fueron complementados por la vía de las aplicaciones que la utilizan. Poco a poco estos elementos faltantes, están siendo incorporados tanto por desarrolladores internos, como por desarrolladores de software libre[22].

En las últimas versiones se pueden destacar las siguientes características principales:

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
- Flexible sistema de contraseñas (passwords) y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- El servidor soporta mensajes de error en distintas lenguas

### Ventajas de MYSQL

- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Conectividad y seguridad[23].

### Desventajas de MySQL

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo, como otros programas (ACCESS)[24].

### 1.5.2.2 PostgreSQL [25]

PostgreSQL es un potente sistema de base de datos de código abierto objeto-relacional. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de fiabilidad, integridad de datos y corrección. Se ejecuta en los principales sistemas operativos, incluyendo Linux, UNIX (AIX, BSD de HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows. Es totalmente compatible con ACID, tiene soporte completo para claves foráneas, uniones, vistas, disparadores y procedimientos almacenados (en varios idiomas). Soporta la mayor parte de los tipos de datos de SQL: 2008, incluyendo integer, numeric, boolean, char, varchar, date, interval y timesamp. También soporta almacenamiento de objetos binarios grandes, como imágenes, sonidos o vídeo. Cuenta con interfaces de programación nativo de C / C + +, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, entre otros, y documentación de carácter excepcional.

Una base de datos de clase empresarial, PostgreSQL cuenta con sofisticadas funciones como Multi-Versión Control de concurrencia (MVCC), punto en el tiempo de recuperación, replicación asincrónica, transacciones anidadas (puntos de retorno), un planeador/ optimizador de consultas sofisticadas, y escribir por delante de registro para tolerancia a fallos. Es escalable, tanto en la gran cantidad de datos que puede manejar y en el número de usuarios concurrentes que puede acomodar.

Hay sistemas activos de PostgreSQL en entornos de producción que manejan en exceso de 4 terabytes de datos. Algunos límites generales de PostgreSQL, están incluidos en la tabla de abajo:

Límite	Valor
Tamaño máximo de la base de datos	32 TB
Tamaño máximo de la tabla	1.6 TB

Tamaño máximo de campo	1 GB
Máximo de Filas por tabla	Ilimitado
Máximo de columnas por tablas	250 - 1600 en función de los tipos de columna
Máximo de índices por tablas	Ilimitado

El código fuente de PostgreSQL está disponible bajo los liberales una licencia de código abierto: la licencia de PostgreSQL. Esta licencia le concede la libertad de usar, modificar y distribuir PostgreSQL en cualquier forma que desee, abierto o de código cerrado. Puede hacerse las modificaciones, mejoras o cambios que se desee realizar.

El equipo de trabajo opta por usar PostgreSQL como SGBD pues este no es sólo un sistema de base de datos de gran alcance capaz de ejecutar multitud de transacciones simultáneas sino que también es una plataforma de desarrollo sobre altamente recomendada a los que desarrollan web, software o productos comerciales que requieran un gestor de base de datos eficiente para el trabajo con grandes volúmenes de datos.

### 1.5.3 Lenguaje de modelado [26]

Cuando se realiza un modelo se trata de simplificar la realidad capturando lo esencial y llevándolo a una representación sencilla pero entendible a la vez.

#### Lenguaje Unificado de Modelado

UML constituye un lenguaje estándar diseñado para visualizar, especificar, construir documentar software orientado a objetos.

UML es una consolidación de muchas de las notaciones y conceptos más usados orientados a objetos. Este tipo de modelado puede ser empleado para modelar variados tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML consta de nueve diagramas para modelar estos sistemas.

Todo sistema puede describirse desde distintos puntos de vista:

- Modelos estructurales (organización del sistema)
- Modelos de comportamiento (dinámica del sistema)

UML estandariza 9 tipos de diagramas para representar gráficamente un sistema desde distintos puntos de vista.

- Diagramas de clases: Muestran un conjunto de clases y sus relaciones, los diagramas de clases proporcionan una perspectiva estática del sistema (representan su diseño estructural)
- Diagramas de objetos: Muestran un conjunto de objetos y sus relaciones (una situación concreta en un momento determinado). Los diagramas de objetos representan instantáneas de instancias de los elementos que aparecen en los diagramas de clases.
- Diagramas de interacción: Muestran una interacción concreta: un conjunto de objetos y sus relaciones, junto con los mensajes que se envían entre ellos.
- Diagramas de secuencia: Muestran la secuencia de mensajes entre objetos durante un escenario
- Diagramas de comunicación/colaboración: Resaltan la organización estructural de los objetos que intercambian mensajes.

Los diagramas de secuencia y de comunicación son isomorfos:

- Un diagrama de secuencia se puede transformar mecánicamente en un diagrama de comunicación.
- Un diagrama de comunicación se puede transformar automáticamente en un diagrama de secuencia.

### **Otros diagramas UML para representar aspectos dinámicos del sistema**

- Diagramas de casos de uso (actores y casos de uso del sistema): Los diagramas de uso se suelen utilizar en el modelado del sistema desde el punto de vista de sus usuarios para representar las acciones que realiza cada tipo de usuario.
- Diagramas de estados (estados y transiciones entre estados): Los diagramas de estados son especialmente importantes para describir el comportamiento de un sistema reactivo (cuyo

comportamiento está dirigido por eventos).

- Diagramas de actividades (flujo de control en el sistema): Los diagramas de actividades muestran el orden en el que se van realizando tareas dentro de un sistema (el flujo de control de las actividades).

### Diagramas UML para representar aspectos físicos del sistema

- Diagramas de componentes (componentes y dependencias entre ellos): Organización lógica de la implementación de un sistema.
- Diagramas de despliegue (nodos de procesamiento y componentes): Configuración del sistema en tiempo de ejecución.

### 1.6 Metodología de desarrollo

En un proyecto de desarrollo de software la metodología define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo. Por la importancia que tiene el uso de una metodología adecuada que guíe el proceso de desarrollo de un software la selección de la misma debe realizarse cuidadosamente. Las metodologías son utilizadas fundamentalmente con el objetivo de lograr un producto final que cumpla con los requisitos establecidos por los clientes y aumentar la calidad del software controlando su ciclo de vida.

Con el fin de seleccionar una metodología adecuada, en el presente trabajo se investigó sobre dos de los métodos ágiles más conocidos Programación Extrema ( XP) y Scrum y la metodología pesada más conocida Proceso Unificado de Desarrollo(RUP).

#### 1.6.1 Programación Extrema (XP) [27]

El proyecto de Programación Extrema o como su nombre en inglés Extreme Programming (XP), tuvo sus inicios el 6 de marzo 1996 y en la actualidad es considerada como una de las metodologías ágiles más populares en el desarrollo de software. Ya se ha demostrado su gran éxito en muchas empresas de todos los tamaños e industrias en todo el mundo.

XP es un éxito porque hace hincapié en la satisfacción del cliente. Extreme Programming les da a los desarrolladores del software confianza a la hora de responder a las cambiantes necesidades de los

clientes, incluso a finales del ciclo de vida. Los administradores, clientes y desarrolladores son socios iguales en un equipo de colaboración. Extreme Programming implementa un equipo simple, pero efectivo ambiente propicio para llegar a ser altamente productiva.

Esta metodología mejora un proyecto de software en cinco aspectos esenciales: la comunicación, la sencillez, la retroalimentación, el respeto y el coraje:

- **Comunicación:** Todo el mundo es parte del equipo existiendo una comunicación cara a cara todos los días. El equipo trabaja juntos en todo. Juntos se crea la mejor solución al problema.
- **Simplicidad:** Se hace lo que sea necesario y solicitado, pero no más. Esto maximizará el valor creado para la inversión realizada hasta la fecha. Se toman pequeñas medidas simples para lograr los objetivos y mitigar los fracasos a medida que ocurren.
- **Retroalimentación:** Existencia de una retroalimentación entre el grupo de trabajo y el cliente. Se muestran con frecuencia los adelantos del software y se escucha con atención tanto al equipo de trabajo como al cliente.
- **Respeto:** Todo el proyecto da y siente el respeto que merece como miembro del equipo. Todo el mundo aporta un valor incluso si es simplemente el entusiasmo.
- **Valor:** Vamos a decir la verdad sobre los avances y estimaciones. No documento que sirvan de excusas para un posible fracaso, pero si un plan para tener éxito. Nadie es culpable de nada porque nadie trabaja solo. Adaptación total a los cambios, cada vez que ocurran.

### **El Reglamento de la Programación Extrema**

#### **Planificación**

- Las Historias de Usuarios están escritas.
- Realiza frecuentes versiones pequeñas
- El proyecto se divide en iteraciones.
- La planificación de la iteración se inicia en la misma iteración.

#### **Gestión**

- El equipo tendrá dedicado un espacio de trabajo abierto

- Establecimiento de un ritmo sostenible.
- La velocidad del proyecto es medida.
- Mover a las personas.

### Diseñar

- Simplicidad
- Usar Tarjetas CRC para las sesiones de diseño.
- Crear una solución punta para reducir riesgos.
- Ninguna funcionalidad es añadida al principio.
- Refactorizar cuando y donde sea posible.

### Codificación

- El cliente está siempre disponible.
- El código debe ser escrito con las normas acordadas.
- Establecer un equipo dedicado a la integración de código.
- La propiedad es de uso colectivo.

### Pruebas

- Todo el código debe tener pruebas de unidad.
- Todo el código debe pasar las pruebas de unidad antes de que pueda ser liberado.
- Las pruebas de aceptación se ejecutan a menudo y la puntuación se publica.

XP no es en realidad un conjunto de reglas, sino más bien una manera de trabajar en armonía con sus valores personales y corporativos.

### ¿Cuándo se debe usar la metodología Extreme Programming?

XP se creó para abordar los problemas de riesgo del proyecto. Si sus clientes necesitan un nuevo sistema en una fecha determinada, el riesgo es alto. Si ese sistema es un nuevo reto para el grupo de software, el riesgo es aún mayor. Si ese sistema es un nuevo desafío para la industria de software de todo el riesgo es

incluso mayor todavía. Las prácticas de XP son creados para mitigar el riesgo y aumentar la probabilidad de éxito.

XP está configurado para pequeños grupos de programadores. Entre 2 y 12, aunque los proyectos más grandes en los que se ha utilizado han sido de 30 y han tenido éxito. Sus programadores pueden ser ordinarios, pues no se necesita de programadores con un doctorado para utilizar XP. Pero no se puede usar XP en un proyecto con un equipo de trabajo grande.

XP requiere un equipo de desarrollo extendido. El equipo de XP incluye no sólo a los desarrolladores, también a los administradores y clientes, así, todos trabajando juntos codo a codo. Hacer preguntas, la negociación de ámbito de aplicación y los horarios, y la creación de pruebas funcionales requieren más que los desarrolladores estén solamente involucrados en la producción del software.

Otro requisito es la capacidad de prueba. Usted debe ser capaz de crear unidad automatizada y pruebas funcionales. Lo último en la lista es la productividad. Los proyectos XP por unanimidad reportan informe con una mayor productividad del programador en comparación con otros proyectos en el entorno de la misma empresa. Aunque esto no fue el fin de esta metodología pues el objetivo real siempre ha sido entregar el software que se necesita cuando se necesita.

