

Website MA: Herramienta para Automatizar Métricas Web

Guillermo Lafuente^{1,2}, Luis Olsina¹

¹Grupo de I+D en Ingeniería de Software (GIDIS),
Departamento de Informática, Facultad de Ingeniería, UNLPam.
Calle 110 esq. 9, 6360 General Pico, La Pampa

²Departamento de Sistemas, UNLu.

Rutas 5 y 7, Luján, Buenos Aires

E-mail [\[lafuente,olsinal\]@ing.unlpam.edu.ar](mailto:[lafuente,olsinal]@ing.unlpam.edu.ar)

Resumen. El presente paper analiza detalles de diseño e implementación de la herramienta Website MA (Metric Automation), la cual permite automatizar algunos pasos del proceso de evaluación de la metodología Web-site QEM (Quality Evaluation Methodology). Particularmente, permite la recolección de datos automática a partir de un URL de un sitio o aplicación Web; la obtención de métricas Web, y la posterior generación de informes textuales y gráficos de las mismas. Por lo tanto, se analizará los principales módulos funcionales de la herramienta y se describirá el conjunto de métricas Web automatizables. Las métricas obtenidas sirven de entrada al proceso de evaluación elemental y global de la metodología Web-site QEM.

Palabras Claves. Métricas Web, Atributos, Calidad, Web-site QEM, Website MA.

1. Introducción.

Actualmente el incesante crecimiento de las tecnologías y aplicaciones Web genera interesantes desafíos a los desarrolladores y evaluadores desde varios puntos de vista. Por ejemplo, desde la perspectiva del aseguramiento de la calidad de los productos Web, es preciso definir requerimientos funcionales y no funcionales de manera de poder medir, analizar, comprender, controlar y, potencialmente, mejorar la calidad producida. Cabe destacar que los sitios Web ganaron en interacción y funcionalidad para pasar de ser sólo una forma de presentar información y contenido, a ser aplicaciones con soporte a complejidad de software tradicional. Por lo tanto, la presentación e información del contenido Web y funcionalidad percibida por una audiencia general, surge de la combinación y uso racional de varios tipos de medios y tecnologías que van desde el simple texto, imágenes, animaciones, videos y sonidos, hasta interacciones complejas soportadas por scripts, applets, componentes ActiveX, entre otras. En consecuencia, un empleo no sistemático, ad hoc de estos recursos puede provocar muchos problemas para el usuario de la Web, que sólo intenta encontrar información, navegar, acceder a contenidos y, en definitiva, a funcionalidad específica del dominio de acuerdo a sus necesidades.

Así, la falta de una clara estructura organizativa del sitio (un pobre agrupamiento de las opciones principales y navegacionales, la ausencia de tabla de contenidos, mapa del sitio o índices), la falta de contenidos relevantes, la baja velocidad en cargar las páginas, o la existencia de enlaces rotos o inválidos, entre otros aspectos, pueden conducir a que el usuario abandone el sitio Web, con una percepción de baja calidad y hasta posiblemente con una imagen deslucida de la institución visitada. Por ejemplo, en el dominio de sitios de comercio electrónico, entre atributos deseables del producto que se ofrece en el catálogo, están el proveer una clara descripción del mismo, una imagen (o varias vistas, posiblemente con capacidad de ampliación), la disponibilidad del producto, el

precio, etc. Pero además es preciso contar con atributos obligatorios como transacción segura en la compra, en el acceso a la cuenta del cliente, u otros opcionales como mecanismos de compra rápida, redirección de compras (p.ej., gift service), entre otros.

Por una parte, como se podría observar, es difícil considerar todas esas características y atributos si no se cuenta con un esquema o modelo de calidad que nos permita especificar ordenadamente dichas características y atributos (que podrían superar las ciento cuarenta [12]). El modelo de calidad, en la forma de un árbol de requerimientos, se debe tener principalmente presente a la hora de diseñar requerimientos para una ulterior evaluación y aseguramiento de la calidad. Por otra parte, contar con una metodología que facilite los procesos de especificación, diseño e implementación de la evaluación de sitios y aplicaciones Web es un recurso importante en el proceso de aseguramiento de calidad de proyectos Web. Para tal fin, la metodología Web-Site QEM [11], define un enfoque integral, sistemático y cuantitativo para evaluar y comparar productos Web, tanto en la fase operativa como en la fase de desarrollo del ciclo de vida Web. Se han realizado casos de estudio en diversos dominios de proyectos operativos, como en museos [9], sitios académicos [10] y de comercio electrónico [12]. Además, se ha empleado la metodología en el aseguramiento de calidad en proyectos de desarrollo Web.

En este contexto, para agilizar el proceso de evaluación, es necesario contar con herramientas que ayuden a automatizar algunas de las actividades de esta metodología. Por lo tanto, el objetivo principal de este trabajo consiste en describir claramente cuáles son las métricas Web automatizables (de las que han intervenido en los estudios antes citados) como así también, analizar los principales módulos funcionales de la herramienta Website MA [6]. Los valores de las métricas automatizadas sirven de entrada al proceso de evaluación elemental y global de la metodología Web-site QEM.

