

Trabajo Final  
Especialista en Ingeniería de Software

# **‘Mashups Patterns’**

---

Ingeniero en Sistemas de Información  
Marcos Gabriel Torres

**Director: Doctor Gustavo Rossi**

Facultad de Informática  
Universidad Nacional de La Plata

Febrero del 2013

## Contenido

Introducción.....	4
Web 2.0.....	4
El origen de los Mashups.....	5
Que es un Web Mashup?.....	5
Patrones de Diseño.....	6
Objetivo.....	7
Alcance.....	7
Presentación de cada patrón.....	9
Presentación de los patrones.....	10
Harvest Patterns.....	11
Enhance Patterns.....	11
Assemble Patterns.....	11
Manage Patterns.....	12
Harvest Patterns.....	13
Alerter.....	15
API Enabler.....	18
Competitive Analisis.....	22
Infinite Monkeys.....	26
Leading Indicator.....	29
Sentiment análisis.....	34
Time Series.....	38
Enhance Patterns.....	42
Feed Factory.....	44
Field Medic.....	47
Folksonomy Enabler.....	49
Fragility Reducer.....	52
Smart Suggestions.....	55
Super Search.....	58
Translation.....	61
Usability Enhancer.....	64
Assemble Patterns.....	68
Filter.....	70
Location Mapping.....	73
Splinter.....	77
Content Integration.....	80
Content Aggregation.....	83
Emergency Response.....	86
Distributed Drill-Down.....	89
Communication and Collaboration.....	92
Manage Patterns.....	95
Content Migration.....	96

Dashboard.....	100
Portal Enabler.....	105
Quick Proof-of-Concept.....	107
Single Sign-on.....	109
Widget Enabler.....	113
Conclusiones.....	117
Bibliografía y Referencias.....	119

## - Capítulo 1 -

### **Introducción**

Al comienzo el proyecto WWW empezó con el fin de permitir a los físicos compartir datos, noticias y documentación. La Web consistía en un sitio solitario y casi nadie que no perteneciera a un laboratorio universitario o de investigación tuvo la posibilidad de utilizar el primer navegador web. La cantidad de contenido no creció mucho hasta dos años más tarde cuando ocurrió el primero de varios Big Bangs. En 1993, se libero el primer programa de PC capaz de navegar por la Web con la intención de poner la web al alcance de un público mucho más amplio. Aun así, la conectividad a Internet seguía siendo en gran parte restringida a las universidades, institutos de investigación y empresas. Los consumidores disfrutaban de las comunidades en línea, en general a través de servicios pre-empaquetados como CompuServe, Prodigy y América Online (AOL). La conectividad se lograba a través de conexiones lentas de acceso telefónico. En 1994, comenzaron a aparecer los primeros proveedores de servicios independientes (ISP). Mediante la instalación de un software especial en sus equipos, los consumidores pudieron acceder a todo el contenido de la Web. AOL comenzó a abrir el acceso Web para sus millones de suscriptores. Al poco tiempo surgieron los editores HTML facilitando la creación de páginas Web. En respuesta, se produjo la segunda explosión Web en crecimiento. Ya en 1996, las empresas no vieron la presencia Web como un lujo, sino como una necesidad.

Impulsado por planes de negocio incorrecto y especulación salvaje, se llevó a cabo la expansión mundial de infraestructura de la web. Mientras tanto, el navegador saltó de ordenadores a teléfonos celulares y dispositivos móviles mediante la conectividad de banda ancha. Finalmente la tercera explosión fue el estallido de la burbuja Web, que vio como empresas impresionan en masa. Este evento marcó el final de la primera ola de evolución web, que en retrospectiva, se etiqueta como Web 1.0.

### **Web 2.0**

El término Web 2.0 se asocia a las aplicaciones web que facilitan el compartir información, la interoperabilidad, el diseño centrado en el usuario [1] y la colaboración en la World Wide Web. Un sitio Web 2.0 permite a los usuarios interactuar y colaborar entre sí como creadores de contenido generado por usuarios en una comunidad virtual, a diferencia de sitios web donde los usuarios se limitan a la observación pasiva de los contenidos que se ha creado para ellos. Ejemplos de la Web 2.0 son las comunidades web, servicios web, aplicaciones web, servicios de redes sociales, servicios de alojamiento de videos, wikis, blogs, Mashups y folksonomías [2].

## El origen de los Mashups

Cuántas veces has visto a un sitio web y has dicho: 'Esto es lo que quería exactamente , aunque ...' Si tan sólo se pudiera combinar estas estadísticas con los datos de las proyecciones de ganancias de mi empresa. Si tan sólo pudiera tomar las direcciones de los restaurantes y colócalos en un mapa. Después de todo, ¿no debería poder combinar toda información digital?

En el 2005 Paul Rademacher hizo ingeniería en reversa de Google Maps para poner las casas y pisos en alquiler de Craigslist en un mapa de su sitio. El sitio esencialmente pre-lanzó la categoría mapa Mashup, porque no había ninguna API de Google Maps en su momento. Incluso ahora, la aplicación de Paul HousingMaps.com es el ejemplo prototípico de Mashup y todavía es utilizado por muchos para encontrar su nuevo hogar. Desde entonces, la API de Google Maps ha pasado a ser el API de uso más popular [4].

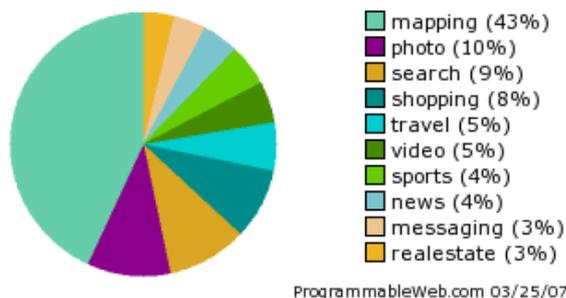
## Que es un Web Mashup?

Un Mashups es una página web o aplicación que usa, combina datos, presentaciones y funcionalidad procedentes de una o más fuentes para crear nuevos servicios. Los Mashups suponen uno de los avances más importantes en el desarrollo web, pues facilitan la combinación de información de una manera sencilla, eficaz y visualmente atractiva. Los Mashups están siendo utilizados tanto para uso público como privado en multitud de combinaciones que facilitan, ordenan y presentan la información de una manera innovadora [3].

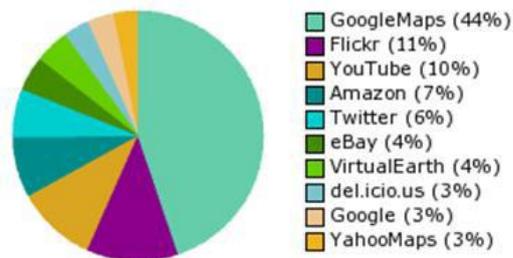
Con las tecnologías Web 2.0 esparciéndose rápidamente a través de internet, los Mashups han traído recientemente mucho la atención como promesa de acercamiento de información ad-hoc e integración de servicios.

Cuando la API de Google Maps se liberó en junio del 2005 hubo una explosión de desarrollo de software basados en ubicación, entre ellas páginas web de clima, transporte, zonas con alto índice de crimen etc.

En el siguiente gráfico se puede ver la distribución aproximada de los Mashups:



Un API es un conjunto de protocolos, rutinas y herramientas para la creación de aplicaciones de software. Una buena API hace que sea más fácil desarrollar un programa proporcionando todos los bloques de construcción. Mediante la combinación de varios tipos de peticiones, se puede utilizar una API para impulsar sus propias aplicaciones personalizadas. A continuación se muestran las siguientes 10 mejores APIs para Mashups:



Desde el surgimiento de los Mashups aparecieron varios tipos de patrones y cada herramienta de Mashups tiene como objetivo apoyar los patrones específicos.

## Patrones de Diseño

La idea de patrones ha sido usada para presentar soluciones generalizadas a problemas que se pueden encontrar cuando se diseñan websites o se escribe software [7].

Los patrones son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que se debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas en distintas circunstancias [5].

## **Objetivo**

El trabajo final propone como objetivo relevar el Estado del Arte de uso de patrones en Mashups y descubrir cuáles son las mejores prácticas en su desarrollo. Con el fin de ofrecer una visión preliminar de la tecnología Mashups para la construcción de sistemas distribuidos orientados al usuario u organizaciones que deseen adoptar una nueva técnica de desarrollo de software.

## **Alcance**

Este trabajo tiene como alcance mostrar el estado actual de los patrones de desarrollo de Web Mashups. Las ventajas y desventajas de la aplicación de cada patrón, como también su aplicación en la vida real.

## Organización del texto

El trabajo final se organiza en 7 capítulos los cuales se presentan a continuación:

**Capítulo 1:** Introducción global, objetivo y alcance del trabajo.

**Capítulo 2:** Organización de los patrones a representar.

**Capítulo 3:** Harvest Patterns: Los patrones Harvest son una clase general de soluciones que se prestan a la obtención de datos procedentes de fuentes fuera del alcance de las herramientas tradicionales. Por ejemplo, la World Wide Web, la cual generalmente no se aprovecha como fuente de datos de primera clase.