### 1.6.2 Scrum [28]

Scrum es una metodología ágil de desarrollo de proyectos que toma su nombre y principios de los estudios realizados sobre nuevas prácticas de producción por Hirotaka Takeuchi e Ikujiro Nonaka a mediados de los 80.

Aunque surgió como modelo en el desarrollo de productos tecnológicos, sus principios son válidos para entornos que trabajan con requisitos inestables, y necesitan agilidad: situaciones frecuentes en el desarrollo de determinados sistemas de software.

Jeff Sutherland aplicó los principios de los campos de Scrum al desarrollo de software en 1993 en Easel Corporation (Empresa que en los macro-juegos de compras y fusiones se integraría en VMARK, luego en Informix y finalmente en Ascential Software Corporation). En 1996 presentó junto con Ken Schwaber, las prácticas que empleaba como válidas para gestionar el desarrollo de software OOPSLA 96 (Schwaber & Sutherland, 1996).

### Características de Scrum

Varias de las características que presenta la metodología Scrum son:

- Se dividen los productos grandes en pedazos manejables - un producto consta de pocas características con el fin de que los equipos pequeños puedan crear el producto en pocos meses.
- Facilita la competencia basada en la retroalimentación de los clientes, las características del producto, y la duración del tiempo de desarrollo, proporcionando un mecanismo para incorporar aportes de los clientes.
- Continuamente se puede probar el producto.

### Principios de la metodología Scrum

Los principios fundamentales del proceso de desarrollo Scrum son:

- Pequeños equipos de trabajo que maximizan la comunicación, reducen al mínimo los gastos generales y maximizan el intercambio de conocimientos tácitos e información.
- La adaptabilidad al usuario técnico o mercado (y cliente) cambios para asegurar la posible producción de un mejor producto.
- La construcción de ejecutables, que pueden ser inspeccionados, ajustado, documentado y construidos.
- Compartimentación del trabajo y las tareas en el equipo de trabajo.

**El funcionamiento interno de Scrum:** Scrum está compuesto por procesos de desarrollo y por las mediciones que se utilizan para controlar el trabajo en el proyecto. La clave para el éxito de Scrum es utilizando las medidas de control para maximizar la flexibilidad y minimizar los riesgos.

#### 1.6.3 Proceso Unificado de Desarrollo (RUP) [29]

RUP es una metodología de desarrollo de software que intenta integrar todos los aspectos a tener en cuenta durante el ciclo de vida del software, con el objetivo de hacer abarcables tanto pequeños como grandes proyectos de software. Además Rational proporciona herramientas para todos los pasos del desarrollo así como documentación en línea para sus clientes.

Las características principales de RUP son:

- Guiado o manejado por casos de uso: la razón de ser de un sistema software, es servir a usuarios, ya sean humanos u otros sistemas; un caso de uso es una facilidad que el software debe proveer a sus usuarios. Los casos de uso reemplazan la antigua especificación funcional tradicional y constituyen la guía fundamental establecida para las actividades a realizar durante el proceso de desarrollo, incluyendo el diseño, la implementación y las pruebas del sistema.
- Centrado en la arquitectura: la arquitectura involucra los elementos más significativos del sistema y está influenciada entre otros, por plataformas, software, sistemas operativos, manejadores de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados y requerimientos no funcionales. Los casos de uso guían el desarrollo de la arquitectura y la arquitectura se realimenta en los casos de uso, los dos juntos permiten conceptualizar, gestionar y desarrollar adecuadamente el software.
- Iterativo e incremental: para hacer más manejable un proyecto se recomienda dividirlo en ciclos. Para cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un mini proyecto, cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo.

RUP está constituido por cuatro fases que representan el ciclo de desarrollo del software y nueve flujos de trabajo, seis ingenieriles y tres de soporte que intervienen dentro de cada fase.

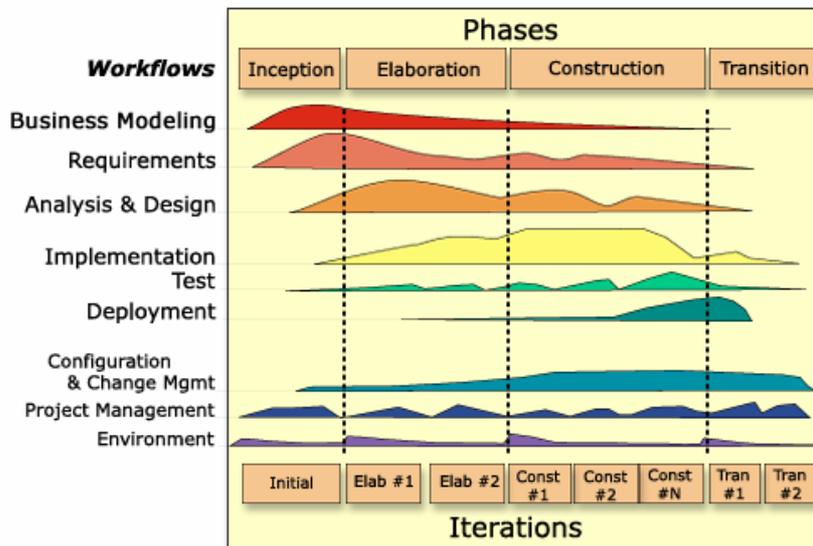


Figura 1.1 Fases, Flujos e iteraciones de RUP

#### 1.6.4 Proceso Unificado de Desarrollo (RUP) y Programación extrema (XP) [30]

Mientras que RUP intenta reducir la complejidad del software por medio de estructura y la preparación de las tareas pendientes en función de los objetivos de la fase y la actividad actual, XP, como metodología ágil, solo intenta por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción.

XP intenta minimizar el riesgo de fallo del proceso por medio de la disposición permanente de un representante del cliente a disposición del equipo de desarrollo. Este representante debería estar en condiciones de contestar rápida y correctamente a cualquier pregunta del equipo de desarrollo de forma que no se retrase la toma de decisiones. A continuación una breve comparación de XP y RUP como metodologías de desarrollo de software.

#### Tamaño de los equipos:

RUP está pensado para proyectos y equipos grandes, en cuanto a tamaño y duración. XP se implementa mejor para proyectos cortos y equipos más pequeños.

### **Obtención de requisitos:**

RUP y XP ambos describen los requerimientos de la aplicación desde el punto de vista del usuario. Ambos definen los requisitos técnicos sin meterse con detalles de la implementación.

### **Carga de trabajo:**

En el desarrollo de un proyecto con XP es más importante la entrega al cliente del software que necesita (lo que a menudo no es lo mismo que lo descrito en el documento de especificación de requisitos) que las funcionalidades que quedan por implementar. Cuando durante el desarrollo se descubre que se deben cambiar las funcionalidades, se acuerda directamente con el representante del cliente. Con esos cambios de ajustará el plan de iteraciones y el de releases y se tomará la nueva dirección en el desarrollo.

RUP es un proceso pesado, basado mucho en la documentación, en la que no son deseables todos esos cambios volátiles. Existen diferentes elementos de planificación con los que se controla el desarrollo del software. A través de un predefinido esquema de escalabilidad y gestión de riesgos, se pueden reconocer previamente problemas y fallos de forma temprana y nivel de detalles, y por qué rol, se deben crear. Se definirán que artefactos son necesarios para poder realizar una actividad y que artefactos se deberán crear durante dicha actividad.

### **Relación con el cliente:**

Con RUP se le presentarán al cliente los artefactos del final de una fase y se valorarán las precondiciones para la siguiente (definición de riesgos, aceptación de plan de iteraciones, prototipos, etc) y solo después de que el cliente acepte los artefactos generados se pasará a la siguiente fase. La calidad de los artefactos generados será probada durante la totalidad del ciclo de vida del proyecto a través de distintas medidas de calidad, como convenciones, revisiones y auditorías periódicas, pruebas, etc.

En contrapartida, la aseguración de la calidad en XP no se basa en formalismos en la documentación, si no en controles propios y una comunicación fluida con el cliente. El cliente recibe después de cada iteración un pedazo funcional del programa. A través de un ciclo de iteración corto (pocas semanas) el cliente está informado constantemente sobre la situación del proyecto y puede intervenir rápidamente si el desarrollo se aleja de sus necesidades.

### Código fuente

XP es el único que presenta la compartición del código, mayormente debido a que esta pensado en equipos pequeños que trabajan conjuntamente y en comunicación constante e inmediata, mientras que RUP opta por la propiedad del código, aunque define grupos y sesiones de trabajo conjuntos de forma que los propietarios de clases relacionadas trabajan juntos, y la comunicación es directa.

### Resume de puntos claves

#### RUP

- Pesado
- Dividido en cuatro fases
- La fase se dividen en iteraciones
- El discurrir del proyecto se define en WorkFlows (flujos de trabajo)
- Los artefactos son el objetivo de cada actividad
- Se basa en roles
- UML
- Muy organizativo
- Mucha documentación

#### XP

- Ligero
- Cercano al desarrollos
- Se basa en UserStories(Historias de Usuarios)
- Fuerte comunicación con el cliente
- El código fuente pertenece a todos
- Programación por parejas
- Test como base de la funcionalidad
- Pobre en cuanto a documentación
- Solo el mínimo de organización

### 1.6.5 Scrum-XP (SXP)

Teniendo en cuenta la necesidad e importancia de utilizar una metodología, en el caso particular del ciclo de vida del producto módulo de procesamiento de Airesweb se decidió utilizar la unión de Scrum- XP. La decisión final quedó sustentada por los siguientes elementos: La unión de estas dos metodologías ágiles (Scrum y XP) darán el seguimiento, apoyo y hegemonía correcta al proyecto pues Scrum puede servir como la interfaz entre los clientes y el equipo de desarrollo mientras que XP controla el trabajo diario del equipo y entre los dos se hace más rápido detectar y corregir los errores que puedan surgir en el ciclo de vida del producto. Además se tuvo en cuenta que el equipo de trabajo es pequeño.

## 1.7 Herramientas

### 1.7.1 Herramienta de programación

Actualmente existen muchas herramientas de programación que brindan al implementador la comodidad para desarrollar software y productos para un mercado que crece cada día más.

Una de las cosas a tener en cuenta a la hora de programar es seleccionar un IDE correcto y que apoye al desarrollador en términos como el completamiento de código y el detector de errores.

#### 1.7.1.1 Eclipse [31]

Eclipse es un IDE conocido para el desarrollo de aplicaciones Java. Dicho entorno de desarrollo es de código abierto y multiplataforma. Esta herramienta de programación fue creado originalmente por IBM en noviembre del 2001 y apoyado por un consorcio de proveedores de software, es un IDE más flexible de lo imaginado, pues gracias a una infinidad de plugins éste permite, entre otras cosas, editar clases visuales de Java, programar aplicaciones J2EE, C/C++ y, en varios lenguajes más, conectarse a bases de datos y escribir consultas SQL, etc.

#### Características de Eclipse, como IDE de desarrollo:

- Posee una excelente integración con herramientas para el control de versiones.
- Posee un potente administrador de proyecto.
- Es multi-plataforma.
- Cuenta con derivados plataforma para COBOL, C / C + + y Perl

- El núcleo de Eclipse IDE es robusto y basado en normas.

