

Fresita: Una Herramienta Distribuida para la Creación Comunitaria de Recetas

Martín Farías*[†], Mariano Guerra*, Diego Sarmentero*[†],
César Martínez Spessot*[†], Ricardo Medel*[†]

*Universidad Tecnológica Nacional
Facultad Regional Córdoba
Departamento de Ingeniería de Sistemas de Información

[†]Intel Corp.
Argentina Software Development Center

Resumen. Transmitir el conocimiento empírico a través de recetas (o *howtos*), tiene como contrapartida la dificultad para sistematizarlas a través de un léxico y una estructura estándar, almacenarlas en repositorios comunes y mantener una jerarquía entre recetas originales y derivadas. *Fresita* apunta a resolver estos problemas por medio de la definición de una estructura estándar para recetas y la creación de las herramientas necesarias para automatizar, en la medida de lo posible, la creación y administración de recetas. En este artículo presentamos una herramienta que permite a cada miembro de una comunidad crear y editar recetas e ingredientes individuales y hacerlos accesibles al resto de los usuarios.

Palabras clave: recetas, how-to, sistemas distribuidos, sistema de resolución de dependencias

1. Introducción

En el ámbito informático el conocimiento empírico sobre uso de la tecnología y las mejores prácticas son sistematizadas y transmitidas a través de documentos llamados *howto* por su nombre en inglés¹. Obviamente, este proceso no está acotado solamente al ámbito informático, las recetas de comidas o los protocolos médicos, por ejemplo, son formas similares de apropiación y distribución del conocimiento a través de canales más o menos formales. En este artículo utilizaremos el término **receta** para referirnos a documentos utilizados para sistematizar y transmitir las mejores prácticas de una disciplina a través de una lista de **ingredientes** y una lista de **pasos** a seguir.

Los problemas que los usuarios de recetas enfrentan es la falta de repositorios comunes donde hallar las recetas buscadas, la falta de un léxico y/o estructura estándar y la dificultad para derivar una nueva receta a partir de una anterior, estableciendo claramente las relaciones entre ambas. En el caso particular de las recetas en informática se suma, además, la dificultad de mantener las recetas actualizadas frente a la constante variación en las versiones de programas o librerías.

A fin de resolver el primer problema, es decir la falta de repositorios comunes donde hallar recetas, se han creado sitios web, propietarios o comunitarios, tales como *Howtopedia* [7], un *wiki*² dedicado a la creación comunitaria y la publicación de

1 La forma correcta en inglés es *how-to*. Sin embargo, en la comunidad Linux y, por extensión, en la comunidad informática en general se los denomina *howto*.

2 *Wiki*: sitio web cuyas páginas pueden ser editadas por los visitantes a través de un

recetas informáticas, *The Recipe Circus* (El Circo de las Recetas) [4,16], un sitio web para compartir recetas de cocina y *Recipes Wiki* (Wiki de Recetas) [10], un *wiki* de recetas de cocina. Estas soluciones, sin embargo, no son suficiente para resolver los otros problemas mencionados, y además funcionan solamente a través de Internet, sin copia local. Por lo cual el usuario depende de una conexión a Internet permanente y confiable.

Este proyecto apunta a resolver los problemas de creación y administración de recetas por medio de la definición de una estructura estándar para las recetas, de modo que sea lo suficientemente flexible como para poder ser usada en distintas disciplinas, y la creación de las herramientas necesarias para la automatización de los procesos involucrados.

En este artículo presentamos el sistema **Fresita**, un sistema distribuido para la creación y administración de recetas en forma colaborativa. El sistema aporta a la creación y administración de recetas distribuyendo las recetas y sus partes constituyentes (**ingredientes**) en un sistema distribuido. Fresita, además, provee una interfaz de usuario sencilla, que no requiere entrenamiento especial para poder usarla en todo su potencial. El artículo presenta la estructura y funcionalidad actual de la herramienta en la Sección 2. Se realiza un análisis de sus limitaciones y se proponen mejoras en la Sección 3. La Sección 4 reporta las conclusiones de este trabajo.