**Capítulo 4:** Enhance Patterns: Los Patrones Enhance demuestran cómo los sistemas pueden ser ampliados y mejorados. La perspectiva es que estos cambios se pueden hacer sin la ayuda de los desarrolladores originales.

**Capítulo 5:** Assemble Patterns: Dichos patrones muestran cómo las soluciones pueden ser dificultadas por la combinación de datos y la presentación de múltiples fuentes. Para muchas empresas, el desafío no reside en la construcción de nuevas soluciones, sino en la gestión de las que ya tienen.

**Capítulo 6:** Manage Patterns: Los patrones Manage, tiene como objetivo utilizar los Mashups como un vehículo para ayudar a una organización a aprovechar sus activos existentes de forma más eficaz.

**Capítulo 7:** Conclusiones y reseña de trabajos relacionados, muchos de los cuales sirvieron de base para la elaboración de este trabajo

## - Capítulo 2 -

### Presentación de cada patrón

- Nombre del patrón: nombre estándar del patrón
- Icono. Para facilitar el uso de los patrones en los diagramas de arquitectura.
- Problema. En esta sección se explica la dificultad o la oportunidad que el modelo trata de abordar.
- Solución. Aborda el problema identificado anteriormente para mostrar cómo los Mashups pueden utilizarse para crear una solución. El éxito en la implementación de una solución puede depender del conjunto de herramientas que está utilizando, los sitios que participan en el Mashup y su fragilidad.
- Diagrama. El diagrama proporciona una representación visual de cómo el patrón se construye.
- Fragilidad. Los Mashups contienen elementos de inestabilidad sobre la base de los recursos que aprovechan. El grado de esta fragilidad varía con cada aplicación. En esencia, los Mashups permiten cierta fragilidad para el beneficio de la agilidad. En esta sección se le asigna una calificación de riesgo en general a un patrón particular, la cual se representa con el siguiente gráfico:

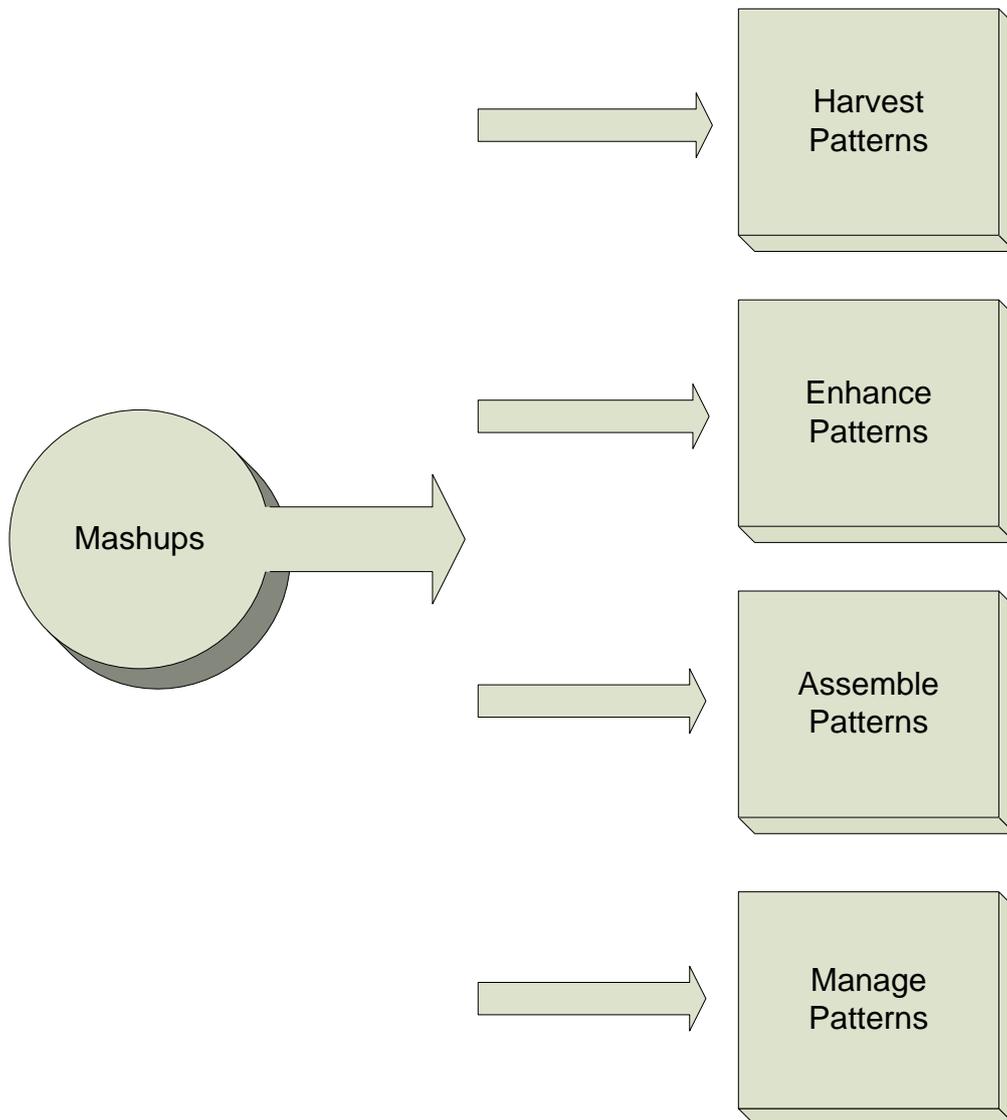
 Menor fragilidad.

 Máxima fragilidad.

- Ejemplos: ejemplo de los usos del patrón en un contexto empresarial.

## Presentación de los patrones

Los patrones de Mashups se organizan en 4 grupos principales los cuales se presentan a continuación:



## Harvest Patterns

Normalmente, los Mashups creados por el usuario se basan en datos y funcionalidad que se hace pública a través de una interfaz definida.

Dentro de la empresa, los datos necesarios para resolver los problemas o para desbloquear oportunidades de negocio se encuentran atrapados dentro de sistemas propietarios. Los Harvest Patterns describen métodos para extraer información de los recursos que son vistos como cerrados. La premisa básica de estos patrones es simple: extraer de los activos los datos importantes.

Dentro de la empresa, el valor potencial de Mashups también es grande, pero los obstáculos para su adopción pueden ser mayores. Si bien es posible exigir la construcción de sistemas abiertos con interfaces documentadas (SOA) estos no ayudan a extraer datos de sistemas cerrados o recursos en Internet [6].

Los 'Harvest Patterns' se utilizan para recuperar datos de fuentes estructuradas y no estructuradas. Fuentes estructuradas pueden consistir en bases de datos, RSS, XML y secuencias de datos tabulares. Fuentes no estructuradas pueden ser sitios Web, archivos binarios (por ejemplo, Excel o PDF), o el texto de forma libre. La clave para convertir la Web en una fuente de información es abandonar las ideas preconcebidas acerca de cómo datos tienen que ser organizados los datos.

## Enhance Patterns

Toda organización ha enfrentado el reto de mejorar sus sistemas existentes para obtener el máximo valor de su inversión. Con demasiada frecuencia, la solución requiere una inversión considerable de tiempo y dinero [7], como también la definición de cuales requerimientos de los usuarios son más importantes y cuáles no.

Si bien estos problemas no eran irresolubles en la empresa antes del surgimiento de los Mashups, los enfoques existentes por lo general requieren recursos técnicos calificados. Los Mashups permiten a los sistemas sean mejorados y actualizados con mucho menos esfuerzo y riesgo.

## Assemble Patterns

Dentro de la empresa moderna, los datos fluyen todos los días a través de la red corporativa en grandes volúmenes. Imagínese si los datos creados por cada sistema estuvieran disponibles para su examen, disección, y recombinación. Trabajar con datos en bruto puede ser un desafío particular para los profesionales de negocio pero no hay razón de que estos recursos deban permanecer ocultos.

El hecho de que los datos de una empresa sean controlados por el departamento de IT no significa que este departamento comprende el valor de los datos de negocio.

Liberar todo el conocimiento almacenado en la empresa puede dar lugar a nuevos descubrimientos y de los productos.

## Manage Patterns

La gestión de la información se encuentra en el núcleo de todos los servicios que provee la tecnología de información. Las aplicaciones de la empresa adquieren información a través de la actividad del usuario o de feeds. La infraestructura de red lleva estos datos entre los servidores, bases de datos y sistemas. En definitiva, la salida se devuelve al usuario en un ciclo sin fin de colección, proceso y presentación.