### 1.7.1.2 Netbeans [32]

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados *módulos*. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

NetBeans es el principal competidor en el campo de código abierto frente a Eclipse. Netbeans es una iniciativa de Sun y tiene características de base similares a la herramienta de Sun Studio Creator. Sun fue el lanzador de Netbeans.

NetBeans es un proyecto de código abierto con una impresionante cantidad de características y capacidades. Las versiones anteriores de Netbeans han sufrido el hecho de ser más lentas que esta. El IDE NetBeans tiene una interfaz clara e intuitiva.

### 1.7.2 Herramienta de modelado [33]

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Presenta un entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de casos de usos.

Está diseñado para una amplia gama de usuarios, incluidos los Ingenieros de Software, Analistas de Sistema, Analistas de Negocios, Sistema de Arquitectos igual, que estén interesados en la construcción

de sistemas de software a gran escala mediante el uso fiable, es Orientado a Objetos y facilita la comunicación entre ellos.

Visual Paradigm soporta un conjunto de lenguajes, tanto en la generación de código como en la ingeniería inversa. Puede generar código a partir de los modelos y viceversa. Cualquiera de los cambios que se realicen en el código existente puede reflejarse en el modelo. Esta herramienta visual permite construir la aplicación con mayor rapidez, mayor exactitud, mejor trabajo en equipo y fácil de utilizar, además de que aumenta las expectativas mediante la interfaz gráfica.

Facilita la interoperabilidad con otras herramientas CASE, la mayoría de los IDE's principalmente y permite la integración de todos los componentes. Aunque es una herramienta gratuita, este programa se encuentra bajo licencias que no permiten el estudio ni modificación de la misma.

Por las características antes expuestas ha sido seleccionada por el equipo de trabajo como herramienta de modelado.

### 1.8 Conclusiones

En este capítulo se han cumplido los objetivos establecidos al inicio del mismo, el cual consistía en investigar y documentar información que ayude a la implementación del módulo para Airesweb. También se hizo una valoración de herramientas, tecnologías y lenguajes con el fin de seleccionar los más adecuados respectivamente para la implementación y control del producto en cuestión incluyendo en la misma la selección de la metodología a utilizar en el ciclo de vida del software.

Al finalizar el capítulo se concluye que como lenguaje de programación para la implementación del módulo se utilizará C/C++ por presentar gran soporte de librerías, en cuanto al IDE de desarrollo se decidió por Eclipse, como herramienta CASE el Visual Paradigm, PostgreSQL como gestor de base de datos pues resalta la característica de que cuenta con interfaces de programación nativo de C/C++ y como metodología de desarrollo la unión de Scrum-XP por su constante intercambio con los clientes y la hegemonía que esto traerá consigo al equipo de trabajo.

### **Capítulo 2. Características del sistema.**

Cuando se desarrolla un sistema es importante que los involucrados tengan establecido un entendimiento para llegar a fines que satisfagan tanto al cliente como al equipo de trabajo, las características del sistema, aplicación o software deben quedar claramente redactadas con el objetivo de realizar un producto con la calidad requerida. Para la implementación del módulo de procesamiento de datos para Airesweb y utilizando Scrum-XP como metodología de desarrollo en este capítulo se realizará una descripción de la solución propuesta, el diagrama de componentes y el levantamiento de requisitos dejando conformadas las Historias de Usuarios y sus respectivas descripciones.

#### **2.1 Descripción del sistema**

Anteriormente, en el capítulo 1, donde se realizó una investigación asociada al objeto de estudio y se expuso la necesidad de implementar un módulo procesador de datos para el sistema Airesweb.

Airesweb es un software que se incluye dentro de la analítica web y que analiza los archivos de registro generando estadísticas a partir de ellos. Estos reportes son guardados en la base de datos del sistema permitiendo tener conocimiento a los interesados sobre el comportamiento del sitio monitorizado y las acciones que se realizan en él. Tal proceso actualmente no es muy ágil, se necesita de una mayor optimización en el traslado de los datos desde el archivo de registro hacia la base de datos.

#### **2.2 Solución propuesta**

La solución a lo antes planteado está en la implementación del módulo que optimice el traslado de los reportes del sitio monitorizado hacia la base de datos del sistema Airesweb.

Las Historias de Usuarios que se listan seguidamente con sus respectivas descripciones abordan más específico sobre las funcionalidades con las que contará el módulo de procesamientos de datos para Airesweb.

#### **2.3 Lista de reserva del producto**

La tabla 2.1 Lista de reserva del producto que a continuación se muestra relaciona las Historias de Usuario con la prioridad que tienen, la estimación del tiempo para efectuarlas en días y los usuarios que se encargan de desarrollarlas.

Prioridad	Ítem *	Descripción	Estimación (días)	Estimado por
<b>Alta</b>				
	1	Establecer conexión con la Base de Datos.	1	PROG
	2	Inicializar parámetros del módulo.	1	PROG
	3	Explorar directorio de archivos de registros.	2	PROG
	4	Analizar archivo de registros.	1	PROG
	5	Insertar URL y obtener id.	1	PROG
	6	Estimar visita.	1	PROG
	7	Insertar visita en la base de datos.	1	PROG
	8	Insertar acciones en la base de datos.	1	PROG
	9	Chequear si un visitante es recurrente.	1	PROG
	10	Generar respaldo de visitas no completadas.	1	PROG
	11	Restaurar visitas no completadas desde fichero de respaldo.	1	PROG

Tabla 2.1 Lista de reserva del producto.

## 2.4 Descripción de las Historias de Usuarios

En este epígrafe se realiza una descripción de las Historias de Usuarios y las tareas ingenieriles que cada una tiene asignada además del responsable del cumplimiento de dicha tarea.

Historia de Usuario	
<b>Número:</b> 1	<b>Nombre Historia de Usuario:</b> Establecer conexión con la base de datos.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yordan Suarez Santiago	<b>Iteración Asignada:</b> sprint 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 2 semana
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1 semana
<b>Descripción:</b> Se necesita realizar la conexión con la base de datos del sistema Airesweb por lo que es necesario implementar dicha conexión.	
<b>Observaciones:</b>	
<b>Prototipo de interfase:</b>	

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Realizar estudio de las bibliotecas para conectarse a la base de datos.	
<b>Tipo de Tarea :</b> Investigativa	<b>Puntos Estimados:</b> 1 semana
<b>Fecha Inicio:</b> 04/02/2010	<b>Fecha Fin:</b> 11/02/2010
<b>Programador Responsable::</b> Dunia V. Ruz Góngora	
<b>Descripción:</b> Se realiza una investigación de las bibliotecas que se utilizan para conectarse a la base de datos.	

Historia de Usuario	
<b>Número:</b> 2	<b>Nombre Historia de Usuario:</b> Inicializar parámetros del módulo.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yordan Suarez Santiago	<b>Iteración Asignada:</b> sprint 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 3 semana
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1 semana
<b>Descripción:</b> Leer ficheros de configuración para inicializar los parámetros del módulo.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

Tarea de Ingeniería	
<b>Número Tarea:</b> 2	<b>Número Historia de Usuario:</b> 2
<b>Nombre Tarea:</b> Realizar estudio de las librerías para inicializar parámetros del módulo.	
<b>Tipo de Tarea :</b> Investigativa	<b>Puntos Estimados:</b> 1 semana
<b>Fecha Inicio:</b> 12/02/2010	<b>Fecha Fin:</b> 19/02/2010
<b>Programador Responsable::</b> Dunia V. Ruz Góngora	
<b>Descripción:</b> Se realiza una investigación de las bibliotecas que se utilizan para inicializar parámetros del módulo.	

Historia de Usuario
---------------------

<b>Número:</b> 3	<b>Nombre Historia de Usuario:</b> Explorar directorio de archivos de registros.	
<b>Modificación de Historia de Usuario Número:</b> Ninguna		
<b>Usuario:</b> Yordan Suarez Santiago	<b>Iteración Asignada:</b> sprint 1	
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 2 semana	
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1 semana	
<b>Descripción:</b> Se recorre el directorio y se toman los archivos de registros para futuro procesamiento.		
<b>Observaciones:</b>		
<b>Prototipo de interfaz:</b>		

		Tarea de Ingeniería
<b>Número Tarea:</b> 3	<b>Número Historia de Usuario:</b> 3	
<b>Nombre Tarea:</b> Implementar la clase Directory		
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 1 semana	
<b>Fecha Inicio:</b> 20/02/2010	<b>Fecha Fin:</b> 27/02/2010	
<b>Programador Responsable::</b> Yordan Suarez Santiago		
<b>Descripción:</b> Se implementa la clase directory correspondiente a la gestión de todos los archivos de registros.		

<i>Historia de Usuario</i>
----------------------------

<b>Número:</b> 4	<b>Nombre Historia de Usuario:</b> Analizar archivo de registros.		
<b>Modificación de Historia de Usuario Número:</b> Ninguna			
<b>Usuario:</b> Yordan Suarez Santiago		<b>Iteración Asignada:</b> sprint 1	
<b>Prioridad en Negocio:</b> Alta		<b>Puntos Estimados:</b> 2 semana	
<b>Riesgo en Desarrollo:</b> Bajo		<b>Puntos Reales:</b> 1 semana	
<b>Descripción:</b> Dada la necesidad de explorar y revisar los ficheros de registros, los cuales arrojarán datos significativos, se debe realizar la implementación para analizar dichos archivos de registros.			
<b>Observaciones:</b>			
<b>Prototipo de interfaz:</b>			

Tarea de Ingeniería			
<b>Número Tarea:</b> 4		<b>Número Historia de Usuario:</b> 4	
<b>Nombre Tarea:</b> Implementar la clase Parser			
<b>Tipo de Tarea :</b> Desarrollo		<b>Puntos Estimados:</b> 1 semana	
<b>Fecha Inicio:</b> 01/03/2010		<b>Fecha Fin:</b> 08/03/2010	
<b>Programador Responsable::</b> Yordan Suarez Santiago			
<b>Descripción:</b> Se implementa las clases Parser que analizar los archivos de registros.			

Tarea de Ingeniería			
<b>Número Tarea:</b> 5		<b>Número Historia de Usuario:</b> 4	
<b>Nombre Tarea:</b> Implementar la clase ParsedLine			

<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 2 semana
<b>Fecha Inicio:</b> 09/03/2010	<b>Fecha Fin:</b> 16/03/2010
<b>Programador Responsable::</b> Yordan Suarez Santiago	
<b>Descripción:</b> Se implementa las clases ParsedLine que analiza por líneas los archivos de registros.	

Historia de Usuario	
<b>Número:</b> 5	<b>Nombre Historia de Usuario:</b> Insertar URL y obtener id.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yordan Suarez Santiago	<b>Iteración Asignada:</b> sprint 2
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 4 semana
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 2 semana
<b>Descripción:</b> Se inserta la URL desde la cual han accedido al sitio monitorizado y se obtiene el id del mismo.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

Tarea de Ingeniería	
<b>Número Tarea:</b> 6	<b>Número Historia de Usuario:</b> 5
<b>Nombre Tarea:</b> Implementar la clase insertURL	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 1 semana
<b>Fecha Inicio:</b> 17/03/2010	<b>Fecha Fin:</b> 23/03/2010

<b>Programador Responsable::</b> Yordan Suarez Santiago
<b>Descripción:</b> Se implementa la clase insertURL que inserta en la base de datos la URL y guarda el ID obtenido de los archivos de registros.