2. Estructura y Funcionalidad Actual de Fresita

El principal objetivo tras el desarrollo de Fresita es automatizar la creación y distribución de recetas a fin de permitir a los miembros de una comunidad aumentar el conocimiento general de dicha comunidad en una forma sencilla y sin requerir entrenamiento especial. Es por esto que la herramienta abstrae el formato de los documentos, almacenados como documentos XML, y establece un estándar simple para el formato visible de las recetas. En este sentido, para un usuario de Fresita una receta es un documento con un título, uno o varios autores, un resumen, una lista de palabras clave, un índice de contenidos y una lista de pasos a realizar o ingredientes. Recetas e ingredientes son fáciles de crear a través de la interfaz gráfica que provee el sistema.

Aprovechando los protocolos de comunicación SOA (Arquitectura Orientada a Servicios, o *Service Oriented Architecture* por su nombre en inglés) [5, 6] y las ventajas ofrecidas por los servicios web [2, 9, 12], Fresita permite el intercambio de información entre estaciones de trabajo distribuidas, permitiendo la integración de recetas e ingredientes desde cualquier punto de la red, en forma independiente del sistema operativo utilizado en cada nodo. A pesar de tener una arquitectura descentralizada, en su versión actual Fresita es una aplicación *standalone* que requiere de la creación de sub-redes *ad-hoc* a fin de permitirle al usuario definir con cuáles nodos intercambiar información.

Internamente, una receta es un documento XML con marcadores para cada parte de la receta, historial de revisiones y referencias a sus ingredientes. Esto permite reutilizar ingredientes, mantener actualizadas las recetas que comparten ingredientes y redefinir recetas en forma sencilla. En esta versión la estructura jerárquica entre

navegador web. Los usuarios pueden crear, modificar o borrar texto compartido, con poca o ninguna limitación.

recetas e ingredientes es sencilla: un **XML Base** (recetas) contiene **XML Hijos** (ingredientes). Un análisis formal más completo de la estructura y relaciones entre recetas e ingredientes se está llevando a cabo en la actualidad (ver sección 3.7).

2.1. Conectividad a través de Redes Ad-hoc

El sistema puede trabajar localmente, en cuyo caso el usuario podrá trabajar sin conexión a red, pero no podrá acceder a documentos externos ni podrá poner los suyos a disponibilidad del resto de la comunidad.

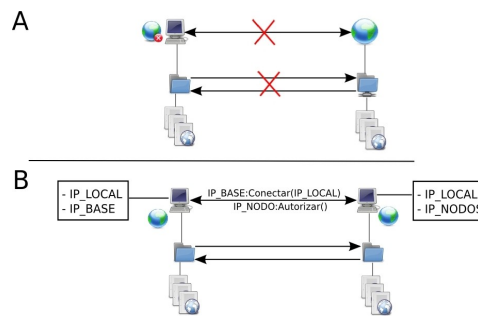


Figura 1. (A) Sistema Fresita funcionando localmente desconectado de la red. .
 (B) Sistema Fresita pidiendo autorización de conexión a Nodo Base y habilitando intercambio de Recetas/Ingredientes.

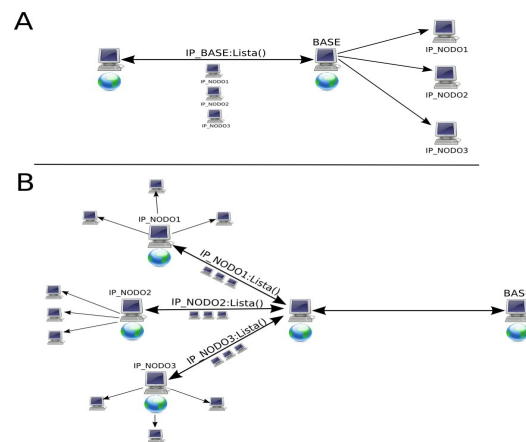


Figura 2. (A) Sistema Fresita local solicita a nodo base su lista de nodos conocidos.
 (B) Sistema Fresita local utiliza la lista de nodos conocidos para requerir las listas de nodos conectados y así extender la red.