Los retos de la Gestión de Datos y su relación con Mashups pueden ser separados a través de una serie de temas diferentes:

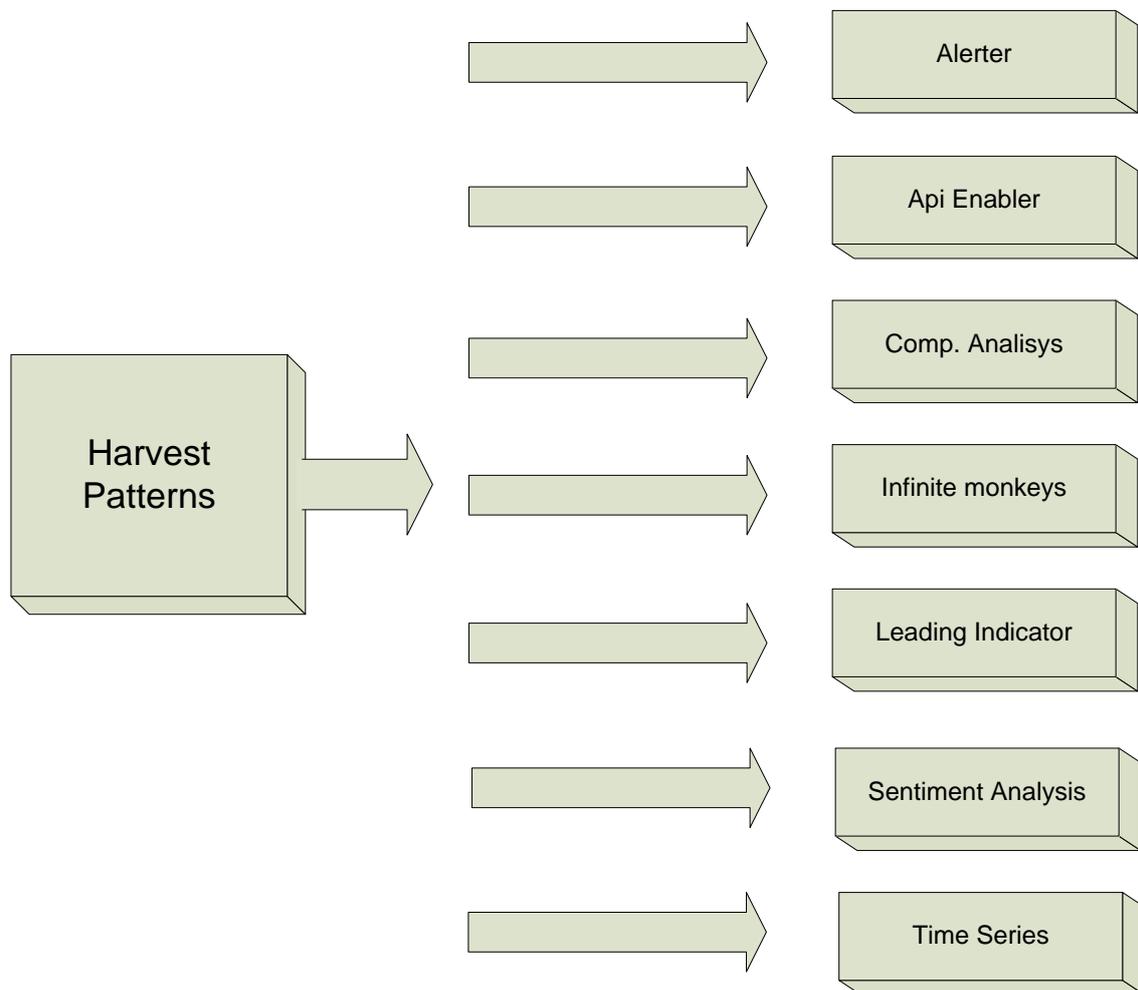
- La transición entre los sistemas. A veces la mejor manera de manejar la información es migrarla a nuevos ambientes para analizarla o procesarla de manera más eficiente.
- Condensación de datos o procesos para alinearlos con los objetivos específicos. La presentación de un usuario con el universo de datos y todas las acciones posibles que se pueden realizar en él es una manera segura de crear confusión y provocar errores involuntarios.
- Asegurar el acceso al conocimiento valioso. Si a un empleado se le presenta información fuera de su responsabilidad, los riesgos son mayores que la mera desorientación.

Otro aspecto que diferencia a estos patrones de sus predecesores es que la mayoría de los patrones de gestión caen bajo la jurisdicción de un departamento de Sistemas.

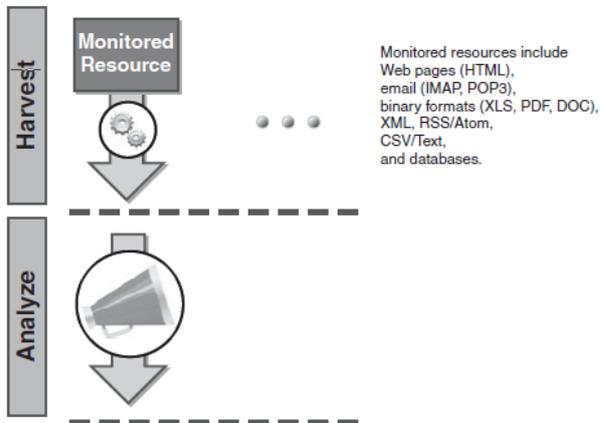
- Capítulo 3 -

# Harvest Patterns

Los 'Harvest Patterns' se subdividen en los siguientes patrones:



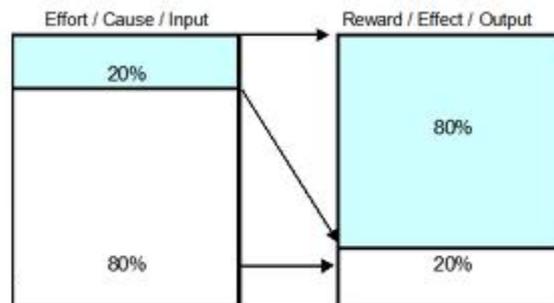
## Alerter



### Problema

El negocio es un objetivo en movimiento. Nuevos productos se liberan, y la gerencia no puede darse el lujo de tomar decisiones basadas en información obsoleta. Mantenerse al día con todos estos cambios es crítico. Por desgracia, el rango de datos que podrían ser controlados es por lo general enorme. Dada la magnitud del problema, las empresas deben limitar las áreas donde centrar su atención.

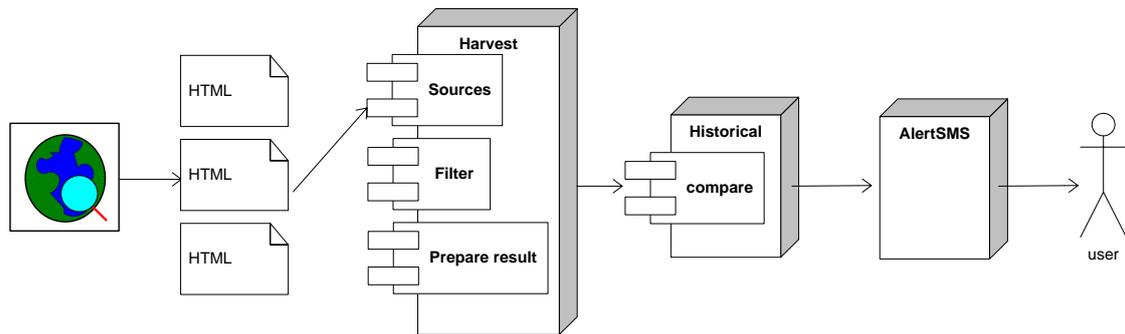
El principio de Pareto indica que, para muchos eventos, aproximadamente el 80% de los efectos provienen del 20% de las causas. [8]



### Solución

Los Mashups no necesitan que interactúen regularmente con un usuario final para agregar valor. Los agentes inteligentes pueden programarse para controlar automáticamente las condiciones y anunciar los cambios por correo electrónico, SMS, u otros canales. Un Alerter en general, sigue las pistas de dos datos puntuales: una lectura actual y la lectura histórica más reciente. El Alerter se puede encadenar con muchos otros patrones para agregar un servicio de vigilancia y notificación.

## Diagrama UML:



**Fragilidad:** 

El patrón Alerter tiene la puntuación más baja la fragilidad. Si el cambio de los recursos controlados es de manera significativa como para inducir un fallo, este fenómeno es esencialmente una alerta en sí mismo.

## Ejemplo

No importa el negocio en el que una empresa se especialice, lo más probable es que dependa en gran medida de diferentes versiones de aplicaciones de softwares. Por desgracia, la cantidad de productos en uso en una empresa puede contarse por miles, demasiados para que el personal compruebe periódicamente las actualizaciones.

En este ejemplo, se verifican nuevas versiones de chrome, informando los cambios de dicha versión e informando el link para la descarga de dicho explorador. Dicha respuesta puede alimentar a un webService interno, el cual puede notificar vía mail a las personas necesarias

<http://pipes.yahoo.com/pipes/pipe.edit?id=a1bcfb2ab67263121f633d56250bd207>

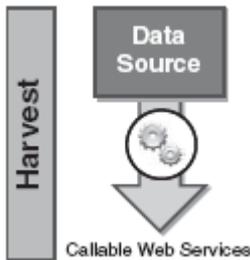
The screenshot shows a Yahoo! Pipes workflow with the following components:

- Find First Site Feed:** URL: `http://googlechromereleases.bl`
- Filter:** Permit: items that match all of the following. Rules:
  - item.title: Contains Beta Channel Upd
  - item.description: Contains download
  - item.pubDate: is greater than 2012-05-08
- Sub-element:** item.description
- Pipe Output:** (Orange box)

Time taken: 0.256s Refresh

content  
[html | source]  
The Beta channel has been updated to 19.0.1084.46 for Windows, Mac, Linux, and Chrome Frame.  
Take a look at the [changelog](#) to see what happened in this release.  
If you'd like to get on the Beta channel, you can download it from our [Beta download page](#). If you find a new issue, please let u

## API Enabler



### Problema

Los Mashups utilizan contenidos de otras fuentes. A menudo el contenido más valioso está encerrado en formatos propietarios o cerrados, los cuales no exponen un API (interfaz de programación de aplicaciones). Aunque es posible abrir una hoja de cálculo y copiar o pegar sus datos, este procedimiento no facilita la automatización.