Historia de Usuario	
<b>Número:</b> 6	<b>Nombre Historia de Usuario:</b> Estimar visita.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yordan Suarez Santiago	<b>Iteración Asignada:</b> sprint 2
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 3 semana
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1 semana
<b>Descripción:</b> Tomando el id desde los archivos de registros se realiza la estimación de una visita.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

Tarea de Ingeniería	
<b>Número Tarea:</b> 7	<b>Número Historia de Usuario:</b> 6
<b>Nombre Tarea:</b> Investigar concepto de visita.	
<b>Tipo de Tarea :</b> Investigativa	<b>Puntos Estimados:</b> 1 semana
<b>Fecha Inicio:</b> 24/03/2010	<b>Fecha Fin:</b> 31/03/2010
<b>Programador Responsable::</b> Dunia V. Ruz Góngora	
<b>Descripción:</b> Se realiza una investigación de los conceptos de visitas en el ámbito de la analítica	

web.

Historia de Usuario	
<b>Número:</b> 7	<b>Nombre Historia de Usuario:</b> Insertar visita en la base de datos.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yordan Suarez Santiago	<b>Iteración Asignada:</b> sprint 2
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 3 semana
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1 semana
<b>Descripción:</b> Se inserta en la base de datos las visitas que ha recibido el sitio monitorizado.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

Tarea de Ingeniería	
<b>Número Tarea:</b> 8	<b>Número Historia de Usuario:</b> 7
<b>Nombre Tarea:</b> Implementar la clase InsertVisit.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 1 semana
<b>Fecha Inicio:</b> 01/04/2010	<b>Fecha Fin:</b> 08/04/2010
<b>Programador Responsable::</b> Yordan Suarez Santiago	
<b>Descripción:</b> Se implementa las clases InsertVisit donde serán registradas y gestionadas las visitas.	

Historia de Usuario	
<b>Número:</b> 8	<b>Nombre Historia de Usuario:</b> Insertar acciones en la base de datos.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yordan Suarez Santiago	<b>Iteración Asignada:</b> sprint 2
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 3 semana
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1 semana
<b>Descripción:</b> Insertar en una tabla las acciones que un determinado usuario ha estado realizando en el sitio monitorizado.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

Tarea de Ingeniería	
<b>Número Tarea:</b> 9	<b>Número Historia de Usuario:</b> 8
<b>Nombre Tarea:</b> Implementar la clase InsertVisitAction.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 1 semana
<b>Fecha Inicio:</b> 09/04/2010	<b>Fecha Fin:</b> 16/04/2010
<b>Programador Responsable::</b> Yordan Suarez Santiago	
<b>Descripción:</b> Se implementa la clase InsertVisitAction donde se registrará las acciones realizadas por las visitas.	

Historia de Usuario
---------------------

<b>Número:</b> 9	<b>Nombre Historia de Usuario:</b> Chequear si un visitante es recurrente.	
<b>Modificación de Historia de Usuario Número:</b> Ninguna		
<b>Usuario:</b> Yordan Suarez Santiago	<b>Iteración Asignada:</b> sprint 3	
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 3 semana	
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1 semana	
<b>Descripción:</b> Se debe tomar el ID desde los archivos de registros y compararlo con los de la base de datos, si coinciden entonces el visitante en cuestión es recurrente.		
<b>Observaciones:</b>		
<b>Prototipo de interfaz:</b>		

		Tarea de Ingeniería
<b>Número Tarea:</b> 10	<b>Número Historia de Usuario:</b> 9	
<b>Nombre Tarea:</b> Implementar la clase CheckVisitorReturning.		
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 1 semana	
<b>Fecha Inicio:</b> 17/04/2010	<b>Fecha Fin:</b> 23/04/2010	
<b>Programador Responsable::</b> Yordan Suarez Santiago		
<b>Descripción:</b> Se debe implementar la clase CheckVisitorReturning que determinará si un usuario es un visitante recurrente en el sitio monitorizado o no.		

		Historia de Usuario
<b>Número:</b> 10	<b>Nombre Historia de Usuario:</b> Generar respaldo de visitas no	

	completadas.	
<b>Modificación de Historia de Usuario Número:</b> Ninguna		
<b>Usuario:</b> Yordan Suarez Santiago	<b>Iteración Asignada:</b> sprint 3	
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 3 semanas	
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 3 semanas	
<b>Descripción:</b> Se necesita implementar una funcionalidad que complete las visitas en caso de que el sistema haya tomado los archivos de registros antes de finalizada una determinada visita por lo que esta no fue tomada completamente quedando guardado el resto en un fichero de respaldo hasta que se efectúe la otra recogida de los archivos de registros.		
<b>Observaciones:</b>		
<b>Prototipo de interfaz:</b>		

Tarea de Ingeniería	
<b>Número Tarea:</b> 11	<b>Número Historia de Usuario:</b> 10
<b>Nombre Tarea:</b> Implementar la clase BackupVisit.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 2 semana
<b>Fecha Inicio:</b> 24/04/2010	<b>Fecha Fin:</b> 07/05/2010
<b>Programador Responsable::</b> Yordan Suarez Santiago	
<b>Descripción:</b> Se implementa la clase BackupVisit.	

Historia de Usuario	
<b>Número:</b> 11	<b>Nombre Historia de Usuario:</b> Restaurar visitas no completadas desde

	respaldo.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yordan Suarez Santiago	<b>Iteración Asignada:</b> sprint 3
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 3 semana
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1 semana
<b>Descripción:</b> Se necesita implementar una clase que restaure las visitas guardada en un fichero de respaldo.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

Tarea de Ingeniería	
<b>Número Tarea:</b> 12	<b>Número Historia de Usuario:</b> 11
<b>Nombre Tarea:</b> Reutilizar la clase BackupVisit.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 1 semana
<b>Fecha Inicio:</b> 08/05/2010	<b>Fecha Fin:</b> 15/02/2010
<b>Programador Responsable::</b> Yordan Suarez Santiago	
<b>Descripción:</b> Se reutiliza la clase BackupVisit para efectuar la restauración de las visitas.	

## 2.5 Diagrama de componentes

Los diagramas de componente representan la división de un sistema de software en componentes mostrando la dependencia que existe entre ellos. Estos tipos de diagramas pueden ser utilizados para modelar la vista estática y dinámica de un sistema donde se sitúan las librerías, tablas, archivos,

ejecutables y documentos que forman parte del sistema. En la figura 2.1 se muestra el diagrama de componente del módulo de procesamiento para el sistema Airesweb.

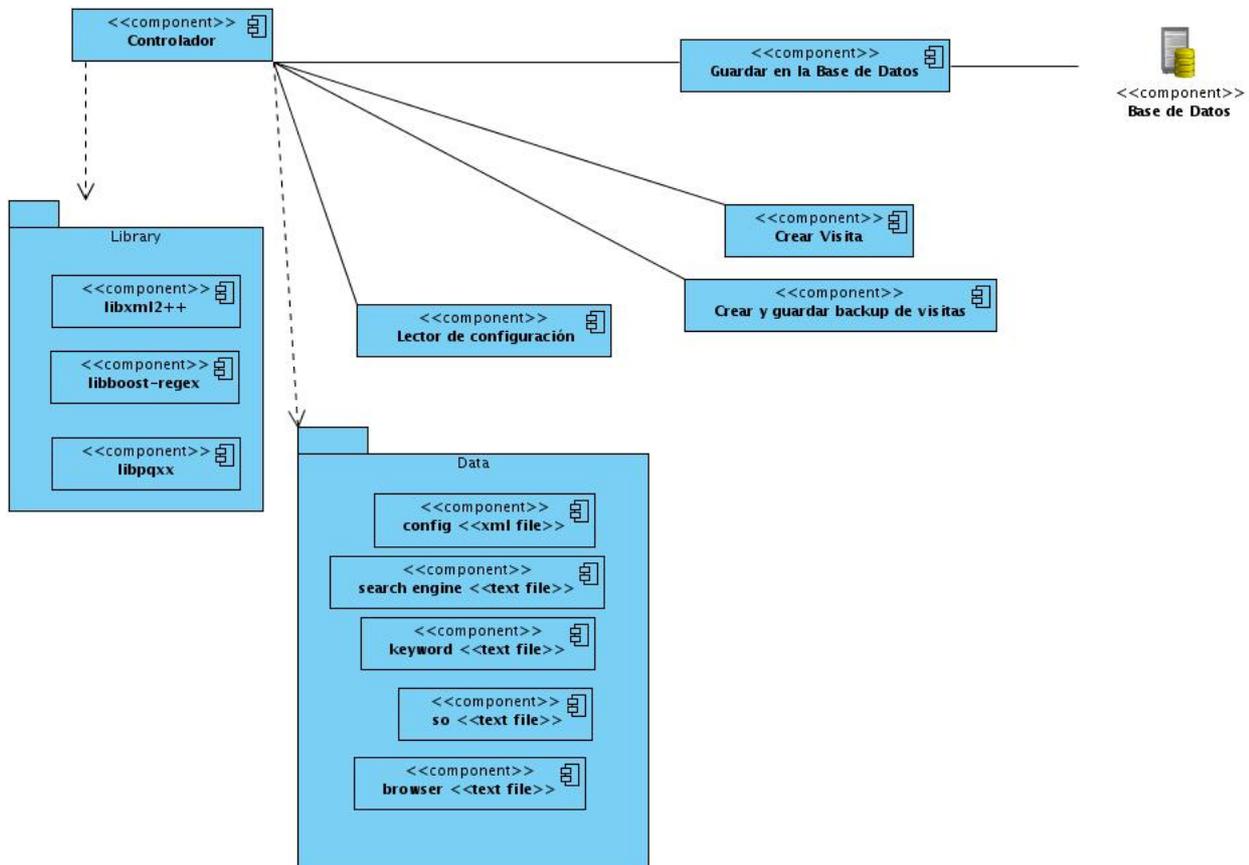


Figura 2.1 Diagrama de componentes del módulo de procesamiento de datos.

Componente	Propósito
Controlador	Es donde se encuentra el código fuente y el encargado de controlar el resultado en el fichero de salida.
Libxml12++	Es la librería para el manejo de archivos XML.
Libboost-regex	Librería para el trabajo con expresiones regulares.

Libpqxx	Librería para la conexión con la Base de Datos.
Lector de configuración	Componente para leer la configuración inicial de la aplicación.
config	Fichero donde se almacena la configuración del módulo.
Search engine	Este fichero contiene las definiciones para cuando una URL pertenece a un buscador.
Keyword	Fichero contenedor de las definiciones para determinar las palabras claves de acuerdo a un buscador específico.
so	Este fichero contiene las definiciones para determinar desde que sistema operativo se hace la visita.
Browser	Este fichero contiene las definiciones de los diferentes navegadores para determinar desde cual se visita el sitio.
Guardar en la Base de Datos	Componente que se encarga de guardar toda la información en la base de datos.
Crear Visita	Componente para determinar las visitas que se realizaron en el sitio.
Crear y guardar respaldo de visitas	Guarda y/o recupera las visitas pendientes a procesar.
Base de datos	Lugar donde se almacenará toda la información luego de ser procesada.