En caso de querer trabajar en red, el usuario debe ingresar la configuración o IP de algún nodo conocido de la red, el cual se denominará **nodo base**. Una vez que esta configuración es cargada, el sistema local requiere que el nodo base le envíe la lista de nodos conectados a dicho nodo base (Figura 1). De esta manera la instancia local de Fresita toma conocimiento de las otras instancias conectadas con las cuales le será

posible dialogar. De ser necesario, el sistema puede requerir a otros nodos la lista de los nodos conectados a ellos, a fin de seguir ampliando la red de nodos alcanzables (Figura 2).

2.2. Estructura de Recetas e Ingredientes

En nuestro modelo, una receta está compuesta por una serie de ingredientes. Cada ingrediente explica los pasos requeridos para completar una parte de la receta. Recetas e ingredientes son almacenados como archivos XML con estructuras especializadas. La Figura 3 muestra los esquemas XML para recetas (a) e ingredientes (b).

<pre> <recipe> <intro> <id/> <reciperevision/> <title/> <authors> <author> <name/> <surname/> <email/> </author> </authors> <revisions/> <tags> <tag/> </tags> </intro> <contenido> <abstract/> <index/> <chapter/> <order/> <name/> <ingredients> </ingredients> </contenido> </recipe> </pre>	<pre> <ingredient> <id/> <ingredientrevision/> <title/> <description> TEXT </description> <authors> <author> <name/> <surname/> <email/> </author> </authors> <revisions/> <steps> <step> <title/> <content> TEXT </content> </step> </steps> <tags> <tag/> </tags> </ingredient> </pre>
---	--

Figura 3. (a) Esquema XML para recetas. (b) Esquema XML para ingredientes.

Una receta es almacenada como un archivo XML conteniendo una descripción a alto nivel de la receta. Esta descripción incluye un título, información sobre el o los autores que intervinieron en su creación, una lista de palabras clave o *tags*, un resumen y referencias a los ingredientes. La lista de *tags* ayuda a identificar la receta y será utilizada en la realización de búsquedas.

Por cada receta, Fresita genera un índice utilizando las distintas secciones de los ingredientes. Los ingredientes son incluidos en una receta por medio de referencias relativas, ya que el directorio que Fresita utiliza es fijo y creado automáticamente.

La estructura de los archivos XML representando ingrediente es sencilla; sólo contienen una descripción del ingrediente, una lista de autores y tags, y cada uno de los pasos que componen el ingrediente. Cada paso a su vez está compuesto por un título y un texto que explicita el contenido del paso.

2.3. Estructura de Directorios

Fresita trabaja con una estructura fija de cuatro directorios para almacenar la información del usuario y el programa: `database`, `lib`, `ingredients` y `howto`.

El directorio `howtos` contiene los archivos XML correspondiente tanto a las recetas creadas por el usuario como a las descargadas de la red.

Todos estos directorios son creados durante la instalación de Fresita. En la versión actual tanto la ubicación como los nombres de estos directorios son fijos y no pueden ser cambiados.

El directorio `database` contiene un archivo XML que contiene los tags pertenecientes a los ingredientes locales. La información de cada *tag* incluye el nombre de las recetas y/o ingredientes a los cuales pertenece y referencias a la posición de estos elementos en la estructura de directorios. Esta información permite agilizar la construcción y modificación de recetas, mostrando en la interfaz de usuario el nombre de las recetas e ingredientes hallados en la búsqueda basada en *tags* y permitiendo acceder a los documentos directamente.

El directorio `ingredients` contiene los archivos XML correspondientes tanto a los ingredientes creados por el usuario como a los descargados de la red. Fresita lista todos estos ingredientes en la interfaz al comenzar la creación de una receta.

El directorio `lib` contiene información sobre las configuraciones del usuario y del servidor. No se almacena información histórica en este directorio, sólo la información de la última configuración establecida por el usuario.