### Solución

Hay tres pasos en el proceso del API, que permite la recolección de datos, y la transformación.

-Navegación: El primer paso en la extracción de datos implica la localización. Si la información se encuentra en una página Web, entonces el Mashup que simula ser una persona con un navegador.

-Recolección: Una vez que la información ha sido localizada, el siguiente paso es recuperarla [10].

- Paginas web: Web harvesting: Esta técnica consiste en navegar por la estructura del HTML suministrando y eliminando información.

- RSS/Atom feed: Analizar el texto base del feed.

- Email (IMAP/POP3): La construcción de una capa de WS-proxy que llama al protocolo subyacente.

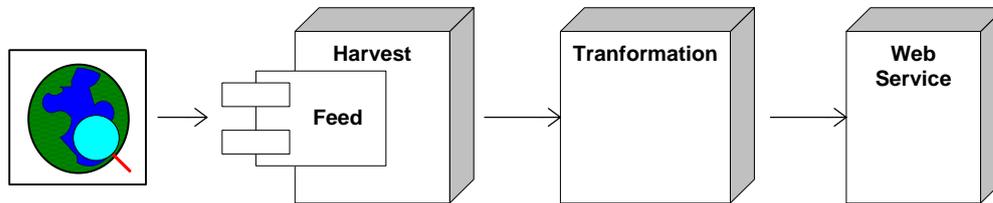
- Formato binario propietario: La herramienta puede incluir la habilidad para leer estos formatos, o de convertirlo a HTML.

- XML, CSV, texto plano, o en otros formatos abiertos: Analizar el texto ya sea que contenga o no estructura discernible.

-Transformación. El último paso en el proceso implica la conversión de los datos a un formato común para su posterior procesamiento.

API Enabler se puede utilizar en una variedad casi infinita de situaciones para imponer una interfaz de datos estructurados o ad hoc.

## Diagrama UML:



## Fragilidad:

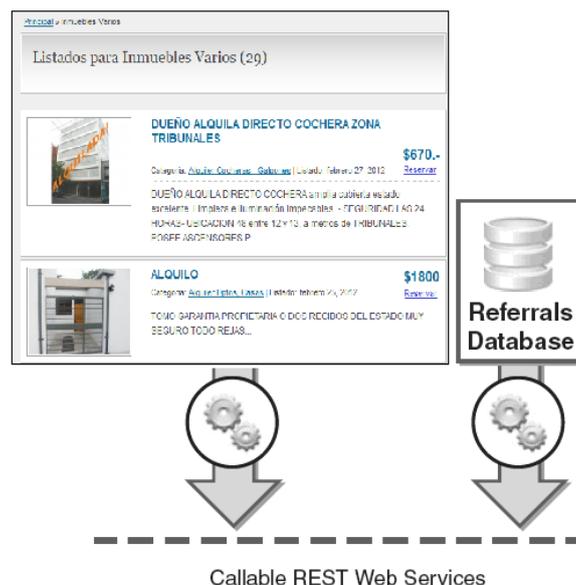
Si bien los cambios de aspecto o comportamiento de un sitio web pueden romper un API Enabler, mientras la información no haya sido eliminada o trasladada puede ser relativamente fácil su corrección.

Como este patrón interactúa con una gran variedad de datos, cada uno con su propia vulnerabilidad, es difícil asignar una calificación de fragilidad.

## Ejemplo

InmoTop es una inmobiliaria dedicada al alquiler y venta de departamentos. La inmobiliaria es reconocida por la calidad de precio de sus ventas y alquileres.

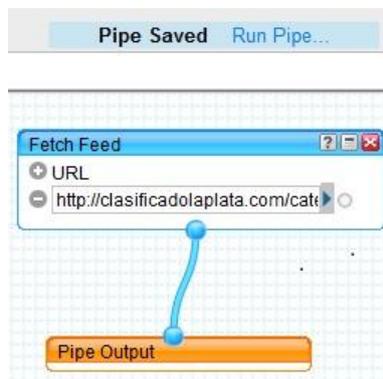
La empresa quiere aprovechar su reserva Web y exponerla como un API donde sitios externos puedan utilizarla para colocar un enlace junto a la descripción de producto. A cambio de este servicio, InmoTop ofrecerá una comisión sobre las reservas para el sitio de referencia.



De la página online de InmoTop se recolecta la información necesaria para crear una API y realizar las siguientes tareas:

- Navegación de un artículo (en forma de un parámetro de entrada)
- Identificar el lugar donde se originó la solicitud
- Automáticamente enviar al cliente a la página de pago.
- Al completar con éxito una transacción, la adición de una comisión para la cuenta de los referentes.

Ejemplo realizado en Yahoo pipes de la extracción de los datos de inmoLaplata y



El resultado del pipe es:

**List** 7 items

**ALQUILO**  
TOMO GARANTIA PROPIETARIA O DOS RECIBOS DEL ESTADO MUY SEGURO TODO REJAS

**ALQUILER TEMPORARIO DEPARTAMENTO BUENOS AIRES**  
ALQUILER TEMPORARIO DEPARTAMENTO BUENOS AIRES EQUIPADO PREMIUM ALQUILER TEMPORARIO – BUENOS AIRES Los dtos. se encuentran estratégicamente ubicados, , sobre la calle Gurruchaga a escasos metros de los mejores Oulet de las primeras marcas de indumentaria y los más variados locales gastronomicos. Con fácil acceso a las zonas bancarias, céntricas, teatrales y de Puerto Madero; [...]

**Excelente Departamento en Santa Teresita**  
Departamento en Planta Baja 2 ambientes + patio en Santa Teresita -Excelente ubicación: 5 e/37 y 38, a 3 cuabras y media del mar y 3 de la peatonal 2 -Capacidad: 4 personas -Comodidades: Living comedor con 2 camas -Habitación matrimonial- Baño con bañera para bebés Placard cómodo- Lavadero techado y cerrado- Patio privado- TV [...]

**Departamento a compartir**  
se busca chica estudiante responsable apra compartir. mas gastos compartidos.

**San Clemente frente a la playa**  
El departamento se encuentra ubicado en inmejorable posición, cercana a todo, frente a la playa, a escasos 40 metros de la calle uno, arteria comercial y peatonal. Se compone de dormitorio con cama matrimonial con colchón de alta densidad, living, cocina comedor, baño, lavadero, balcón terraza con hermosa vista a la playa, balnearios, y plaza [...]

Exportando el pipe obtenemos la siguiente url:

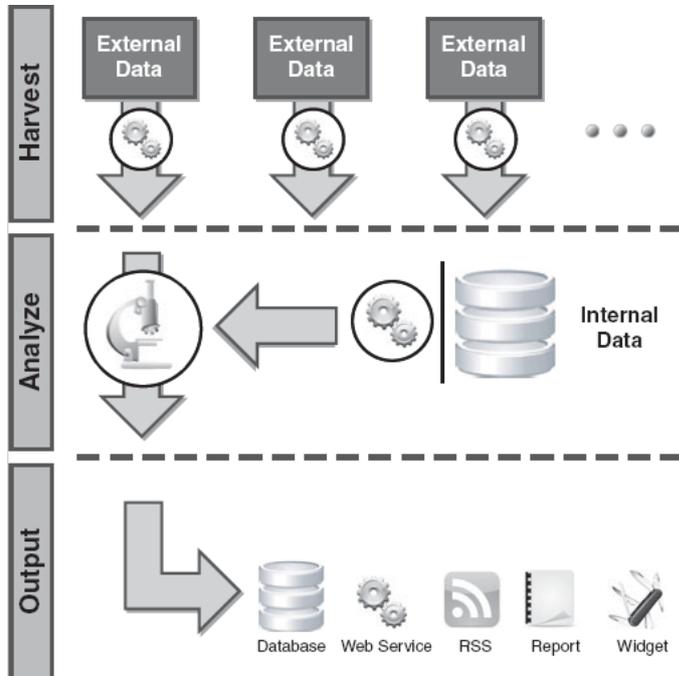
[http://pipes.yahoo.com/pipes/pipe.run?\\_id=bcfdf9b253da51b58b24e7544d3277e8&\\_render=json](http://pipes.yahoo.com/pipes/pipe.run?_id=bcfdf9b253da51b58b24e7544d3277e8&_render=json)

Luego podemos utilizar dicha api mediante javascript, para obtener la información de los departamentos en alquileres.

```
<html><head>
<title></title>
</head>
<body>
<div id="test"></div>
<script>
  function myfunction(o) {
    var o = o.value.items[0];
    var title = o.title;
    var description = o.description;
    .....
  }
</script>
<script
src="http://pipes.yahoo.com/pipes/pipe.run?_id=bcfdf9b253da51b58b24e7544d3277e8&_render=json&_callback=myfunction"></script>
</body>
</html>
```



## Competitive Analysis



### Problema

El Análisis de la competencia es la piedra angular de la formulación y ejecución de estrategias eficaces. El análisis ayuda a los responsables de toma de decisiones para comprender y predecir los cambios críticos en el mercado de acciones las cuales pueden ser tomadas por los competidores y otros agentes [9].