### 2.6 Plan de releases

En la siguiente tabla se muestra el plan de iteraciones para la realización de las entregas que se efectuarán del producto. Para la realización de este plan se tiene en cuenta la prioridad dada a cada historia de usuario por parte del cliente llegando a la siguiente planificación:

Release	Historias de Usuarios	Tiempo estimado(Semanas)
1	1,2,3,4	4
2	5,6,7,8	4
3	9,10,11	3

## 2.7 Clases del módulo

El módulo de procesamiento de datos estará programado en C/C++ y contará con diferentes clases que unidas formarán las funcionalidades del mismo:

**Action:** Clase para almacenar la información referente a una acción en una visita.

**Aires::Application:** Clase que gestiona algunas tareas generales de la aplicación.

**BackupVisit:** Clase para manejar la lectura y/o escritura de ficheros de respaldo para las pendientes a procesar.

**CheckVisitorReturning:** Clase para determinar desde la BD si un usuario es un visitante nuevo o recurrente.

**Config:** Clase para obtener la configuración del sistema desde un fichero XML.

**Date Time:** Clase para manejar el tipo de datos DateTime(Fecha - Hora).

**Directory:** Clase que gestiona los archivos de un directorio

**File:** Clase para la gestión de un archivo

**InsertURL:** Clase para insertar una URL en la Base de Datos y obtener el id asociado a esta URL.

**InsertVisit:** Clase para insertar las estadísticas de una visita, en la Base de Datos.

**InsertVisitAction:** Clase para insertar las acciones referentes a una visita en la Base de Datos.

**Logger:** Clase para gestionar la escritura de mensajes informativos en la consola.

**ParsedLine:** Clase para almacenar los datos extraídos de una línea del archivo de registro.

**Parser:** Clase para separar la información contenida en una línea de los ficheros de registro

## **2.8 Diseño de la base de datos**

Siendo de gran importancia el diseño de la base de datos del módulo procesamiento para el sistema Airesweb se encuentra su descripción detallada en el documento adjunto **BD.pdf**.

## **2.9 Conclusiones**

Luego de abordar en este capítulo la solución propuesta para la implementación del módulo de procesamiento se puede concluir lo siguiente: La descripción de las Historias de Usuarios así como las tareas ingenieriles serán un aspecto fundamental y de gran ayuda en el desarrollo del módulo procesador de datos además se espera que teniendo en cuenta la descripción anterior la implementación sea efectuada de forma eficiente permitiendo un producto final con una buena calidad.

### Capítulo 3. Validación de la solución propuesta

Un producto con la calidad requerida debe contar con un plan de pruebas que abarque su ciclo de vida dándole seguimiento a aquellos pequeños o grandes cambios que ocurren en el desarrollo iterativo de este.

A la hora de realizar las pruebas al módulo de procesamiento se tuvo en cuenta los pilares de la metodología XP, la cual consiste en el uso de test para comprobar el funcionamiento de los códigos que se vayan implementando. Esta metodología divide las pruebas en dos grupos, los cuales se conocen como: pruebas unitarias y pruebas de aceptación.

**Las pruebas unitarias** son realizadas por los programadores y son las encargadas de verificar el código de forma automática.

**Pruebas de Aceptación:** Conocido también como los **casos de PICO** o **Test Case** estas pruebas son un conjunto de condiciones o variables bajo las cuáles el analista determina si el requisito de una aplicación es parcial o completamente satisfactorio. El cliente es el mayor responsable de verificar cada una de las pruebas y de priorizar la corrección de las pruebas que fallan.

Se determinó por parte del equipo de desarrollo documentar la realización de las pruebas al módulo procesador de datos a nivel de aceptación, seguidamente se presentan los casos de pruebas o tests que fueron realizados al sistema en cada una de las iteraciones. Teniendo en cuenta que cada caso de prueba fuera superado consecutivamente se fue avanzando hacia las iteraciones siguientes.

#### 3.1 Casos de pruebas

Cuando se realiza un caso de prueba se busca determinar si un requisito de un sistema o software es parcial o completamente satisfactorio. Con el fin de probar todos los requisitos de una aplicación, debe haber al menos un caso de prueba para cada requisito.

Durante el desarrollo del módulo procesador de datos para Airesweb, se diseñó un conjunto de casos de prueba al que fue sometido el sistema para comprobar el funcionamiento de acuerdo a las Historias de Usuario. Se definieron casos de prueba para todas las Historias de Usuario. A continuación se relacionan las pruebas más significativas realizadas.

**Caso de prueba para la historia de usuario: Establecer conexión con la Base de Datos.**

Esta sección cubre la prueba que se le realizó a la historia de usuario: Establecer conexión con la Base de Datos.

En esta historia de usuario se prueba que se realice la conexión con la base de datos del sistema de forma correcta.

Casos de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> 1	<b>Nombre Historia de Usuario:</b> Establecer conexión con la base de datos
<b>Nombre de la persona que realiza la prueba:</b> Programador(Yordan)	
<b>Descripción de la Prueba:</b> Se ejecuta la prueba y se verifica que se establezca una correcta conexión con la base de datos.	
<b>Condiciones de Ejecución:</b> La aplicación no puede contener errores de ningún tipo en su compilación.	
<b>Entrada / Pasos de ejecución:</b> Se compila la aplicación y se ejecuta.	
<b>Resultado Esperado:</b> Que se realice correctamente la conexión con la base de datos.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Caso de prueba para la historia de usuario: Inicializar parámetros del módulo.**

La prueba que a continuación se expone se le realizó a la historia de usuario: Inicializar parámetros del módulo.

En esta historia de usuario se prueba que se realice la inicialización correcta de los parámetros del módulo.

Casos de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> 1	<b>Nombre Historia de Usuario:</b> Inicializar parámetros del módulo.
<b>Nombre de la persona que realiza la prueba:</b> Programador(Yordan)	
<b>Descripción de la Prueba:</b> Se ejecuta la prueba y se verifica que sean leídos correctamente los parámetros.	
<b>Condiciones de Ejecución:</b> Que la aplicación no tenga errores a la hora de compilar y exista un fichero de configuración.	
<b>Entrada / Pasos de ejecución:</b> Se compila la aplicación y se ejecuta.	
<b>Resultado Esperado:</b> Parámetros del módulo inicializados de forma correcta.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Caso de prueba para la historia de usuario: Explorar directorio de archivos de registros.**

Esta prueba se le realizó a la historia de usuario explorar directorio de archivos de registros y se intenta probar que exista el directorio con una correcta lectura de los archivos.

Casos de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> 1	<b>Nombre Historia de Usuario:</b> Explorar directorio de archivos de registros.
<b>Nombre de la persona que realiza la prueba:</b> Programador(Dunia)	
<b>Descripción de la Prueba:</b> Se ejecuta la prueba y se verifica que se lean los archivos de registros correctamente.	
<b>Condiciones de Ejecución:</b> Que la aplicación no tenga errores a la hora de compilar y exista el directorio donde se almacenan los archivos de registros.	

<b>Entrada / Pasos de ejecución:</b> Se compila la aplicación y se ejecuta.
<b>Resultado Esperado:</b> Leer archivos de registro satisfactoriamente.
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.

**Caso de prueba para la historia de usuario: Analizar archivo de registros.**

Prueba realizada a la historia de usuario: Analizar fichero de registro.

En esta historia de usuario se prueba que se realice una correcta extracción de la información desde los archivos de registros.

Casos de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> 1	<b>Nombre Historia de Usuario:</b> Analizar fichero de registro.
<b>Nombre de la persona que realiza la prueba:</b> Programador(Yordan)	
<b>Descripción de la Prueba:</b> Se ejecuta la prueba y se comprueba la extracción correcta de la información desde el archivo de registro.	
<b>Condiciones de Ejecución:</b> Que la aplicación no tenga errores a la hora de compilar y exista el archivo de registro.	
<b>Entrada / Pasos de ejecución:</b> Se compila la aplicación y se ejecuta.	
<b>Resultado Esperado:</b> Información extraída de los archivos de registro correctamente.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Caso de prueba para la historia de usuario: Insertar URL y obtener id.**

Prueba que se le realizó a la historia de usuario insertar URL y obtener id con el fin de verificar que una vez ejecutada la aplicación se inserten de forma adecuada la URL en la base de datos del sistema y se obtenga la identificación del mismo.

Casos de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> 1	<b>Nombre Historia de Usuario:</b> Insertar URL y obtener id.
<b>Nombre de la persona que realiza la prueba:</b> Programador(Dunia)	
<b>Descripción de la Prueba:</b> Se ejecuta la prueba y se verifica que el URL esté insertado y el id obtenido sea el correspondiente en la base de datos.	
<b>Condiciones de Ejecución:</b> Que la aplicación no tenga errores a la hora de compilar, exista conexión correcta con la base de datos.	
<b>Entrada / Pasos de ejecución:</b> Se compila la aplicación y se ejecuta.	
<b>Resultado Esperado:</b> URL insertado en la base de datos y el id obtenido sea el correcto.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

#### Caso de prueba para la historia de usuario: Estimar visita.

Escenario dedicado a probar la funcionalidad: Estimar visita examinando que dicha estimación se efectúe correctamente.

Casos de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> 1	<b>Nombre Historia de Usuario:</b> Estimar visita.
<b>Nombre de la persona que realiza la prueba:</b> Programador(Yordan)	
<b>Descripción de la Prueba:</b> Se ejecuta la prueba y se verifica que las visitas obtenidas sean estimadas correctamente.	
<b>Condiciones de Ejecución:</b> Exista un archivo de registros.	
<b>Entrada / Pasos de ejecución:</b> Se compila la aplicación y se ejecuta.	
<b>Resultado Esperado:</b> Estimación correcta de las visitas.	

<b>Evaluación de la Prueba:</b> Prueba satisfactoria.
---

**Caso de prueba para la historia de usuario: Insertar visita en la tabla.**

Esta sección cubre la prueba que se le realizó a la Historia de Usuario: Insertar visita en la tabla base de datos.

En esta historia de usuario se prueba que se inserte las visitas en la base de datos de forma correcta.

Casos de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> 1	<b>Nombre Historia de Usuario:</b> Insertar visita en la base de datos.
<b>Nombre de la persona que realiza la prueba:</b> Programador(Yordan)	
<b>Descripción de la Prueba:</b> Se ejecuta la prueba y se verifica que las visitas sean insertadas correctamente en la base de datos del sistema.	
<b>Condiciones de Ejecución:</b> Que la aplicación no tenga errores a la hora de compilar, exista conexión correcta con la base de datos.	
<b>Entrada / Pasos de ejecución:</b> Se compila la aplicación y se ejecuta.	
<b>Resultado Esperado:</b> Visitas insertadas correctamente en la base de datos del sistema.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Caso de prueba para la historia de usuario: Insertar acciones en la base de datos.**

Esta sección cubre el conjunto de pruebas que se le realizaron a la Historia de Usuario: Insertar acciones en la base de datos.