2.4. Funcionalidad de la Herramienta

La funcionalidad provista por Fresita puede dividirse en cuatro funciones principales: creación de ingredientes, creación de recetas, manejo de revisiones y búsqueda. Cada una de estas funcionalidades son descritas con más detalle en las subsecciones siguientes. Una interfaz gráfica permite a usuarios sin entrenamiento formal acceder a la funcionalidad de la herramienta en una forma sencilla e intuitiva. La interfaz (ver Figura 5) representa los contenidos de los archivos XML en forma gráfica, escondiendo su complejidad, y divide las tareas de creación y modificación de recetas en simples pasos secuenciales. En la sección 3.4 se plantea una simplificación aún mayor para esta interfaz, teniendo en cuenta principalmente el modo en que un usuario no entrenado ataca el problema de creación de recetas.

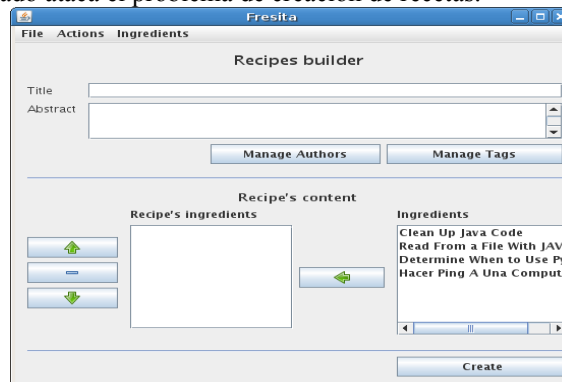


Figura 5. Interfaz principal de Fresita.

2.4.1. Crear Ingrediente

Los ingredientes son los bloques básicos de las recetas. En Fresita, la interfaz para crear ingredientes solicitará al usuario un conjunto de palabras clave que identifiquen al nuevo ingrediente. Dado que uno de los peligros presentes cuando se utiliza un gran conjunto de recetas creadas comunitariamente es la reinención de recetas o ingredientes ya existentes, antes de crear el nuevo ingrediente el sistema sugerirá al usuario un conjunto de documentos que tengan las mismas palabras clave y que ya existan localmente o en los nodos conocidos. Esto evita, en algunos casos, reescribir un documento que ya existe.

En caso de que uno de los documentos presentados por el sistema sea similar al ingrediente que el usuario desea crear, el sistema puede comenzar a funcionar en modo de edición a fin de modificar el documento para adaptarlo a las necesidades del usuario. Si no hay ingredientes similares al creado, el usuario deberá proceder a crear uno desde cero. Las palabras clave ingresadas son entonces asociadas a este nuevo ingrediente.

2.4.2. Crear Receta

A fin de crear una receta, el usuario debe ingresar el título, el o los autores, las palabras claves asociadas a la nueva receta y un resumen. Luego el usuario procede a seleccionar los ingredientes que formarán la receta y a ordenarlos adecuadamente. Como resultado de este proceso se genera un documento XML que referencia a los ingredientes y además contiene un índice generado automáticamente.

Luego de ser creado, el documento queda automáticamente disponible para el resto de la comunidad de usuarios. Una ventaja de usar el formato XML es que a través de XSLT [15] es posible exportar y visualizar los documentos generados en formato HTML, haciéndolos accesible a una amplia audiencia.

2.4.3. Control de Revisiones

El módulo de control de revisiones de Fresita analiza documentos similares disponibles en la red a fin de establecer, en base a la cantidad de modificaciones recibidas, cuáles son las recetas sobre las cuales se han realizado más desarrollo. Esto permite al usuario elegir la versión más actual de una receta, la que haya sido más trabajada o la que le parezca más segura.

A fin de identificar distintas versiones de recetas e ingredientes, Fresita agrega a cada documento tres números separados por un punto. Cada número representa cambios producidos en el documento, siendo el número de la izquierda el más representativo y el número de la derecha el menos representativo.

Si bien es posible recuperar versiones anteriores de una receta buscando en nodos donde esta receta no haya sido modificada, Fresita no brinda aún la posibilidad de recuperar una copia previa de una receta revisando la evolución de dicha receta en una instancia local del sistema. Es decir, el módulo de control de revisiones no mantiene una historia de cada receta que permita recuperar la receta en el estado previo a las modificaciones, sólo provee la receta en su estado actual y una descripción mínima de los cambios realizados.