En el mercado global de hoy, es imposible mirar hacia abajo del globo para ver lo que los rivales están cobrando. Los Mashups son una manera excelente de obtener información de una o más fuentes. Además de los factores además de los costos a menudo influyen en la decisión de una empresa: Cuanta publicidad el competidor está haciendo y en donde? ¿Sus productos respetan más el medio ambiente? ¿El rival puede entregar sus productos más rápido? Se necesitan respuestas para desarrollar estrategias de negocio, así como anticipar las tácticas de los oponentes.

### Solución

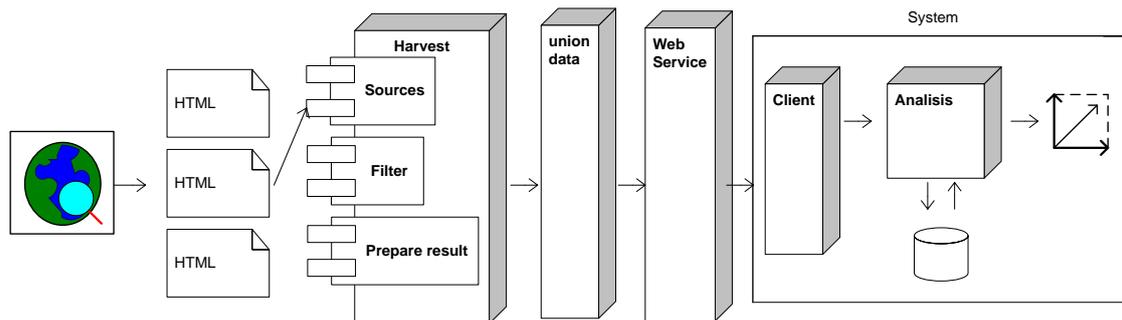
La habilidad de negocios y el análisis de competencia son de vital importancia en la determinación de cómo una empresa puede competir y ofrecer más valor a sus partes interesadas.

La práctica no es ilegal o poco ética, ya que se basa en el examen de material público y entorno operativo de los competidores. La clave para realizar 'Competitive Analysis' es determinar qué información está disponible para ir sistemáticamente a adquirirla.

Un Mashup se puede configurar para descubrir los sitios web competitivos, y dirigirse a sus páginas de precio de productos. Luego de obtener los datos se puede decidir qué datos extraer y representar de una manera que pueda ayudar a la empresa para tomar decisiones estratégicas.

El examen que resulta de estos datos se puede introducir en una base de datos para probar los escenarios alternativos de negocio, generar informes estratégicos, o monitorear las tendencias del mercado.

## Diagrama UML



**Fragilidad:**

La fragilidad de una solución de CA es baja porque los sitios que exponen los datos suelen ser muy estables debido a las siguientes razones:

- Muchos de los documentos sobre las empresas son formularios estandarizados y no está sujeta a cambios frecuentes.
- Los sitios que vigilan enfermedades, tiempo o el clima político se suelen ejecutar por los organismos gubernamentales los cuales no suelen modificar sus interfaces.
- Los catálogos en línea de un competidor pueden ser un medio útil de control de precios, la variedad y cuantía de las existencias.

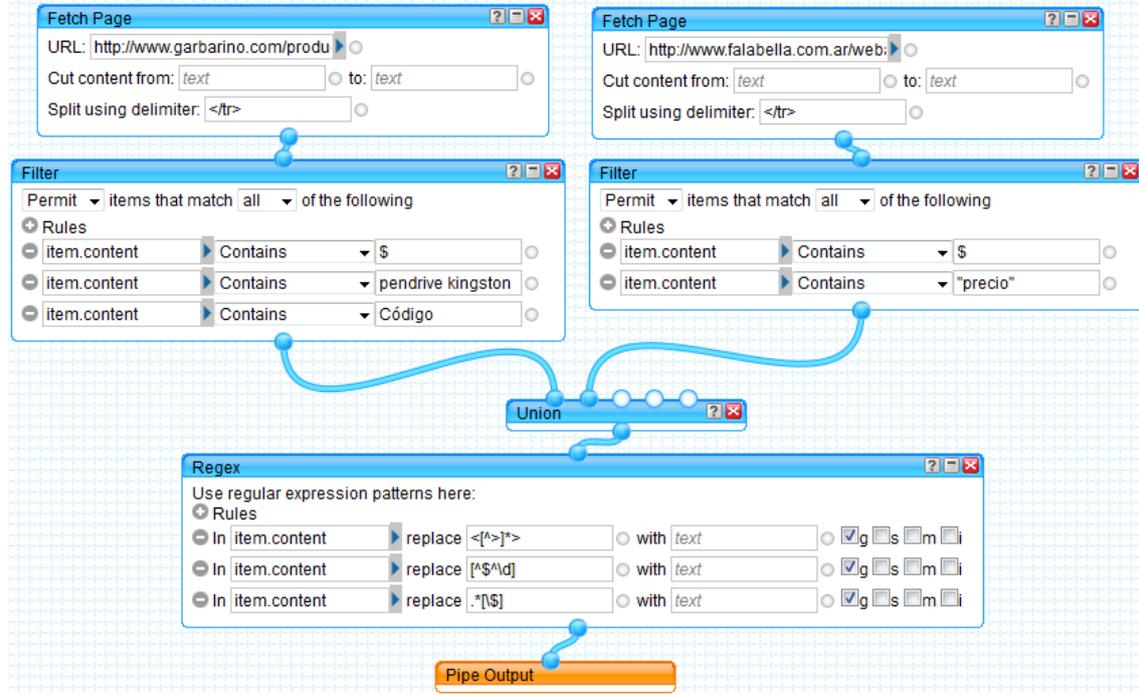
## Ejemplo

HWare es una empresa de suministros de dispositivos de almacenamientos la cual vende pendrives, discos internos y externos. Luego de un estudio HWare ha identificado dos competidores con una importante presencia 'online': Garbarino y Falabela.

Aunque los Directores de HWare tienen la confianza de que pueden competir con el precio, la única manera de saber cuando alguien está cobrando menos es comprobando los productos en los sitios Web de los competidores. Para mantener un mejor control sobre sus competidores, HWare implementa el siguiente Mashup:

Utilizando como sources para el pipe las paginas [www.garbarino.com](http://www.garbarino.com) y [www.falabella.com](http://www.falabella.com) se extrajo la información del precio del producto pendrive Kingston de 8gG de ambas páginas competidoras.

[http://pipes.yahoo.com/pipes/pipe.edit?\\_id=7f838b46ab03a8dfb11ddb64521e09fc](http://pipes.yahoo.com/pipes/pipe.edit?_id=7f838b46ab03a8dfb11ddb64521e09fc)



Obteniendo como resultado del pipe:



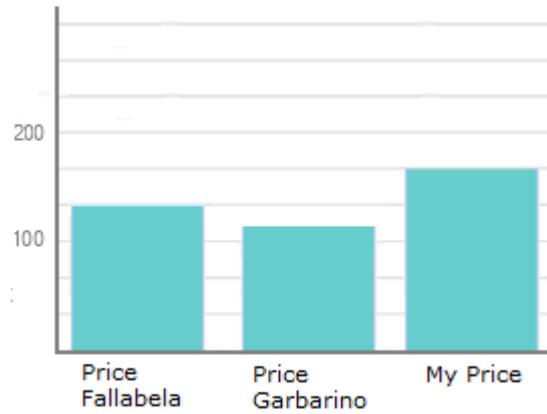
Utilizando javascript se puede utilizar la información obtenida del pipe para crear un grafico de barras.

```

<script type="text/javascript" src="jscharts.js;
http://pipes.yahoo.com/pipes/pipe.run?_id=7f838b46ab03a8dfb11ddb64521e09fc&_render=json"></sc
ript>
    
```

```

<div id="chartcontainer"></div>
<script type="text/javascript">
  var myPrice = 150 // $ precio de la empresa
  var myData = new Array(['one', o.content[0]], ['two', o.content[1]], ['three', myPrice]);
  var myChart = new JSChart('chartcontainer', 'bar');
  myChart.setDataArray(myData);
  myChart.draw();
</script>
    
```



## Usos conocidos

Searched4u es un buscador de palabras clave o productos en Amazon, eBay y Google. Es un motor de búsqueda del sitio y comparador de datos.

### APIs

Amazon S3 +  
eBay

### Creado

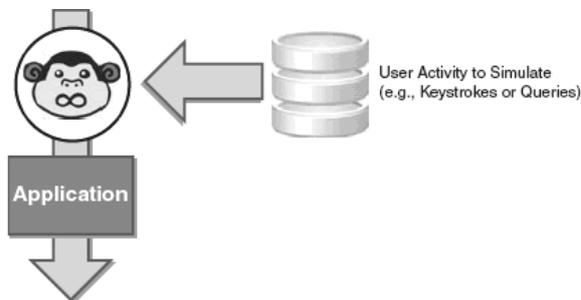
09 Jan 2010

### URL

<http://www.searched4u.com>



## **Infinite Monkeys**



### **Problema**

La web contiene una cantidad inmensa de datos útiles para nuestra empresa. Cuando la cantidad de datos disponibles aumenta en grandes proporciones, puede que el usuario ya no pueda gestionarlos de forma eficaz.