El presente trabajo se estructura de la siguiente manera: en la sección 2, se comenta la motivación que nos llevó a desarrollar Website MA en consideración de las que existían en el mercado. A continuación, en la sección 3, se describen distintos atributos que son automatizables desde el punto de vista de la recolección de datos y obtención del valor de las métricas, y que han participado directa o indirectamente en los estudios arriba citados. En la sección 4, se analiza los principales módulos funcionales de la herramienta Website MA, considerando aspectos de diseño e implementación. Por último, se realizan algunas consideraciones finales y se delimitan futuros avances.

2. Motivación.

Como ha indicado Pfleeger *"metrics are welcome when they are clearly needed and easy to collect and understand"* [13]. El agregado de soporte automático al proceso de recolección de datos tiene varias ventajas, que van desde la reducción de esfuerzo y costos hasta la minimización de errores e imprecisiones debido a la intervención humana.

Si bien al momento de realizar los casos de estudios existían en el mercado varias herramientas que servían para recolectar datos de manera automática y generar informes como *LinkScan*[®], *WebKing*[®] y *SiteSweeper*[®], entre otras; ninguna de las mencionadas proveían un conjunto completo de métricas automatizables, conforme a las necesidades especificadas en los requerimientos de calidad y a las de los evaluadores participantes. Por otra parte, se hacía difícil la integración de las salidas producidas por esas herramientas con ciertos procesos de evaluación definidos en la metodología Web-site QEM.

Por tal motivo, surgió la idea de diseñar y construir una herramienta (Website MA) que automatizara todas las métricas Web (potencialmente automatizables, como veremos en la siguiente

sección), y además, que dicha herramienta pudiera ser integrada a los procesos de la metodología Web-site QEM.

El objetivo principal del proyecto es el de incorporar Website MA a un paquete de herramientas denominado WebQEM_Tool (en fase avanzada de desarrollo), el cual será una aplicación colaborativa centrada en la Web de soporte a la metodología Web-Site QEM. A través de estas herramientas los evaluadores podrán especificar, diseñar, obtener valores elementales y globales de evaluación, y brindar recomendaciones justificables en un ambiente de colaboración (mediante canales de chats, pizarras, hiperdocumentos, etc.). En definitiva, el objetivo final es lograr un conjunto de herramientas homogéneas, interoperables y colaborativas, que ayuden a la automatización de la mayoría de las fases y procesos de la metodología citada.

3. Métricas Web Automatizables.

Los atributos de un ente (producto, proceso o recurso), se categorizan en directos e indirectos [1, 3]. Por otra parte, los atributos pueden ser parcial o totalmente automatizados en consideración del tipo de recolección de datos. Más precisamente, para determinar el valor de un atributo, puede hacerse de un modo manual, o asistido total o parcialmente por un instrumento de medición o herramienta de recolección de datos y/o cálculo.

Si bien muchos de los datos recolectados para las métricas empleadas en los casos de estudio realizados para sitios académicos, de comercio electrónico, entre otros, fueron implementados de un modo manual u observacional, dado que no había otro modo de hacerlo efectivamente, no obstante, la recolección de datos automática fue en muchas instancias el único mecanismo para obtener resultados de un modo confiable y eficiente. Este fue el caso para medir atributos como *Enlaces Rotos*, *Promedio de Enlaces por Página*, *Páginas de Acceso Rápido*, *Imagen con Título*, *Páginas Muertas*, entre otros. A seguir, presentamos el listado de las métricas que hemos logrado automatizar mediante el empleo de la herramienta Website MA:

- *Cantidad de Enlaces Rotos*. Este atributo representa la cantidad de enlaces rotos (broken o dangling links) internos al sitio, como la cantidad de enlaces rotos que referencian a sitios (URLs) externos. Website MA permite almacenar los URLs de los enlaces rotos, tanto internos como externos, para un análisis posterior y posible corrección. El chequeo de un enlace roto se comprueba por medio del tipo de error 404, conforme a los códigos de estado del protocolo HTTP. Asimismo, en consideración del código de error devuelto, se puede determinar la cantidad de enlaces que conducen a páginas no accesibles.
- *Cantidad Total de Enlaces de un Sitio*. Se puede recolectar automáticamente la cantidad total de enlaces que posee un sitio Web. La herramienta permite calcular con gran efectividad esta métrica empleando procedimientos recursivos en el seguimiento de todos los enlaces que posee el sitio que se está analizando.
- *Porcentaje de Enlaces Rotos de un Sitio*. Mediante el empleo de las métricas directas mencionadas arriba (*Cantidad de Enlaces Rotos* y *Cantidad Total de Enlaces de un Sitio*), se computa el porcentaje de enlaces rotos de un sitio, establecido por la fórmula siguiente:

$$\text{PorcentajeEnlacesRotos} = \frac{\text{CantidadEnlacesRotosInternos} + \text{CantidadEnlacesRotosExternos}}{\text{CantidadTotalEnlaces}} \times 100$$

- *Cantidad de Páginas Muertas*. Este atributo representa la cantidad de páginas que no tienen ningún enlace de retorno dentro del sitio (también denominadas páginas huérfanas). Si un

visitante accediera a una página muerta desde un URL externo al sitio, por ejemplo, a través de un enlace recuperado por un buscador en Internet (Yahoo, Altavista, etc.) no tendría la posibilidad de navegar dentro del sitio. Sería simplemente una página sin ningún vínculo. Esto es fácil de automatizar comprobando para cada página visitada la no existencia de la palabra reservada HREF propia del código HTML.