2.4.4. Búsqueda

Como se mencionara previamente, la búsqueda de documentos, ya sean ingredientes o

recetas, se realiza mediante el uso de palabras clave (*tags*). El resultado de una búsqueda exitosa será una lista de los documentos encontrados, lo cual permite descargar los documentos deseados a fin de ser impresos, editados o utilizados en la creación de nuevas recetas. El módulo de búsqueda permite también realizar una previsualización de los documentos encontrados en la red externa antes de descargarlos al directorio de trabajo local.

3. Análisis Crítico y Trabajos Futuros

En esta sección realizamos un análisis crítico de las características y funcionalidades actuales de la herramienta y proponemos trabajos futuros, planeados o en fase de ejecución, para mejorar estos aspectos de Fresita.

3.1. Estructura XML

Como se mencionara anteriormente el modelo actual de Fresita para representar recetas en XML permite agregar referencias a archivos representando ingredientes. Sin embargo, un análisis más profundo de la estructura de las recetas reales permite ver que a veces los ingredientes son en sí mismos una receta. Por otra parte, el formato actual de los ingredientes no permite agregar referencias a otros ingredientes más básicos.

Todo esto lleva a pensar que el esquema actual de dos niveles (receta – ingredientes) es demasiado rígido como para representar recetas de cierta complejidad. Una receta entonces debería ser un árbol de elementos, algunos de los cuales serán en sí mismos tan complicados como una receta. Es por esto que nos proponemos readaptar el esquema XML a fin de reflejar esta nueva visión de las recetas como una estructura recursiva.

3.2. Control de Revisiones

El esquema de manejo de revisiones permite llevar la cuenta de las modificaciones, pero, como se mencionara previamente, no permite recuperar versiones previas del documento. En este momento estamos evaluando la conveniencia de desarrollar más este módulo o adaptar algún software de control de versiones, tales como CVS [14, 16], SVN [3] o Git [13, 17], a Fresita.

3.3. Conexión Dinámica y Acceso Web

El manejo actual de IPs es estático, con lo cual los vínculos entre usuarios de distintos nodos pueden perderse en caso de utilizar IP dinámicas. Se está analizando la posibilidad de establecer un servidor central que permita manejar en forma distribuida las conexiones de los usuarios incluyendo instancias que posean IP dinámicas.

Si bien el primer prototipo de Fresita es *standalone*, hemos decidido encarar una implementación web de la aplicación a fin de que cualquier usuario pueda acceder a ella desde cualquier lugar con acceso a Internet y pueda utilizar los conocimientos de la comunidad y aportar los suyos propios sin necesidad de descargar e instalar la aplicación.

3.4. Mejoras a la Interfaz de Usuario

Se está desarrollando una nueva versión de la interfaz gráfica que permitirá al usuario escribir una nueva receta como si estuviera ingresando texto plano, sin diferenciación

obligatoria de ingredientes. Esta característica, sumada a la posibilidad de agregar secciones completas con un *click*, permitirá al usuario usar el sistema en una forma más intuitiva, con el resultado final de la receta siempre a la vista.

Asimismo, dado que diferentes usuarios o empresas tienen diferentes estilos para sus documentos, la nueva interfaz permitirá modificar el estilo del documento final sin modificar el contenido XML subyacente en cada receta. Nuevos estilos podrán ser definidos y agregados por los usuarios.

3.5. Contadores de Popularidad y Usuarios Especializados

Tal como ya lo hacen muchas páginas web, un contador de popularidad basado en la cantidad de veces que una receta ha sido descargada permitirá establecer un *ranking* que facilite a los usuarios la búsqueda de recetas probadas y aceptadas por la comunidad.

Otro aspecto que se ha generado naturalmente en el ambiente comunitario de la Web 2.0 [11] es el surgimiento de usuarios especializados en determinados temas. Esto permite a los usuarios recurrir a especialistas cuando se desea trabajar sobre cierto tema. Los usuarios especializados, a su vez, reciben información sobre las modificaciones que se realizan a las recetas que caen dentro de su área de conocimiento, permitiéndoles centrar sus responsabilidades en cierto grupo de documentos.