### **Solución**

El teorema de 'Infinite Monkeys' afirma que un mono tecleando una máquina de escribir al azar una cantidad infinita de tiempo casi seguro que escribirá un texto determinado. La premisa es que una tarea mundana con el tiempo puede ser de valor cuando se escala en grandes proporciones.[11]

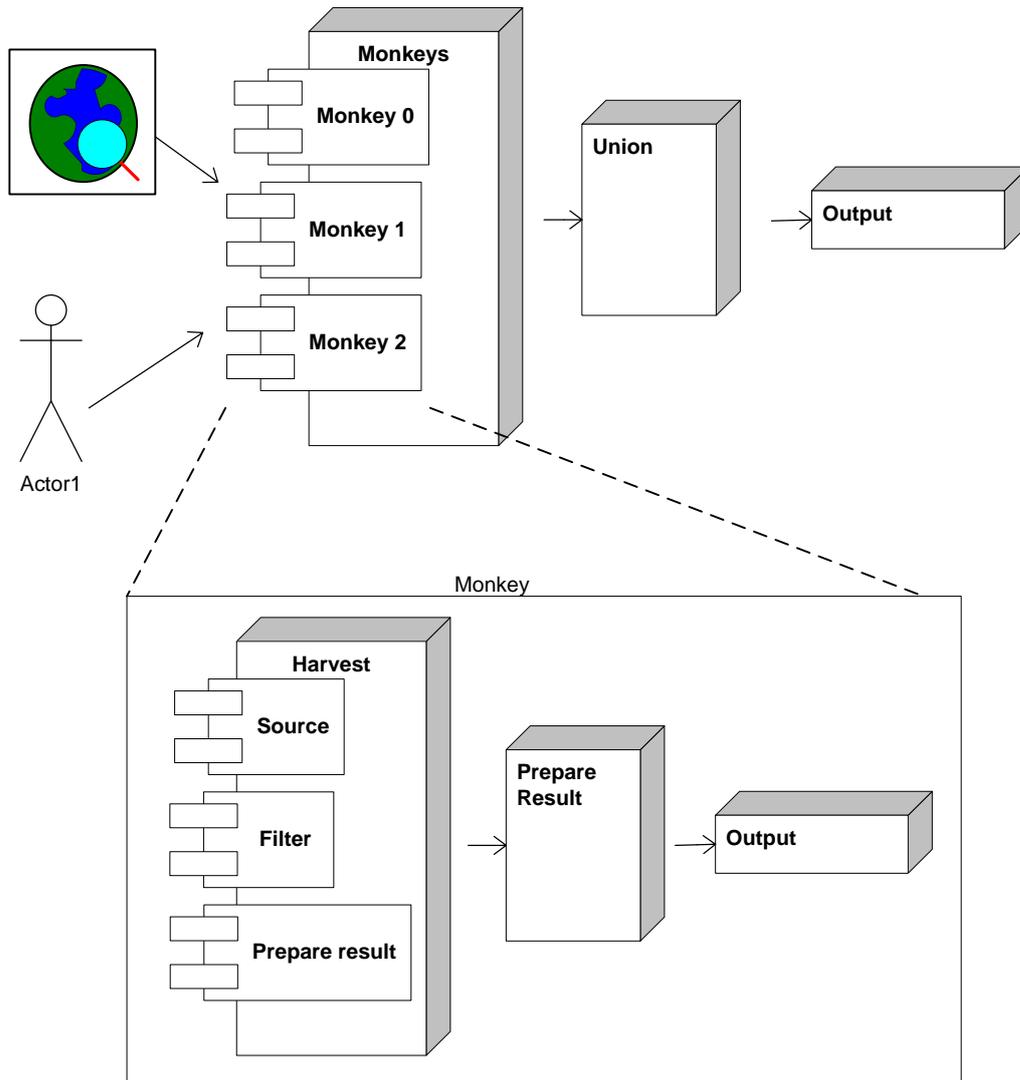
El patrón Infinite Monkeys se basa en las instrucciones relativas a los intereses del usuario y las acciones. Esta información puede ser suministrada a través de una base de datos interna, Mashup, o ambas. La aplicación utiliza estos datos para replicar tareas simples normalmente realizadas por una persona. Este patrón puede ser un elemento eficaz de una estrategia de minería de datos.

La minería de datos es un campo de las ciencias de la computación referido al proceso que intenta descubrir patrones en grandes volúmenes de conjuntos de datos. El objetivo general del proceso de minería de datos consiste en extraer información de un conjunto de datos y transformarla en una estructura comprensible para su uso posterior [12].

El contenido estructurado que una empresa recoge es valioso para la minería, aunque los datos de libre acceso pueden ser aun más valiosos.

'Infinite Monkeys' también puede aplicarse a la tarea de automatización de procesos de negocio (BPA). El BPA es una técnica para reducir los costos y mano de obra, delegando el trabajo manual a un proceso automático.

## Diagrama UML:



**Fragilidad:** 

Este modelo está diseñado para hacer frente a tareas rutinarias que por su naturaleza no cambian a menudo. En consecuencia, los sistemas con los que interactúa este Mashup son típicamente estables.

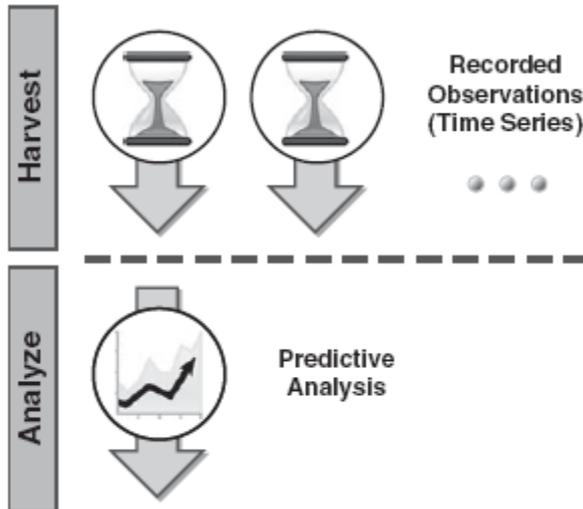
## Ejemplo

InstantSoft es una empresa que se dedica a la contratación de recursos para el desarrollo de aplicaciones web freelance. El sector de evaluación de contratación verifica el perfil de cada posible recurso a contratar dependiendo de su nivel de competitividad en el mercado, por lo que a menudo verifica el perfil del recurso contra las páginas web de desarrollo freelance. Para tener un mejor control y





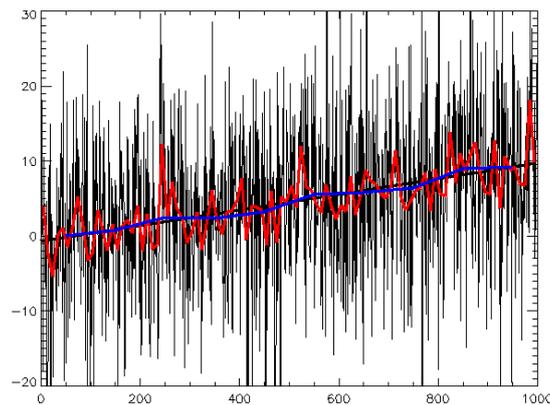
## Leading Indicator



### Problema

El análisis de series de tiempo se refiere a la teoría y la práctica valoración de activos en el tiempo. Es una disciplina muy empírica, pero al igual que la teoría de otros campos científicos es la base para hacer inferencias [13].

El análisis de series de tiempo (TS) comprende métodos para el análisis de datos de series de tiempo con el fin de extraer estadísticas significativas y otras características de los datos. La predicción de series de tiempo es el uso de un modelo para predecir los valores futuros en función de los valores observados anteriormente. Las series temporales son muy frecuentemente representadas mediante gráficos de líneas.