- *Cantidad de Imágenes de un Sitio que Referencian a la Propiedad ALT.* En las páginas Web se debe proveer texto alternativo para cada imagen u objeto gráfico dado que comunican información visual. La presencia de la propiedad ALT (de HTML) en cada imagen permite incluir texto alternativo para cada una de ellas, y de esta manera, se favorece la legibilidad de la página cuando se deshabilita el comando “ver imágenes” del navegador. No obstante, si deseáramos comprobar la calidad del texto alternativo la presencia de esta propiedad por sí sola no la garantiza (el texto puede haber sido generado automáticamente con alguna herramienta de edición como FrontPage, etc.). Analizando el código HTML es fácil detectar la presencia de la propiedad ALT.
- *Cantidad Total de Imágenes de un Sitio.* Al igual que para los enlaces de un sitio es fácil detectar la cantidad de imágenes que posee un sitio Web. Para ello, basta con ir recolectado de cada una de las páginas analizadas, la presencia de imágenes verificando su existencia a través de la propiedad IMG soportada por el código HTML. El valor de la métrica es el resultado de un simple conteo.
- *Porcentaje de Presencia de la Propiedad ALT.* Mediante el empleo de las métricas anteriores, *Cantidad de Imágenes de un Sitio con Propiedad ALT* y *Cantidad Total de Imágenes de un Sitio*, se computa el porcentaje de presencia de la propiedad ALT. La fórmula es la siguiente:

$$\text{PorcentajePresenciaALT} = \frac{\text{CantidadImágenesALT}}{\text{CantidadTotalImágenes}} \times 100$$

- *Cantidad Total de Páginas de un Sitio.* El valor de la métrica es un simple conteo del total de las páginas del mismo.
- *Promedio de Enlaces por Página.* Obtenido a partir de la división de las métricas *Cantidad Total de Enlaces* y *Cantidad Total de Páginas*. Esta métrica da la idea del nivel de interconexión de un sitio (ver [11] para más detalles).
- *Tamaño de una Página.* Se mide el tamaño de una página estática, considerando todos sus componentes gráficos, textuales, tabulares, etc. Así podemos contabilizar el tamaño medido en Kbytes para cada página.
- *Páginas de Acceso Rápido.* Se mide el tamaño de todas las páginas estáticas del sitio Web (del lado del servidor) considerando todos sus componentes gráficos, tabulares y textuales. El tamaño de cada página se especifica como una función del tiempo de espera y de la velocidad mínima establecida para una línea de comunicación dada.
- *Mantenimiento del Color de los Enlaces.* Sería deseable que el color de los enlaces (textuales) no cambie, respetando preferentemente al estándar de tonalidad azul -para los enlaces no visitados, y púrpura -para los ya visitados. La métrica computa la cantidad de cambios en todo el sitio para ambas categorías [2]. Se analizan los códigos de color para las propiedades LINK (para el caso de los enlaces no visitados) y VLINK (para los ya visitados) en la etiqueta BODY del código fuente HTML .
- *Cantidad Máxima de Marcos o Frames.* Esta métrica es de utilidad para determinar el número de vistas, la cual puede afectar a la accesibilidad. Los frames o marcos organizan a una ventana en diferentes áreas o subvistas, a saber: de control y de contenido. Cuanto

mayor es la cantidad de frames, menor es la accesibilidad de ventanas, principalmente para personas con discapacidades, entre otros inconvenientes informados [14]. La guía provista por W3C en *WAI Accessibility Guidelines* [14], dice: “*For visually enabled users, frames may organize a page into different zones. For non-visual users, relationships between the content in frames (e.g., one frame has a table of contents, another the contents themselves). must be conveyed through other means*”. Verificando en cada una de las páginas analizadas si existe la etiqueta FRAME, se puede comprobar de manera automática la cantidad de frames o marcos que contiene la página. Analizando las demás páginas se obtiene la cantidad máxima de frames que contiene un sitio Web.

4. Detalles de Diseño e Implementación de Website MA.

4.1. Módulos Funcionales de la Herramienta.

Como indicado previamente, Website MA sirve de soporte al proceso de evaluación de métricas en la Web. Permite la automatización de las métricas descritas en la sección 3 y la generación de informes en formato gráfico y textual. Además es capaz de administrar distintos proyectos, es decir, sitios Web representados por sus URLs, cada uno con sus respectivas opciones de métricas a seleccionar e informes. La herramienta ha sido diseñada en cuatro módulos principales, a saber:

- *Interface de Usuario de la Aplicación.* Incluida en el módulo *GUI-Java* como se muestra en la fig. 1. Este módulo, permite al usuario interactuar con la aplicación por medio de la IU implementada en Java, haciendo uso de las facilidades del paquete *Swing*.
- *Módulo de Base de Datos.* Es el encargado de administrar la información de los proyectos a evaluar, y de los resultados obtenidos de las métricas configuradas para un proyecto dado. A partir de las tablas de la Base de Datos (BD), la herramienta genera informes personalizados.
- *Módulo de Interface Java-WebL.* Es un módulo con clases Java, que sirve de interface para interactuar entre el módulo *WebL*, con los demás módulos implementados en Java, como se puede apreciar en la fig. 1.
- *Módulo WebL.* Es el encargado de manipular los objetos HTML, para realizar el recorrido y análisis de los sitios, conforme a las métricas configuradas por los evaluadores.