3.6. Algoritmos de Auto-Resumen

Se planea incluir en Fresita algoritmos de auto-resumen (*auto-tagging*), los cuales permiten obtener las ideas principales de elementos de texto con los que se está trabajando, de modo de generar *tags* automáticamente. Estos algoritmos se basan en la eliminación de pronombres, artículos y otros elementos del lenguaje natural considerados irrelevantes en este contexto, a fin de obtener una lista de palabras que identifiquen el elemento.

3.7. Formalización de Relaciones entre Recetas e Ingredientes

El sistema desarrollado permite construir recetas en base ingredientes. Aún extendiendo la actual estructura de dos niveles (receta – ingredientes) a una estructura recursiva, las recetas continúan siendo formadas concatenando ingredientes. Sin embargo, los ingredientes de algunas recetas afectan distintas partes de la receta. Por ejemplo, una receta que incluya el navegador *Firefox* requiere explicar cómo instalarlo al comienzo y cómo usarlo en secciones posteriores. Es decir, las recetas pueden ser una concatenación de **secciones**, con los ingredientes siendo una estructura independiente formada por uno o más **átomos** (en el ejemplo podrían ser *instalación de Firefox* y *uso de Firefox*) que serán incluidos en diferentes secciones de la receta.

ingrediente: conjunto de átomos	
sección: secuencia de átomos	
receta: secuencia de secciones	
\in : átomo \times receta	\in : átomo \times ingrediente
\subseteq : ingrediente \times receta	\supseteq : ingrediente \times ingrediente
\leq_i : átomo \times átomo \times ingrediente	

Figura 6. Definición formal de ingredientes, secciones, recetas y relaciones entre ellos.

Como consecuencia de este descubrimiento, nuestro grupo se embarcó en el desarrollo de una notación formal para la especificación de las relaciones entre recetas, secciones, ingredientes y átomos. La Figura 6 muestra un sistema formal simplificado que establece un formato general para ingredientes (un conjunto no necesariamente ordenado de átomos), de secciones (una secuencia de átomos pertenecientes a ingredientes) y de recetas (una secuencia de secciones). El sistema define cinco relaciones básicas: pertenencia de un átomo a una receta, pertenencia de un átomo a un ingrediente, inclusión de un ingrediente en una receta, orden de dos átomos en un ingrediente y *necesidad* entre dos ingredientes.

Al definir el sistema formal de esta manera, es posible escribir fórmulas lógicas que establezcan restricciones para la creación de recetas. Por ejemplo, puede establecerse que todos los átomos que forman un ingrediente deben incluirse en la receta si se quiere que dicho ingrediente aparezca en la receta.

La fórmula $a \in i \wedge a \in r \Rightarrow \forall a' \in i \bullet a' \in r$ establece que si el átomo a pertenece a un ingrediente i y el átomo ha sido incluido en una receta r , entonces todos los otros átomos del ingrediente i deben también ser incluidos en la receta r . Otra característica deseable de una receta es que sus átomos respeten el orden establecido por los ingredientes.

La fórmula $a \leq_i a' \wedge a, a' \in r \Rightarrow a \leq_r a'$ establece que si un ingrediente i ha definido un orden entre dos átomos (a y a'), cada vez que dichos átomos aparezcan en una receta r , éstos van a aparecer en el orden definido previamente. A veces ciertos ingredientes pueden ser agregados en una receta solamente si otros ingredientes específicos ya han sido agregados. En este caso decimos que un ingrediente es **necesario** para otro ingrediente.

La fórmula $j \supset i \wedge i \subseteq r \Rightarrow j \subseteq r$ establece que si un ingrediente j es necesario para un ingrediente i , dicho ingrediente puede ser agregado a una receta r solamente si j está en dicha receta.

Nótese que mientras las dos primeras fórmulas permiten implementar un sistema de control que indique si las recetas construidas están bien formadas, la última fórmula permite implementar un sistema que ayude a construir recetas en forma semiautomática: cada vez que el usuario agrega un ingrediente a una receta, el sistema debe seguir la última fórmula y agregar automáticamente los ingredientes “necesarios”, si los hubiera.