En esta Historia de Usuario se prueba que se inserte las acciones realizadas por una visita de forma correcta.

Casos de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> 1	<b>Nombre Historia de Usuario:</b> Insertar acciones en la base de datos.
<b>Nombre de la persona que realiza la prueba:</b> Programador(Yordan)	
<b>Descripción de la Prueba:</b> Se ejecuta la prueba y se verifica que las acciones de las visitas sean insertadas correctamente en la base de datos del sistema.	
<b>Condiciones de Ejecución:</b> Que la aplicación no tenga errores a la hora de compilar, exista conexión correcta con la base de datos.	
<b>Entrada / Pasos de ejecución:</b> Se compila la aplicación y se ejecuta.	
<b>Resultado Esperado:</b> Acciones de las visitas insertadas correctamente en la base de datos del sistema.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Caso de prueba para la historia de usuario: Chequear si un visitante es recurrente.**

Esta prueba se enfoca en examinar si el sistema es capaz de determinar si un visitante es recurrente o no en un sitio monitorizado.

Casos de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> 1	<b>Nombre Historia de Usuario:</b> Chequear si un visitante es recurrente.
<b>Nombre de la persona que realiza la prueba:</b> Programador(Dunia)	
<b>Descripción de la Prueba:</b> Se ejecuta la prueba y se verifica si un usuario es recurrente en un sitio específico.	
<b>Condiciones de Ejecución:</b> Que la aplicación no tenga errores a la hora de compilar, exista conexión correcta con la base de datos.	

<b>Entrada / Pasos de ejecución:</b> Se compila la aplicación y se ejecuta.
<b>Resultado Esperado:</b> Que el resultado obtenido de la base datos sea correcto.
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.

**Caso de prueba para la historia de usuario: Generar archivo de respaldo de visitas no completadas.**

Esta sección cubre el conjunto de pruebas que se le realizaron a la historia de usuario: Generar archivos de respaldo de visitas no completadas.

En esta historia de usuario se prueba que se genere en su formato correcto y con la información adecuada los archivos de respaldo.

Casos de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> 1	<b>Nombre Historia de Usuario:</b> Generar archivos de respaldo de visitas no completadas.
<b>Nombre de la persona que realiza la prueba:</b> Programador(Yordan)	
<b>Descripción de la Prueba:</b> Se ejecuta la prueba y se verifica que los archivos de respaldo sean generados con el formato y la información correcta.	
<b>Condiciones de Ejecución:</b> Que la aplicación no tenga errores a la hora de compilar y existan visitas no completadas.	
<b>Entrada / Pasos de ejecución:</b> Se compila la aplicación y se ejecuta.	
<b>Resultado Esperado:</b> Un archivo de respaldo generado con formato e información correcto.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Caso de prueba para la historia de usuario: Restaurar visitas no completadas desde archivos de respaldo.**

Prueba que se le realizó a la historia de usuario: Restaurar visitas no completadas desde archivos de respaldo.

En esta historia de usuario se prueba que se genere en su formato correcto y con la información adecuada los archivos de respaldo.

Casos de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> 1	<b>Nombre Historia de Usuario:</b> Restaurar visitas no completadas desde archivos de respaldo.
<b>Nombre de la persona que realiza la prueba:</b> Programador(Dunia)	
<b>Descripción de la Prueba:</b> Se ejecuta la prueba y se verifica que los archivos de respaldo sean leídos correctamente.	
<b>Condiciones de Ejecución:</b> Que la aplicación no tenga errores a la hora de compilar y exista un archivo de respaldo correspondiente al sitio que se está analizando.	
<b>Entrada / Pasos de ejecución:</b> Se compila la aplicación y se ejecuta.	
<b>Resultado Esperado:</b> Una lista de visitas no completadas.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

### 3.2 Resultados obtenidos

La relación de los resultados obtenidos por parte del equipo de desarrollo serán tratados en este epígrafe. El producto realizado, el módulo procesador de datos para Airesweb, será incluido en el sistema Airesweb como parte de las mejoras propuestas en el mismo para su versión 1.0.

#### 3.2.1 Tiempo de desarrollo

El producto Airesweb en su versión Alfa destacaba dentro de sus características su implementación utilizando como lenguaje de programación perl lo que provocaba lentitud en el traslado de la información

desde los registros de archivos hacia la base de datos del sistema, es aquí donde se palpa la necesidad de realizar un módulo que se encargara de agilizar dicho traslado.

A finales de junio del 2009 se comienzan los primeros estudios con el fin de ir obteniendo los conocimientos necesarios para la implementación del módulo. El equipo de trabajo estuvo conformado por dos miembros, entre ambos se jugaron los roles de programador, diseñador, analista y encargado de pruebas logrando un producto que cumple con las funcionalidades previamente establecidas, además se encuentra disponible para ser utilizado en la futura liberación del software Airesweb versión 1.0.

En el ciclo de vida del módulo, aunque se considera que fue de corta duración, se cumplieron con todos los requisitos establecidos una vez realizada la lista de reserva del producto.

### **3.3 Conclusiones**

Una vez culminado el capítulo donde se presentaron los casos de pruebas y se realizó un breve análisis de los resultados obtenidos, se concluye que: La realización de los casos de prueba guiaron el desarrollo del módulo implementado obteniendo los resultados esperados, las pruebas realizadas permitieron aumentar la calidad final del producto.

**Conclusiones.**

Con la culminación del presente trabajo de diploma y después de estudiar varias soluciones existentes a nivel internacional se puede concluir que:

- El estudio realizado a profundidad en la historia del arte de la analítica web permitió seleccionar las herramientas, tecnologías y la metodología adecuadas para guiar el ciclo de vida del producto.
- La decisión de utilizar Scrum - XP como metodología de desarrollo permitió la existencia de una hegemonía en el equipo de trabajo evitando desacuerdos innecesarios, además la realización de las pruebas incrementó la calidad del producto.
- Se implementó un módulo capaz de gestionar el traslado de la información desde los archivos de registros hacia la base de datos del sistema Airesweb cumpliendo con un 100% los requisitos funcionales establecido.

Se plantea como conclusión universal el cumplimiento del objetivo general planteado al inicio de dicho trabajo el cual consistió en desarrollar un módulo de procesamiento de datos para el sistema de análisis Airesweb que gestionara de forma óptima el traslado de la información desde los archivos de registros hacia la base de datos del sistema.

### **Recomendaciones.**

Después de la investigación realizada y el desarrollo del módulo de procesamiento de datos se recomienda:

- Continuar el perfeccionamiento del módulo de procesamiento para una mayor fiabilidad.
- Estudiar otras posibles características las cuales se pudieran agregar al módulo para lograr resultados superiores.
- Incluir dentro de las futuras versiones del sistema Airesweb el módulo procesador de datos con el fin de probar y darle seguimiento a las utilidades del mismo.
- Aprovechar la arquitectura utilizada en la programación del módulo para adicionar algoritmos de programación en paralelo y distribuida, teniendo en cuenta que el módulo de procesamiento de datos esté optimizado y que su procesamiento sea más rápido y eficiente y que se puedan explotar todas las características del hardware.

### Referencias bibliográficas

- [1] Expertos en Internet y Experiencia Eclesial, "F.A.Q - VE Multimedios :: Expertos en Internet y Experiencia Eclesial," 2010. Disponible en <http://www.ve-multimedios.org/espanol/ve-multimedios/f.a.q.html>.
- [2] Gabriel Marí i Domènec, "La toma de decisiones y la organización del trabajo en la empresa: reflexiones en torno a ellas - Taller Digital de la Universidad de Alicante. Disponible en <http://www.eltallerdigital.com/informacion.jsp?idArticulo=11>"
- [3] Analog, "Analog: WWW logfile analysis," Jun. 2005 Disponible en <http://www.analog.cx/>.
- [4] AWStats, "AWStats - Free log file analyzer for advanced statistics (GNU GPL).," Dic. 2006. Disponible en <http://awstats.sourceforge.net/>.
- [5] Webtrends, "Webtrends. Enterprise Customer Intelligence," 2010. Disponible en <http://www.webtrends.com/>.
- [6] Webalizer, "Home of The Webalizer," Nov. 2009. Disponible en <http://www.webalizer.com/>.
- [7] WebLog Expert, "WebLog Expert - Powerful log analyzer," 2010. Disponible en <http://www.weblogexpert.com/>.
- [8] "DE\_91," *Departamentos de Sistemas y Computación*, 2009. Disponible en [http://sistemas.itlp.edu.mx/tutoriales/algoritmos/tema\\_91.htm](http://sistemas.itlp.edu.mx/tutoriales/algoritmos/tema_91.htm).
- [9] Ídem [8]
- [10] "Software Modules," 2010. Disponible en <http://www.comptechdoc.org/independent/programming/programming-standards/software-modules.html>.
- [11] Ídem [10]
- [12] Ídem [8]
- [13] Ídem [8]
- [14] Real Academia, "Real Academia Española," 2010. Disponible en <http://www.rae.es/RAE/Noticias.nsf/Home?ReadForm>.
- [15] Departamento de Computación, "Departamento de Computación," 2005. Disponible en <http://dc.exa.unrc.edu.ar/.../El%20lenguaje%20de%20programacion%20java.pdf>.
- [16] Francisco Cortijo Bon, "Curso de C# - Introducción al lenguaje de programación C#," 2010. Disponible en <http://elvex.ugr.es/decsai/csharp/language/intro.xml>.
- [17] J. Carlos López Ardao, *Manual De Programacion Lenguaje C | Scribd*, 2001. Disponible en <http://www.scribd.com/doc/22283/Manual-De-Programacion-Lenguaje-C>.