4.2 Detalles de Implementación

En consideración de los módulos arriba descriptos veremos algunos detalles principalmente de implementación. Entre los entornos utilizados para analizar archivos HTML e integrar las facilidades de cálculo y documentación, se encuentran los lenguajes de programación WebL [5, 8], diseñado específicamente para manipular objetos y atributos en el código fuente HTML, y el ambiente de programación Java 2. Website MA se encuentra conformada por varias clases Java. Las clases *WebMA_GUI*, *ConexionDB*, *InterfaceJavaWebL* y *LlamarWebL*, entre otras, como se puede apreciar en la fig. 2

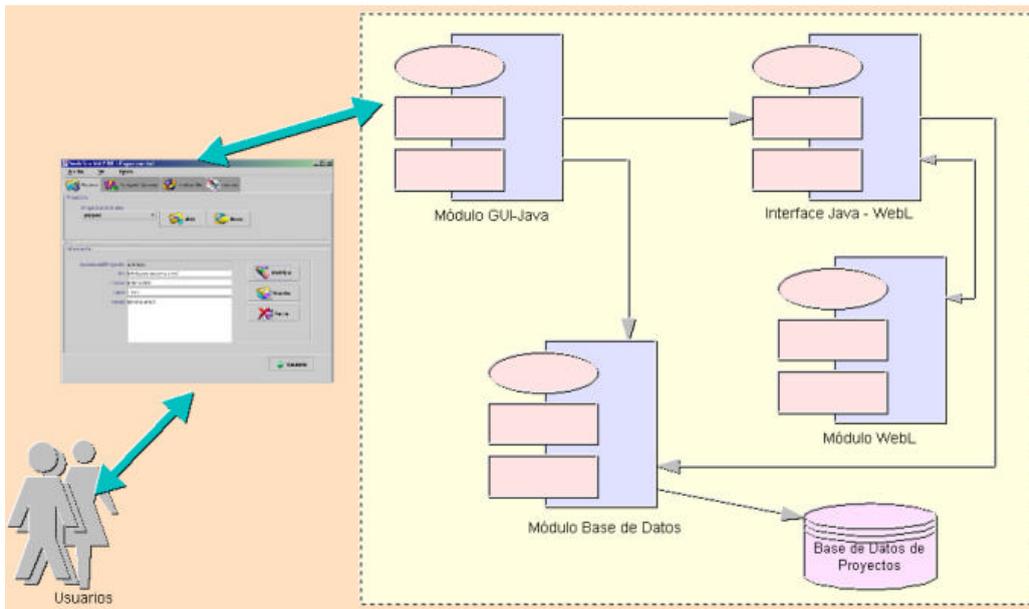


Fig. 1 Representación de los módulos principales que intervienen en Website MA

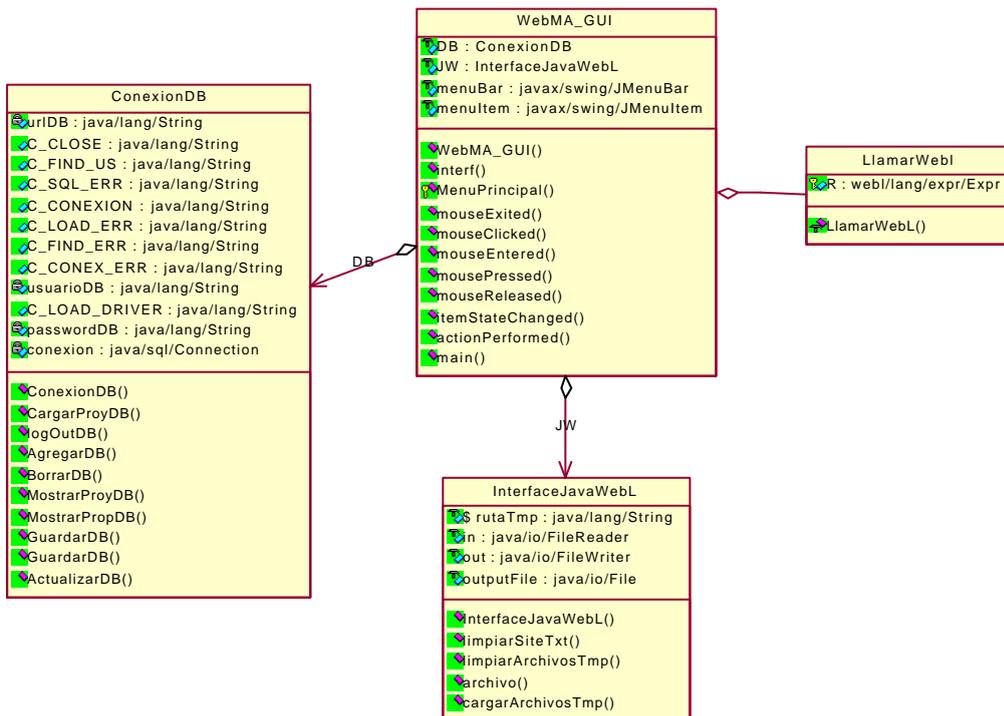


Fig. 2 Diagrama parcial de las principales Clases de la herramienta Website MA.

4.2.1 La Clase WebMA_GUI. Dentro de la funcionalidad provista por esta clase, se encuentran métodos que se encargan de administrar la IU (módulo *GUI-Java*), de monitorear los eventos producidos por el usuario, como así también de entrelazar a los distintos módulos restantes.