Al implementar un sistema que tome en cuenta estas fórmulas para la creación de recetas se define un sistema de resolución de dependencias. Estos formalismos han sido utilizados para sistemas de actualización de paquetes de software, especialmente en distribuciones Linux [1,8,18]. Estamos formalizando estas relaciones de dependencia y analizando las herramientas de resolución de dependencias que se deberían usar o desarrollar para extender a Fresita en esa dirección.

4. Conclusiones

En este paper presentamos Fresita, una herramienta desarrollada como resultado de nuestro estudio de los problemas encontrados al intentar establecer repositorios comunitarios de conocimiento. Los principales problemas de este tipo de transmisión de conocimiento son la falta de un léxico y una estructura estándar que simplifiquen la creación, búsqueda, uso y modificación de recetas en diversas disciplinas.

La herramienta permite establecer repositorios locales de recetas y compartirlas por medio de redes *ad-hoc* de nodos. Permite crear recetas y sus partes constituyentes, los ingredientes. Los usuarios pueden buscar recetas e ingredientes por medio de *tags* o palabras claves, y pueden modificar recetas e ingredientes.

Fresita, sin embargo, tiene limitaciones que han sido detectadas y analizadas. En trabajos subsiguientes se mejorará la estructura de las recetas e ingredientes a fin de permitir mayor complejidad y flexibilidad en las recetas, se mejorará el sub-sistema de control de versiones y se automatizará la obtención de *tags*, se mejorará la conexión de redes y se implementará una versión web de la herramienta, se facilitará la creación de comunidades a partir de la inclusión de rankings de popularidad y especialización de los usuarios, y se analizará formalmente el concepto de receta a fin de automatizar en lo posible la creación y modificación de las mismas.

5. Referencias

1. J. Boender, R. Di Cosmo, B. Durak, X. Leroy, F. Mancinelli, M. Morgado, D. Pinheiro, R. Treinen, P. Trezentos, J. Vouillon, “*News from the EDOS project: improving the maintenance of free software distributions*” en actas del VII Workshop de Software Libre (WSL06), Porto Alegre (Brasil), Abril 2006.
2. E. Cerami, “*Web Services Essentials: Distributed Applications with XML-RPC, SOAP, UDDI & WSDL*”, O’Reilly Media, 2002.
3. B. Collins-Sussman, B. W. Fitzpatrick, C. M. Pilato, “Version Control with Subversion”, O’Reilly Media, 2004.
4. <http://recipecircus.com/> (última visita 25/06/2008)
5. T. Erl, “*Service Oriented Architecture (SOA): Concepts, Technology, and Design*”, Prentice Hall PTR, 2004.
6. T. Erl, “*Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*”, Prentice Hall PTR, 2005.
7. <http://www.howtopedia.org/> (última visita 13/05/08)
8. F. Mancinelli, J. Boender, R. Di Cosmo, J. Vouillon, B. Durak, X. Leroy, R. Treinen, “*Managing the Complexity of Large Free and Open Source Package-Based Software Distributions*”, en Proc. ASE’06, Tokio, Japón, Septiembre 2006.
9. L. Richardson, S. Ruby, “*RESTful Web Services*”, O’Reilly Media, 2007.
10. <http://www.recipeswiki.org/> (última visita 25/06/2008)
11. A. Shuen, “*Web 2.0: A Strategy Guide*”, O’Reilly Media, 2008.
12. J. Snell, D. Tidwell, P. Kulchenko, “*Programming Web Services with SOAP*”, O’Reilly Media, 2001.
13. T. Swicegood, “Pragmatic Version Control Using Git”, The Pragmatics Programmers, 2008.
14. D. Thomas, A. Hunt, “*Pragmatic Version Control using CVS*”, The Pragmatics Programmers, 2003.
15. D. Tidwell, “*XSLT*”, O’Reilly Media, 2001.
16. J. Vesperman, “*Essential CVS*”, O’Reilly Media, 2003.
17. C. Worth, “A Tour of Git: the basics”, on-line book, 2007.
<http://cworth.org/hgbook-git/tour/>
18. WorkPackage2 Team, “*Report on Formal Management of Software Dependencies*”, EDOS Project, Report R. 1269, April 2006.