- [18] Ídem [17]
- [19] Ídem [17]
- [20] Universidad de Jaén, “tema7.pdf (application/pdf Objeto),” *Gestores de Base de Datos*, 2010. <http://www.di.ujaen.es/~barranco/publico/ofimatica/tema7.pdf>
- [21] Universidad Autónoma del Estado de Morelos, *MySQL.pdf (application/pdf Objeto)*, 2009. <http://www.uaem.mx/posgrado/mcruz/cursos/miic/MySQL.pdf>
- [22] Ídem [21]
- [23] Ídem [21]
- [24] Ídem [21]
- [25] PostgreSQL, “PostgreSQL: The world's most advanced open source database,” 1996. Disponible en <http://www.postgresql.org/>.
- [26] Grady Booch, Jim Rumbaugh e Ivar Jacobson, *Lenguaje unificado modelado booch* - Disponible en <http://www.librospdf.net/lenguaje-unificado-modelado-booch/1/>
- [27] XP, “Extreme Programming: A Gentle Introduction.”, 2009. Disponible en <http://www.extremeprogramming.org/>
- [28] Juan Palacio, *Flexibilidad con Scrum*, 2008. Disponible en <http://www.postgresql.org/>.
- [29] Martínez Alejandro y Martínez Raúl. Guía a Rational Unified Process. 2009. Disponible en <http://www.info-ab.uclm.es/asignaturas/>.
- [30] Alberto Molpeceres, “Proceso de desarrollo: RUP, XP y FDD,” 2002. Disponible en <http://www.willydev.net/descargas/Articulos/General/cualxpfdrrup.PDF>
- [31] Eclipse, “Eclipse.org home,” 2010. Disponible en <http://www.eclipse.org/>.
- [32] NetBeans, “Welcome to NetBeans,” 2010. Disponible en <http://www.eclipse.org/>.
- [33] Visual Paradigm, “Visual Paradigm for UML,” 2010. Disponible en [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_%28M%C3%8D%29\\_14720\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/)

### Bibliografía

- deGerencia.com. Toma de decisiones:: deGerencia.com. 2001. [cited 25 Mayo 2010]. Disponible en: <[http://www.degerencia.com/tema/toma\\_de\\_decisiones](http://www.degerencia.com/tema/toma_de_decisiones)>.
- CIPRE. Acerca del CIP — Sitio Web del Centro de Informacion para la Prensa. 2007. [cited 24 Mayo 2010]. Disponible en: <[http://www.cip.cu/contenido/menu1/acerca\\_cip](http://www.cip.cu/contenido/menu1/acerca_cip)>.
- Expertos en Internet y Experiencia Eclesial. F.A.Q - VE Multimedios :: Expertos en Internet y Experiencia Eclesial. 2010. [cited 24 Mayo 2010]. Disponible en: <<http://www.vemultimedios.org/espanol/ve-multimedios/f.a.q.html>>.
- Analítica Web. Analitica Web | MV Consultoria. 2010. [cited 25 Mayo 2010]. Available from Disponible en: <<http://www.analiticaweb.es/>>.
- Gabriel Marí i Domènec. La toma de decisiones y la organización del trabajo en la empresa: reflexiones en torno a ellas - Taller Digital de la Universidad de Alicante. [cited 24 Mayo 2010]. Disponible en: <<http://www.eltalldigital.com/informacion.jsp?idArticulo=11>>.
- Milenium. Páginas Web, Internet, Sitio Web, Posicionamiento, Definicion. 2009. [cited 25 Mayo 2010]. Disponible en <<http://www.informaticamilenium.com.mx/paginas/espanol/sitioweb.htm>>.
- Real Academia. Real Academia Española. 2010. [cited 24 Mayo 2010]. Disponible en: <<http://www.rae.es/RAE/Noticias.nsf/Home?ReadForm>>.
- Ricardo Tayar. El valor de la analítica web para los no habituados a Internet | Blog de Marketing Online de Tristán Elósegui. Febrero 2010. [cited 24 Mayo 2010]. Disponible en: <<http://tristanelosegui.com/2010/02/23/el-valor-de-la-analitica-web-para-los-no-habituados-a-internet/>>.
- Angel Rodríguez Alvarez. La informática en Cuba tiene su historia [Voltaire]. Junio 2006. [cited 25 Mayo 2010]. Disponible en: <<http://www.voltairenet.org/article141015.html>>.

- Simple Machines LLC. LENGUAJES DE PROGRAMACION: Generalidades. 2006. [cited 25 Mayo 2010]. Disponible en: <<http://www.programacionutn.com.ar/foro/c-c/lenguajes-de-programacion-generalidades/>>.
- Jacobson, I., Booch, G., Rumbaugh J. *El Proceso Unificado de Desarrollo de Software*. 2000 [cited 25 Mayo 2010]. Disponible en: <<http://www.librospdf.net/%E2%80%9CEI-Proceso-Unificado-de-Desarrollo-de-Software%E2%80%9D.-/1/>>.
- Wikilearning. Introducción a las Bases de Datos - Tipos de Gestores de Bases de Datos. 2006. [cited 25 Mayo 2010]. Disponible en: <[http://www.wikilearning.com/tutorial/introduccion\\_a\\_las\\_bases\\_de\\_datos-tipos\\_de\\_gestores\\_de\\_bases\\_de\\_datos/10147-2](http://www.wikilearning.com/tutorial/introduccion_a_las_bases_de_datos-tipos_de_gestores_de_bases_de_datos/10147-2)>.
- DIRS. El Lenguaje de Modelado Unificado (UML). 2010. [cited 25 Mayo 2010]. Disponible en: <<http://www.docirs.cl/uml.htm>>.
- José Luis Sánchez Navarro. Conocimientos básicos para desarrollar software. 2003. [cited 25 Mayo 2010]. Disponible en: <<http://www.avemundi.com/archivos/000007.html>>.
- Eduardo Rivera Alva. Arquitectura de Software II - Diagrama de Componentes y Despliegue. 2008. [cited 25 Mayo 2010]. Disponible en <<http://www.scribd.com/doc/7884665/Arquitectura-de-Software-II-Diagrama-de-Componentes-y-Despliegue>>.
- Sparx Systems Pty Ltd. . Sparx Systems - Tutorial UML 2 - Diagrama de Componentes. 2007. [cited 25 Mayo 2010]. Disponible en: <[http://www.sparxsystems.com.ar/resources/tutorial/uml2\\_componentdiagram.html](http://www.sparxsystems.com.ar/resources/tutorial/uml2_componentdiagram.html)>.
- Microsoft Corporation. Trabajar con pruebas unitarias. 2001. [cited 25 Mayo 2010]. Disponible en: <[http://msdn.microsoft.com/es-es/library/ms182515\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/ms182515(VS.80).aspx)>.
- Gerardo Fernández Escribano. Presentacion-XP.pdf (application/pdf Objeto). *Ingeniería del Software II* 2002. [cited 25 Mayo 2010]. Disponible en: <<http://www.info-ab.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion-XP.pdf>>.
- J. J. Gutiérrez, M. J. Escalona, M. Mejías, J. Torres. PRUEBAS DEL SISTEMA EN PROGRAMACIÓN EXTREMA.2010. [cited 25 Mayo 2010]. Disponible en <[http://www.lsi.us.es/~javierj/investigacion\\_ficheros/PSISEXTREMA.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/PSISEXTREMA.pdf)>.

**Anexos.**

**Anexo 1: Interfaz Web de Analog.**

## Browser Summary

(Go To: [Top](#) | [General Summary](#) | [Daily Report](#) | [Daily Summary](#) | [Hourly Summary](#) | [Search Word Report](#) | [Browser Summary](#) | [Operating System Report](#) | [Status Code Report](#) | [File Size Report](#) | [File Type Report](#) | [Internal Search Query Report](#) | [Request Report](#))

*This report lists the vendors of visitors' browsers.*

The wedges are plotted by the number of requests for pages.

Listing the top 20 browsers by the number of requests for pages, sorted by the number of requests for pages.

no.	reqs	pages	browser
1	53274	7045	Firefox
	52108	6921	Firefox/3
	1109	110	Firefox/2
	42	10	Firefox/1
	5	3	Firefox/78990603
2	30201	4081	MSIE
	14267	1671	MSIE/8
	8427	1188	MSIE/7
	7034	999	MSIE/6
	318	129	MSIE/5
	107	79	MSIE/4
	37	14	MSIE/999
	7	1	MSIE/3
3	28245	2487	Netscape (compatible)

Anexo 2: Interfaz Web de AWStats

**Statistics for:**  
destailleur.fr

[Visitor counter](#)  
tel je aantal bezoekers count your visitors  
www.orakel.com

[Data is King](#)  
What really happens on your site? Real-time website statistics  
www.opentracker.net

[Web Analytics](#)  
Accurate, adaptable, actionable information.  
www.site-intelligence.co.uk

Ads by Google

**Last Update:** 24 Feb 2010 - 06:51

**Reported period:** Feb 2010 OK

**Summary**

<b>Reported period</b>	Month Feb 2010				
<b>First visit</b>	01 Feb 2010 - 00:18				
<b>Last visit</b>	24 Feb 2010 - 05:36				
	Unique visitors	Number of visits	Pages	Hits	Bandwidth
Viewed traffic *	678	918 (1.35 visits/visitor)	1771 (1.92 Pages/Visit)	9606 (10.46 Hits/Visit)	28.63 MB (31.93 KB/Visit)
Not viewed traffic *			9759	18494	39.57 MB

\* Not viewed traffic includes traffic generated by robots, worms, or replies with special HTTP status codes.

**Monthly history**



Month	Unique visitors	Number of visits	Pages	Hits	Bandwidth
Jan 2010	899	1171	2317	12550	36.95 MB
<b>Feb 2010</b>	678	918	1771	9606	28.63 MB
Mar 2010	0	0	0	0	0
Apr 2010	0	0	0	0	0
May 2010	0	0	0	0	0
Jun 2010	0	0	0	0	0
Jul 2010	0	0	0	0	0
Aug 2010	0	0	0	0	0
Sep 2010	0	0	0	0	0
Oct 2010	0	0	0	0	0
Nov 2010	0	0	0	0	0
Dec 2010	0	0	0	0	0