Utilizando el componente *JTabbedPane* soportado por el paquete *Swing* de Java, la clase *WebMA_GUI* permite generar a través de sus métodos cuatro pantallas. Cada una de ellas está ubicada en la IU de manera secuencial estableciendo un orden de prioridad natural de

procesamiento de las tareas de evaluación; es decir, el orden de actividades que un evaluador seguiría al valorar un sitio Web.

La primer pantalla que se presenta por defecto una vez iniciado el sistema, se denomina “Proyectos” mostrada en la fig. 3. En ella se proporcionan las opciones necesarias para la administración de los mismos.

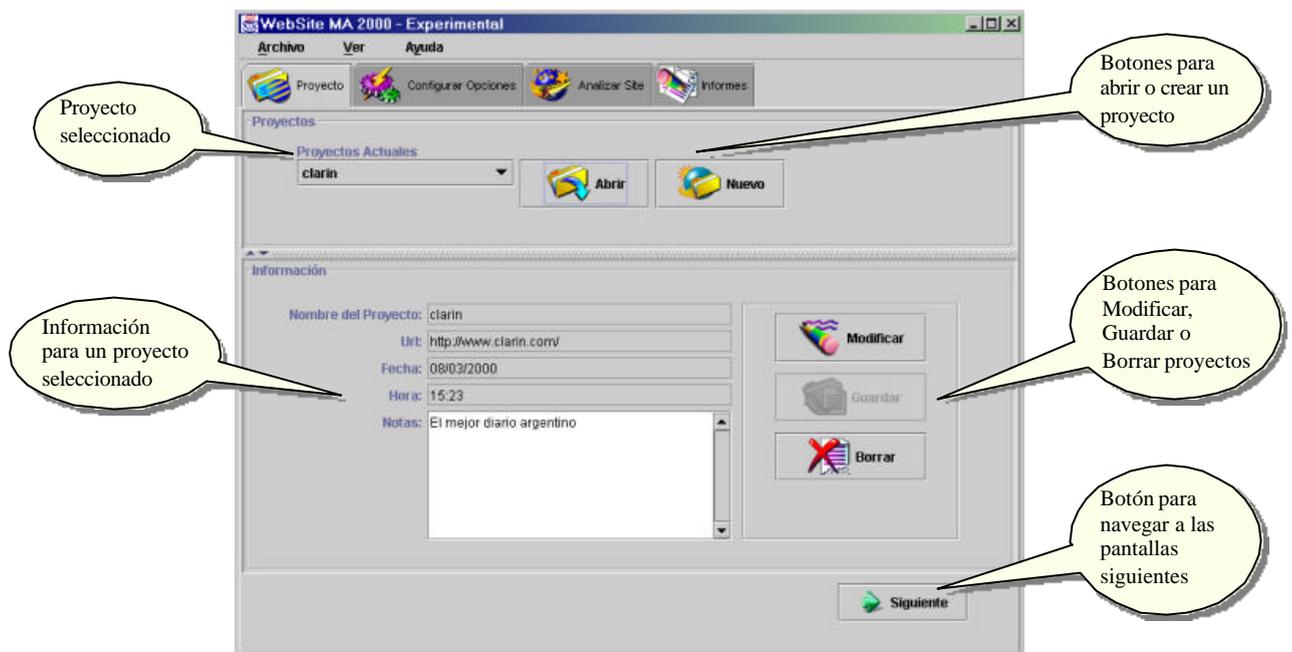


Fig. 3 Pantalla inicial de Website MA.

La segunda pantalla denominada “Configurar Opciones”, mostrada en la fig. 4, permite al usuario evaluador configurar las métricas seleccionadas, para luego realizar el análisis del proyecto Web.

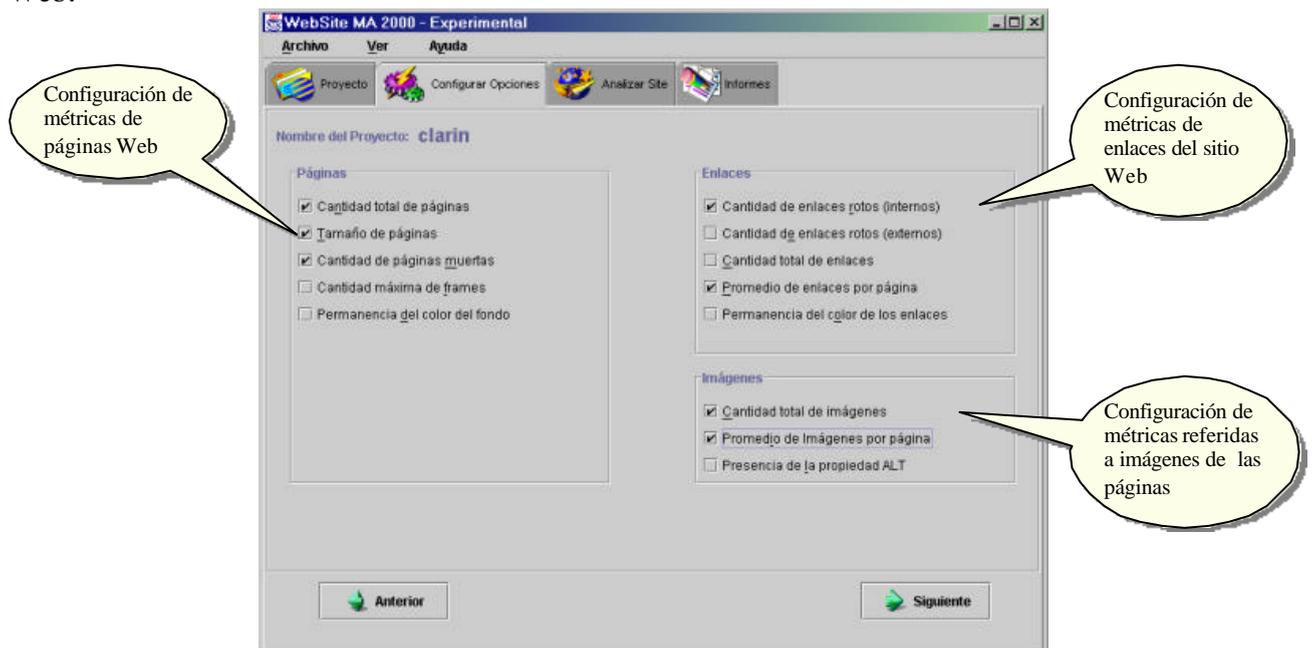


Fig. 4 Segunda pantalla de Website MA. El evaluador configura las métricas que desea analizar para un proyecto Web.

Las opciones (o métricas) se encuentran categorizadas en tres grupos, según se aprecia en la fig. 4.

El grupo de *Páginas* con las métricas *Cantidad Total de Páginas*, *Tamaño de Páginas*, *Cantidad de Páginas Muertas*, *Cantidad Máxima de Frames* y *Permanencia del Color del Fondo*.

El grupo de *Enlaces* con las métricas *Cantidad de Enlaces Rotos*, *Cantidad Total de Enlaces*, *Promedio de Enlaces por Página* y *Permanencia del Color de los Enlaces*.

Por último, en el grupo *Imágenes*, tenemos *Cantidad Total de Imágenes*, *Promedio de Imágenes por Página* y *Presencia de la Propiedad ALT*.

Una vez que el evaluador ha procedido con la apertura de un proyecto y la selección de las métricas, entonces está en condiciones de avanzar a la tercer pantalla denominada “*Analizar Sitio*”. A medida que se analiza el sitio, un área de texto en la pantalla, permite ir registrando y observando todos los acontecimientos que suceden durante el análisis. Ejemplos de estos acontecimientos, pueden ser errores que se produzcan al intentar analizar el sitio debido a URLs equívocos o a problemas de conexión en la red, o eventos que indiquen el comienzo y fin del análisis, entre otros.

Por último, la cuarta pantalla es la encargada de mostrar al usuario los informes generados a partir del análisis de un proyecto Web. Estos informes pueden ser mostrados de manera tabular (fig. 5a), indicando el nombre de la métrica con su respectivo valor, o a través de distintos tipos de gráficos. En la fig. 5b, se aprecia un gráfico de tortas que muestra el porcentaje de presencia de la propiedad ALT.

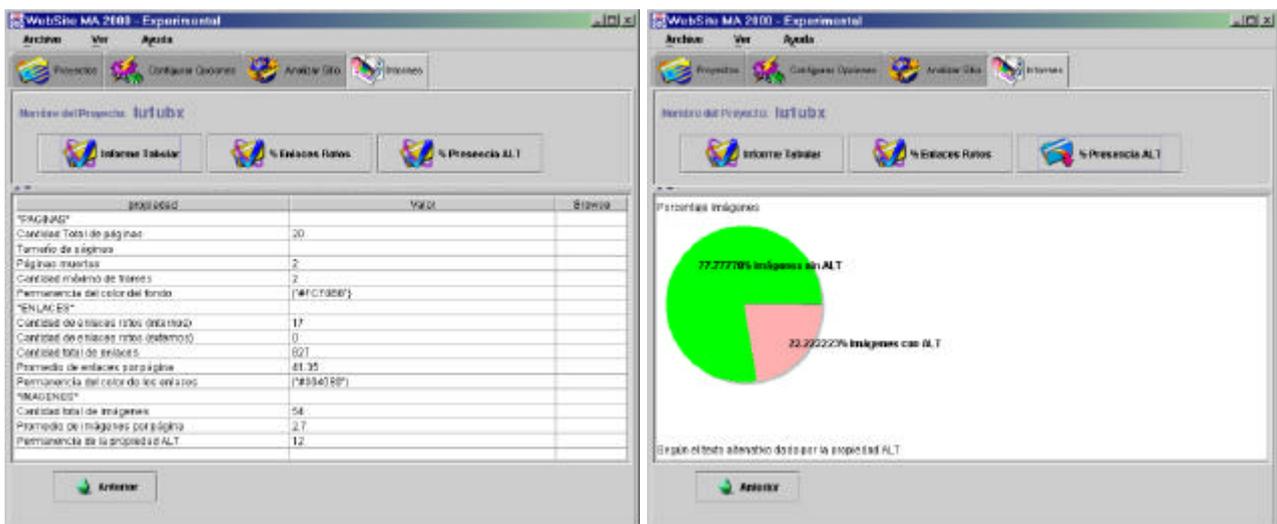


Fig. 5 Cuarta pantalla de Website MA en donde el usuario selecciona el tipo de informe de resultados: **a)** Informe tabular. **b)** Informe gráfico.