**Navigation:**  
Visits duration  
File type  
Viewed  
 Full list  
 Entry  
 Exit

**Operating Systems:**  
 Versions  
 Unknown

**Browsers:**  
 Versions  
 Unknown

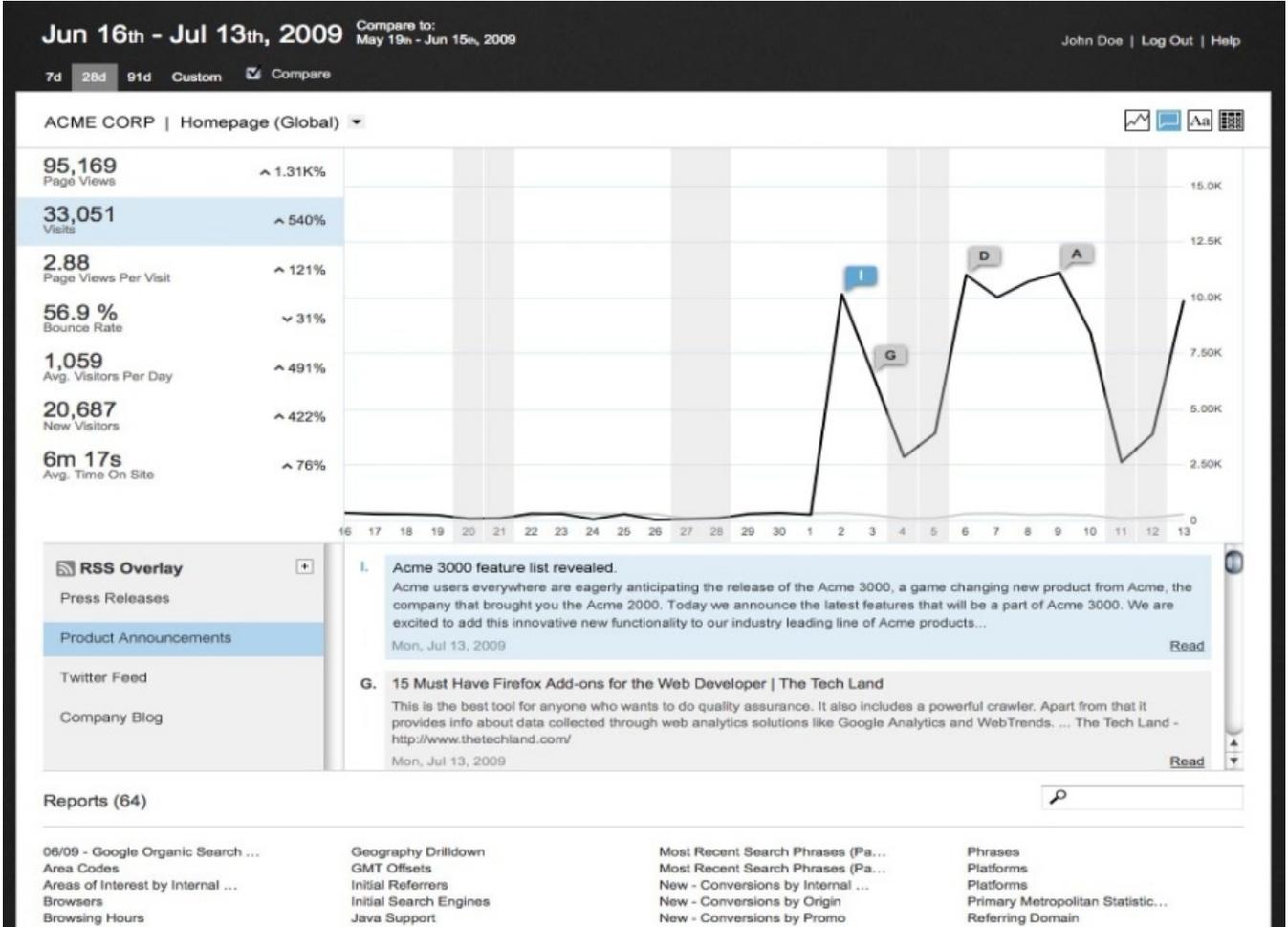
**Screen sizes:**

**Referrers:**  
Origin  
 Referring search engines  
 Referring sites

**Search:**  
 Search Keyphrases  
 Search Keywords

**Others:**  
Web compression  
Miscellaneous  
HTTP Status codes  
 Pages not found

Anexo 3: Interfaz Web de Webtrends.



Anexo 4: Interfaz Web de Webalizer.



WEB HOSTING

LOG OUT  
NEXTGEN CP Build: 20070306 0.1 beta  
LOADING statistics...

-  **Show main menu**  
Pandela Hosting Control
-  **Upgrade**  
Upgrade your account
-  **Script installer**  
Install scripts text
-  **Link to us**  
Support this service
-  **Invite a friend**  
Invite your friends
-  **Bug report**  
Report bug in the system

Storage 0B / 20.0GB  
0%

Traffic 0B / 200.0GB  
0%

MySQL Quota 0B / 100MB  
0%

This usage statistics are based on a monthly traffic and disk space usage of your account.



**Monthly Statistics for April 2007,** [server3.pandela.com](#)

Stats Summary
Daily Statistics
Hourly Statistics
Top Entry/Exit
Top Referrers
Top Countries



■ Pages   
 ■ Files   
 ■ Hits

Close
Need Help? 

**You are logged as:** demouser

Server: [server3.pandela.com](#)

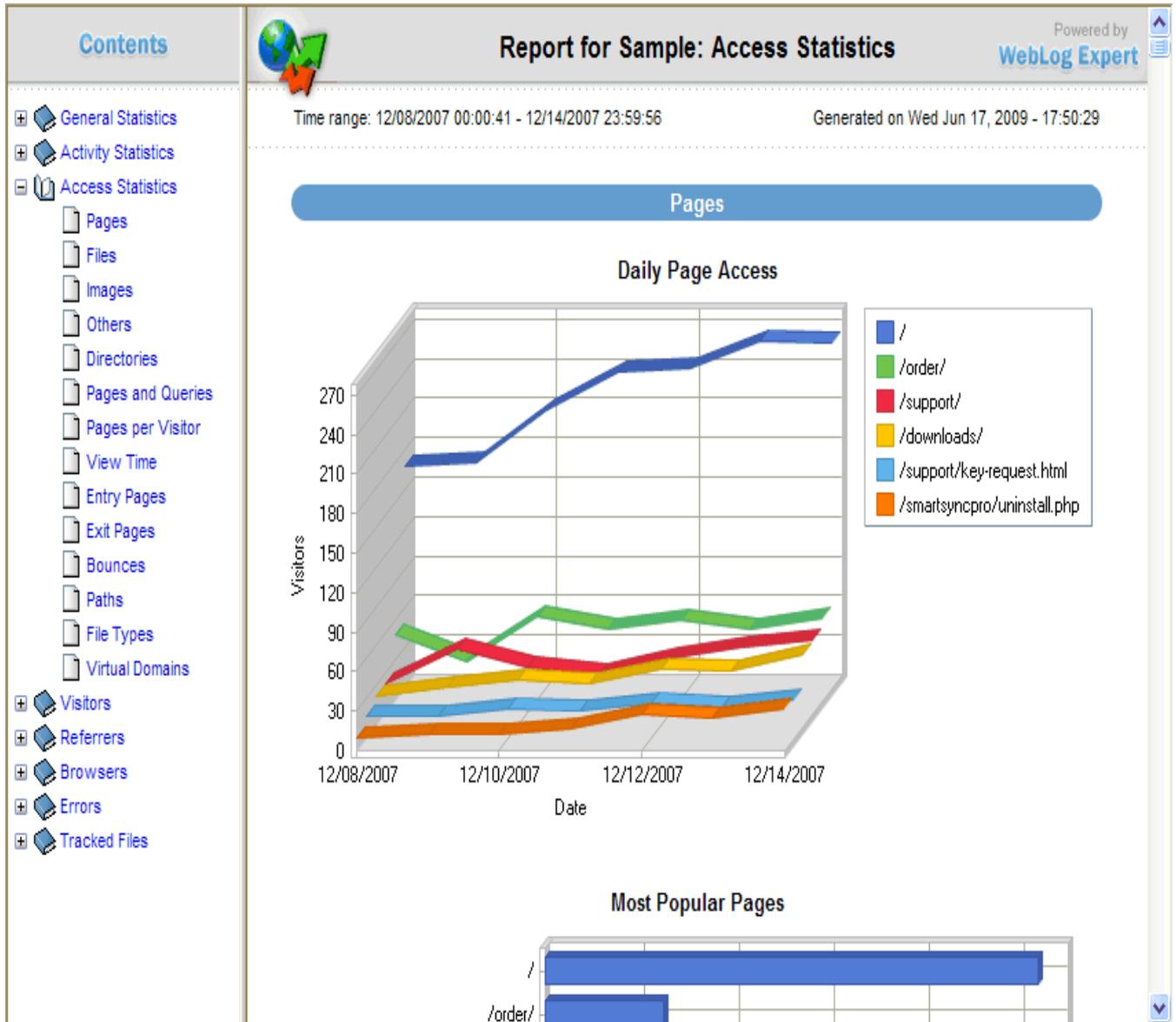
MySQL Server: [sql3.pandela.com](#)

<b>Domains:</b>	<b>1/3</b>	<b>Mailboxes:</b>	<b>1/1000</b>
<b>Subdomains:</b>	<b>1/100</b>	<b>Aliases:</b>	<b>2/1000</b>
<b>MySQL DBs:</b>	<b>2/20</b>	<b>FTP:</b>	<b>1/100</b>

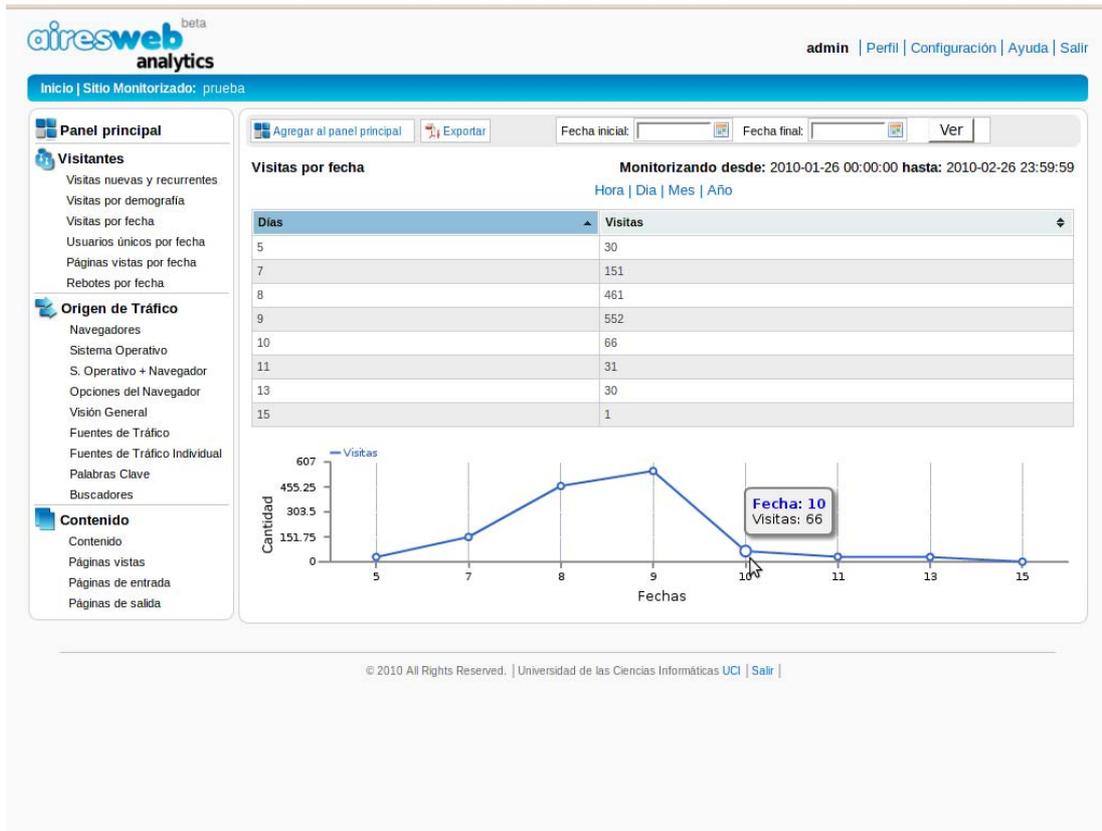
POWERED BY



Anexo 5: Interfaz Web de Web log expert.



### Anexo 6: Interfaz Web de Airesweb.



**Glosario de términos.**

**UCI:** Universidad de las Ciencias Informáticas creada en el 2002 por el comandante Fidel Castro como parte de la batalla de ideas.

**CIPRE:** Centro de Información para la Prensa.

**Airesweb:** Es un software que apoya la toma de decisiones por parte de empresas o entidades permitiendo conocer el funcionamiento de un sitio monitorizado en internet.

**Web:** La traducción literal de esta palabra inglesa es tela de araña, pero en términos informáticos significa mucho más que eso.

**UML:** Lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software.

**XP:** Programación Extrema.

**RUP:** Proceso Unificado de Desarrollo.

**RUP:** Metodología de desarrollo de software creada por la Corporación Rational.

**IDE:** Entorno de desarrollo integrado. Herramienta que se usa para facilitar el desarrollo de software.

**BD:** Base de Datos.

**HU:** Historia de Usuarios.

**POO:** Programación Orientada a Objeto.