4.2.2. La clase ConexionDB. La clase *ConexionDB* provee los mecanismos necesarios para establecer la conexión con la BD física. Esta se logra utilizando el puente JDBC-ODBC [4] soportado por Java y es implementada en el método constructor de la clase *ConexionDB*. Esta clase además de proveer los métodos para establecer el inicio y cierre de la conexión hacia la base de datos, también provee métodos para realizar las altas, bajas y modificaciones de los proyectos que son evaluados, como así también, de métodos para la generación de los informes tabulares y gráficos como los mostrados en la figura anterior.

4.2.3 La clase InterfaceJavaWebL. La clase *InterfaceJavaWebL* (incluida en el módulo de igual nombre) provee a través de sus métodos, mecanismos para establecer una interface de conexión entre los lenguajes Java y WebL. Si bien estos lenguajes son bastante fáciles de comunicarlos entre sí para establecer el pasaje de variables e información temporal, fue necesario construir un módulo interface para coordinar las tareas entre ambos.

Cuando el usuario desde Website MA abre un proyecto, automáticamente se almacena el URL perteneciente al mismo dentro del archivo denominado *site.txt*. Este archivo es tomado luego por el módulo WebL que se encarga de leerlo, interpretarlo, y a partir de él, recorrer la estructura del sitio Web. A medida que va realizando el análisis este módulo va almacenando de manera temporal la información recolectada. Luego esa información se pasa a la BD. De esta manera, todas las operaciones de transacción entre los lenguajes Java y WebL dentro de Website MA son cubiertas por los métodos de la clase *InterfaceJavaWebL* (para mayor detalles ver [6]).

4.2.4 La clase LlamarWebL. Esta clase es la encargada de establecer la conexión desde Java con el módulo WebL. En la figura 6 se muestra una porción del código del módulo *WebL*. Principalmente, queremos representar el módulo que permite analizar las métricas configuradas y el recorrido recursivo [6].

```
//Invocación al módulo Crawler propio de WebL encargado de recorrer
//recursivamente la estructura de un sitio Web.
//Dentro de la variable MyCrawler se analiza cada página visitada

var MyCrawler = Clone(WebCrawler_Crawler,
[.
  Visit = meth(s, page)
  var titulo = (Text(Elem(page, "title")[0]) ? "#- SinTitulo -#");
  PrintLn(page.URL, " titulo=", titulo);

  //analiza si tiene frames para el cálculo de cant. máxima de frames
  every frame in Elem(page, "frame") do
    frames=frames + 1;
  end;

  every image in Elem(page, "img") do
    imagenes=imagenes + 1;
    try
      if image.alt != nil then
        alt=alt + 1;

        //Almacena el texto alternativo de las imágenes en un archivo
        Files_AppendToFile("webl/alt.tmp", ToString(alt)+"\t"+image.alt+"\n");
      end;
    catch e
      on true do
        nil
      end;
    end;

  //Verifica los enlaces internos y externos del sitio
  hrefe=0;
  every a in Elem(page, "a") do
    try
      if a.href != nil then
        hrefe=hrefe + 1;
        try
          var pa;
          pa=GetURL(a.href)
        catch H
```

```

on H.statuscode == 404 do
  if Str_StartsWith(a.href, Str_Trim(site)) then
    error404int= error404int + 1;
    //Almacena los enlaces rotos internos al sitio
    Files_AppendToFile("webl/linklint.tmp",
      ToString(enlaces)+"\t"+a.href+"\n");
  else
    error404ext= error404ext + 1;
    //Almacena los enlaces rotos externos al sitio
    Files_AppendToFile("webl/linklEXT.tmp",
      ToString(enlaces)+"\t"+a.href+"\n");
  end
on H.type == "HttpError" do
  PrintLn("Error de conexión")
on true do
  nil
end;

end
catch W
on true do
  nil
end;
end;
//Analiza la existencia de páginas muertas (huérfanas)
if hrefe==0 then
  muerta=muerta +1;
end;
enlaces=enlaces + hrefe;
paginas=paginas + 1;

..... //Analiza otras métricas .....

try
  var bo=Elem(page, "body");
  var body =First(ToList(bo));
  //Recupera el color de los enlaces (visitados y no visitados) y el
  //color del fondo para cada página
  conjuntolink = conjuntolink + {Str_ToUpperCase(body.link)};
  conjuntovlink = conjuntovlink + {Str_ToUpperCase(body.vlink)};
  conjuntobgcolor = conjuntobgcolor + {Str_ToUpperCase(body.bgcolor)};
catch S
  on true do
    nil
  end;
  //Compara por cada página analizada la cantidad máxima de frames
  if (maxframes < frames) then
    maxframes=frames;
  end;
  frames =0;
end,
ShouldVisit = meth(s, url)
  Str_StartsWith(url, Str_Trim(site))
  and
  Str_EndsWith(url, "(/)|(.html?)")
end,
.);

```

Fig. 6 Implementación en lenguaje WebL del módulo encargado de analizar recursivamente la estructura de un sitio Web y calcular valores de las métricas.

5. Conclusión.

En la actualidad es ampliamente reconocido que un sitio o aplicación Web debe satisfacer necesidades tan diversas (en consideración de las audiencias) que van desde la distribución y presentación de documentos hiperenlazados orientados a diferentes contenidos hasta la automatización de procesos de negocios más o menos complejos. Además, el incesante crecimiento de las tecnologías y aplicaciones Web genera diferentes e interesantes desafíos tanto a los desarrolladores como a los evaluadores. Desde el punto de vista del monitoreo, control y aseguramiento de la calidad en proyectos Web, los procesos de evaluación son actividades relevantes del ciclo de vida Web. Por lo tanto, es preciso definir requerimientos funcionales y no funcionales de manera de poder medir, analizar, comprender y, potencialmente, mejorar la calidad producida. En esta dirección, el empleo sistemático y disciplinado de estrategias, métodos, modelos, y herramientas de Ingeniería Web para el control y aseguramiento de la calidad de los proyectos promoverá el cambio de paradigma de estrategias ad hoc hacia estrategias más objetivas y justificables.

Así, mediante el empleo de la metodología propuesta Web-site QEM [9, 10, 11, 12], el evaluador cuenta con un enfoque sistemático, flexible y cuantitativo para la evaluación, comparación y análisis de la calidad de sitios y aplicaciones Web. Pero además de contar con una metodología, es importante que los evaluadores dispongan de un conjunto de herramientas de soporte que le permitan agilizar los procesos de evaluación, y en la medida de lo posible, es deseable que dispongan de herramientas que se adecuen a las necesidades de la metodología utilizada (ver por ejemplo [7]).

Por una parte, el agregado de soporte automático al proceso de recolección de datos y al cálculo de métricas Web tiene varias ventajas, que van desde la reducción de esfuerzo y costos hasta la minimización de errores e imprecisiones debido a la intervención humana. Por tal motivo, surgió la idea de diseñar y construir la herramienta Website MA [6] la cual automatiza las métricas Web descritas en la sección 3, y además, dicha herramienta se integra a los procesos de la metodología Web-site QEM.

Por otra parte, el objetivo principal del proyecto de I+D, es el de incorporar a Website MA a un paquete de herramientas denominado WebQEM_Tool (actualmente en fase avanzada de desarrollo), la cual será una aplicación colaborativa centrada en la Web de soporte a la metodología. A través de estas herramientas los evaluadores podrán especificar, diseñar, obtener valores elementales, parciales y globales de evaluación, y brindar recomendaciones justificables, en un ambiente colaborativo. En definitiva, el objetivo final es lograr un conjunto de herramientas homogéneas, interoperables y colaborativas, que ayuden a la automatización de la mayoría de las fases y procesos de la metodología.

Agradecimientos.

Esta investigación está soportada por el proyecto UNLPam-09/F013.

Referencias.

1. **Fenton, N.E.; Pfleeger, S.L.**, 1997, “*Software Metrics: a Rigorous and Practical Approach*”, 2nd Ed., PWS Publishing Company.
2. **IEEE Networking the World**, 2000, “*IEEE Web Publishing Guide*”, <http://www.ieee.org/web/developers/style/>

3. **ISO/IEC 9126-1** International Standard (Draft State), “*Information technology – Software product evaluation – Part 1: Quality Model*”
4. **JDBC API Version 2.1**, 1999, Sun Microsystem Inc, Palo Alto, CA, US.
5. **Lafuente, G.J.**, 1999, “*Tutorial de WebL*”, GIDIS, Facultad de Ingeniería, UNLPam. <http://gidis.ing.unlpam.edu.ar/personas/glafuente/WebL/html/index.html>
6. **Lafuente, G.J.**, 2000, “*Automatizando Métricas en la Web*”, Tesis de Licenciatura, Departamento de Sistemas, UNLu., Luján.
7. **Lavazza, L.**, 2000, “*Providing Automated Support for the GQM Measurement Process*”, IEEE Software, V. 17, N. 3, pp. 56-62.
8. **Marais H.**, 1999, “*WebL —A Programming Language for the Web*”, Compaq Systems Research Center (SRC).
9. **Olsina, L.**, 1999, “*Web-site Quantitative Evaluation and Comparison: a Case Study on Museums*”, Workshop on Software Engineering over the Internet, at Int’l Conference on Software Engineering <http://sern.cpsc.ucalgary.ca/~maurer/ICSE99WS/ICSE99WS.html>, Los Angeles, US.
10. **Olsina, L.; Lafuente, G.J.; Godoy, D; Rossi, G.**; 1999, “*Assessing the Quality of Academic Websites: a Case Study*”, In: New Review of Hypermedia and Multimedia (NRHM) Journal, Taylor Graham Publishers, UK, Vol. 5, pp. 81-103.
11. **Olsina L.**, 2000, “*Metodología Cuantitativa para la Evaluación y Comparación de Calidad de Sitios Web*”, Tesis doctoral defendida en Abril, Facultad de Ciencias Exactas, UNLP, La Plata.
12. **Olsina, L.; Lafuente, G.J.; Rossi, G.**, 2000, “*E-commerce Site Evaluation: a Case Study*”, In LNCS of Springer-Verlag, 1st International Conference on Electronic Commerce and Web Technology, London-Greenwich, UK
13. **Pfleeger, S. L.**, 1993, “*Lessons Learned in Buliding a Corporate Metric Program*”, IEEE Software, No. 3, pp. 67-74.
14. **WWW Consortium**, 1999, “*WAI Accessibility Guidelines: Page Authoring*”, W3C Working Draft, “WAI Accessibility Guidelines: Page Authoring”, <http://www.w3c.org/TR/WD-WAI-PAGEAUTH/>