Si bien este es un uso valioso de la información, es totalmente introspectivo. Encontrar las relaciones hasta ahora desconocidas que predicen tendencias y ciclos económicos puede ser extremadamente útil. La clave para descubrir estos enlaces ocultos comienza con la recopilación de datos de 'TS'. Por supuesto, con cada

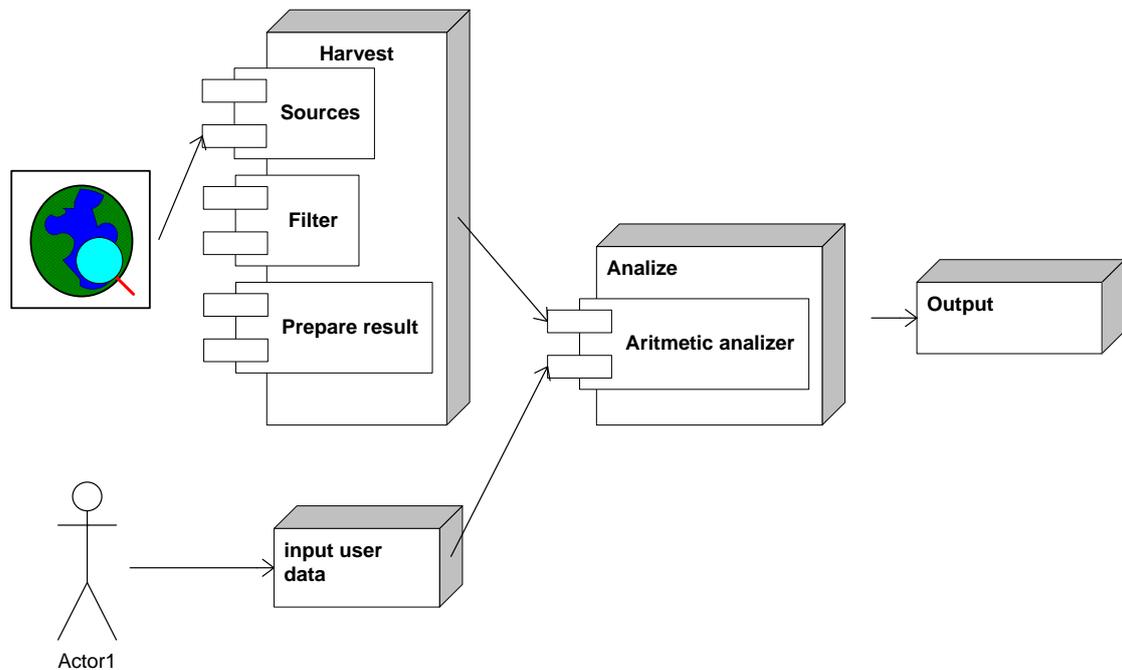
fuentes adicionales que se observa, la complejidad de los procesos de recopilación de datos aumenta.

## Solución

El patrón Leading Indicator (LI) se basa en la recopilación y análisis de datos procedentes de múltiples fuentes de TS para exponer las asociaciones no aparentes. Cuando el patrón LI se implementa como una herramienta de pronóstico, el valor de sus datos es evidente por sí mismo. Por ejemplo, el conocimiento de lo previsto para la semana próxima para la televisión no parece útil para una cadena de tiendas de mascotas. Pero si se realiza un Mashup contra una base de datos de compras de los clientes, un indicador principal podría surgir entre las exposiciones caninas en la televisión y el aumento de ventas. El minorista tiene ahora un mecanismo para el inventario de avance y planificación de publicidad.

**Patrones Relacionados:** Infinite Monkeys, Time Series

## Diagrama UML:



**Fragilidad:**

La fragilidad del modelo LI depende de la cantidad de colecciones de TS y los sistemas internos utilizados en el Mashup. Aunque los componentes no tienen una fragilidad elevada, el patrón se basa en un número potencialmente grande de estos.

## Ejemplo

El patrón Leading Indicator se puede utilizar para investigación de inversión o con fines competitivos:

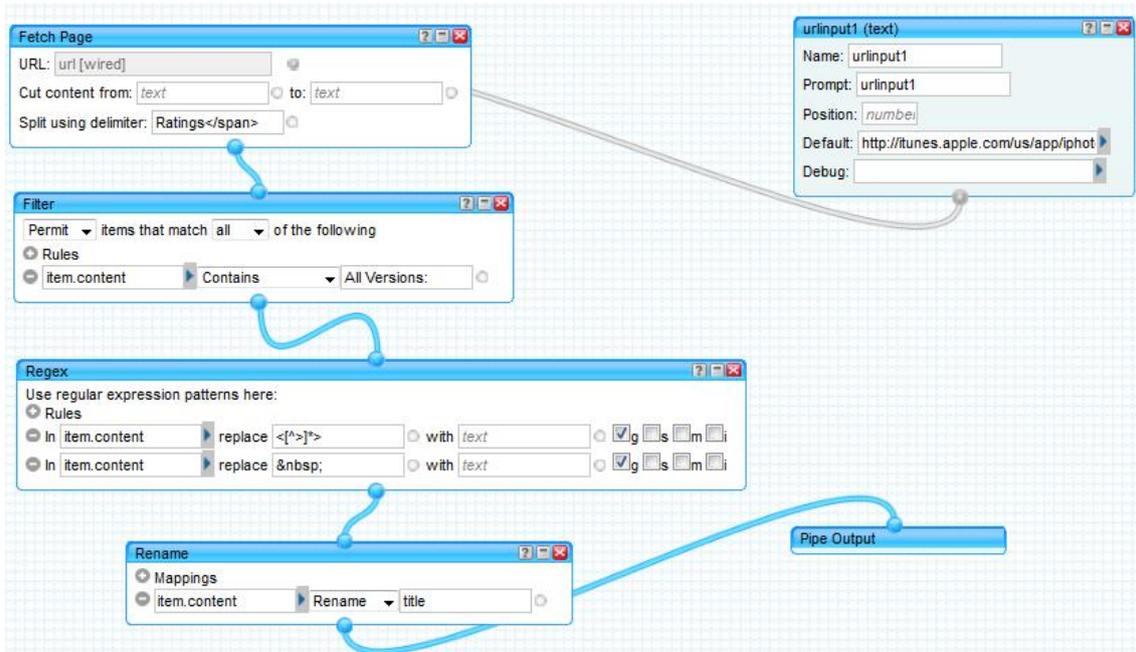


Una pequeña empresa de desarrollo de software verifica las compras de su producto en cada versión liberada. Sus aplicaciones están desarrolladas para iphone/ipad, por lo que utiliza AppleStore para la venta de sus aplicaciones. Como el iphone/ipad permite obtener la cantidad de compras, la empresa quiere realizar una consulta Mashup para verificar la compra por cada nueva versión de sus aplicaciones desarrolladas a fin analizar dicha información posteriormente.

Con esto se puede obtener la siguiente información:

- ¿Qué tan rápido se vende el producto?
- ¿Es necesaria la liberación de un nuevo producto?
- ¿Se venden mejor los productos similares de los competidores?
- ¿Los clientes están satisfechos con sus compras?

[http://pipes.yahoo.com/pipes/pipe.edit?\\_id=d3750694201f56bee9791e9d0c310acb](http://pipes.yahoo.com/pipes/pipe.edit?_id=d3750694201f56bee9791e9d0c310acb)



La aplicación que invoque el Mashup solo tiene que indicarle la pagina donde está la aplicación para su descarga, y el Mashup extraerá las descargas de dicha pagina

Input:

<http://itunes.apple.com/us/app/iphoto/id497786065?mt=8>

## LeadIndicator

*Click to add description*

Pipe Web Address: [http://pipes.yahoo.com/pipes/pipe.info?\\_id=d3750694201f56bee9791e9d0c310acb](http://pipes.yahoo.com/pipes/pipe.info?_id=d3750694201f56bee9791e9d0c310acb) (edit)

☆ Edit Source Delete Publish Clone

Configure this Pipe  
urlinput1  Run Pipe

Use this Pipe  
Get as a Badge MY YAHOO! Google Get as RSS Get as JSON More options

List 1 item  
All Versions: 3373

[Report abusive Pipe](#)

La información obtenida por el Mashup puede ser almacenada en una base de datos para ser posteriormente ser utilizada en reportes de toma de decisión

## Usos conocidos

AnalyticsPortfolio realiza análisis web, interacciones sociales e informes para todos tus sitios en un solo lugar. Compara y analiza las carteras de los sitios web. Se pueden agrupar y ver los totales y promedios

### APIs

Google Analytics +  
Google Chart +  
Google Page Speed Online +  
Postmark +  
RoboWhois +  
SEOMoz

Creado:

04 Nov 2012

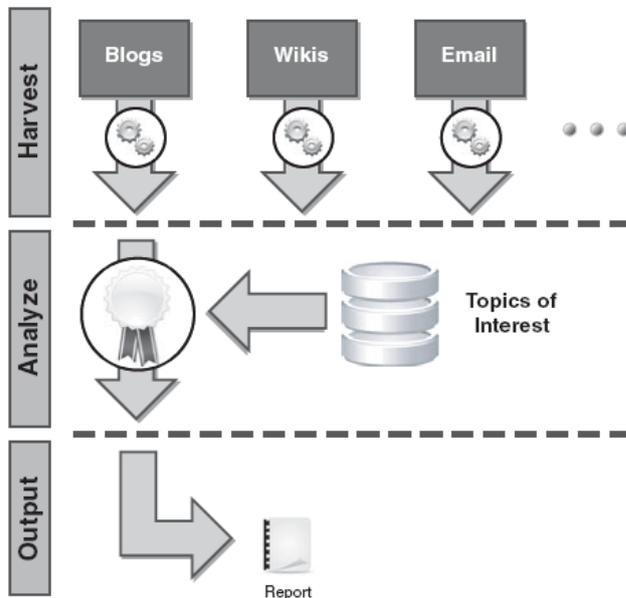
URL

<https://analyticsportfolio.com/>





## Sentiment análisis



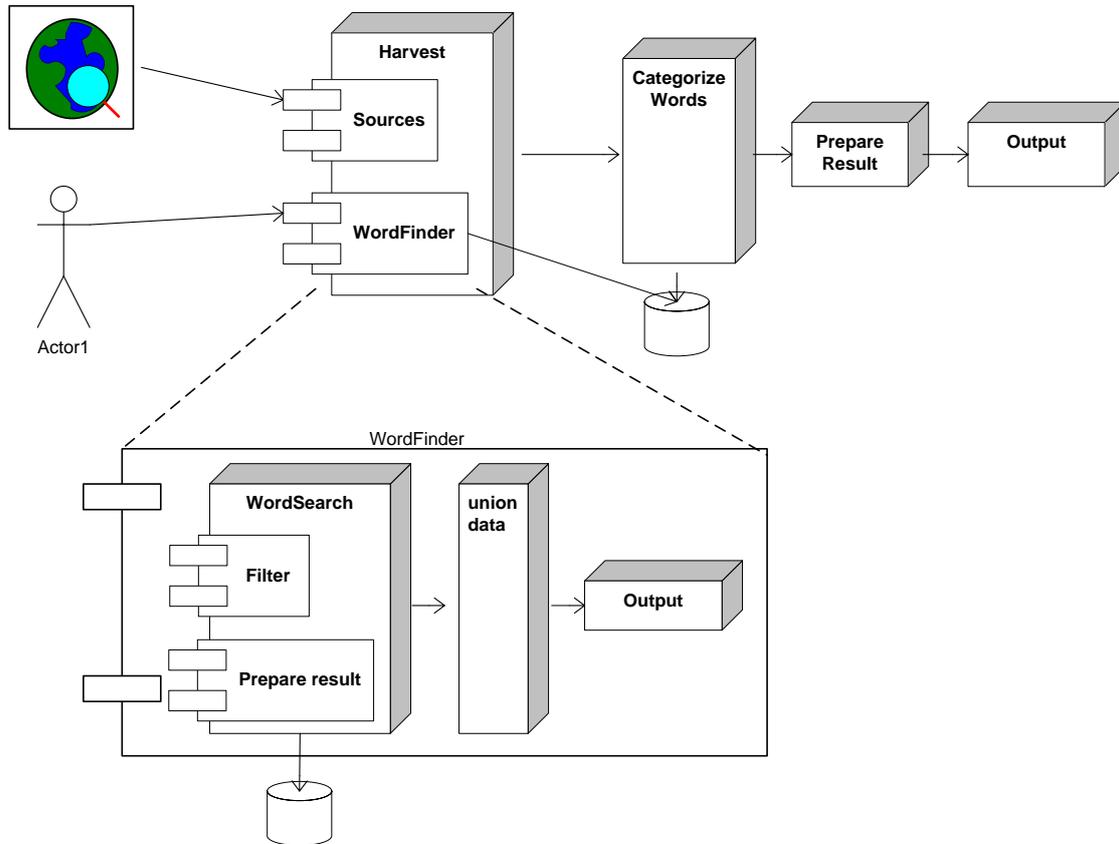
### Problema

'Sentiment Analysis' es el proceso de identificar cómo se percibe la empresa en el mercado y el establecimiento de un plan de acción para corregir, mantener o mejorar dicha reputación. En combinación con una marca fuerte, una buena reputación ayuda a atraer y retener a los clientes. Por el contrario, una mala imagen pública puede ahuyentar a los clientes existentes. Cada acción que realiza una empresa tiene un efecto dominó en su reputación. Los nuevos productos pueden ser recibidos con elogios o con disgusto. La respuesta a un rumor o escándalo puede generar el respeto o indignación.

### Solución

Las técnicas Sentiment Analysis (SA), permite que el texto se analice en busca de palabras que connoten emoción sobre un tema en particular [15]. Al acceder a miles de blogs y publicaciones públicas recogidas por los Mashups, una empresa puede utilizar SA para supervisar su reputación general, así como la opinión pública sobre sus productos.

## Diagrama UML:



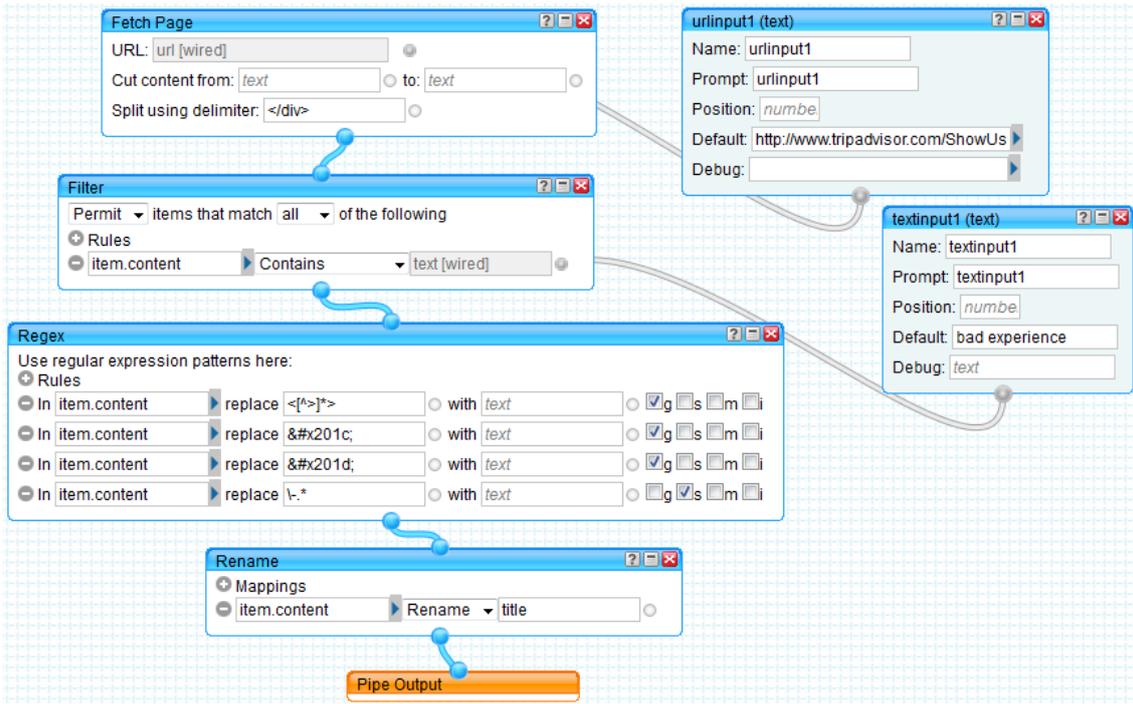
**Fragilidad:**

La calificación de fragilidad de este patrón es baja debido a que el análisis del material aún no es tan flexible como un analista humano (por ejemplo, el sarcasmo puede ser difícil de detectar).

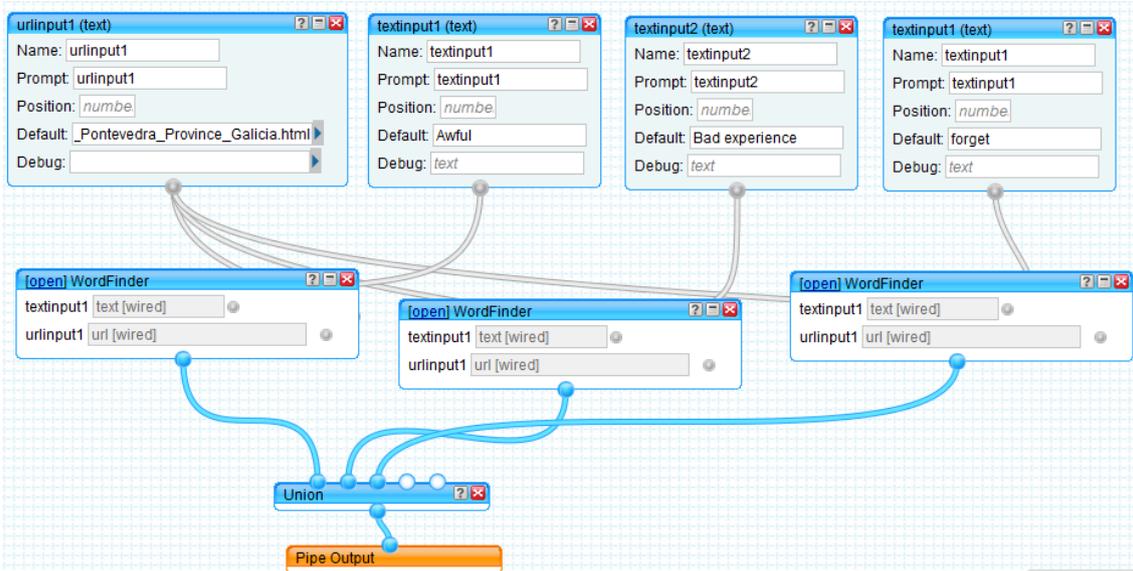
## Ejemplo

El hotel 'Sol Azul' a fin de mejorar su servicio decidió subir la página de su hotel a 'www.tripAdvisor.com', para que sus clientes opinen de sus servicios. A fin de automatizar la búsqueda de las opiniones y poder establecer alertas por si la calidad de su servicio baja decidió realizar un Mashup que permita 'medir' el sentimiento de sus clientes.

A fin de realizar el Mashup se realiza un subpipe el cual es el encargado de encontrar ciertas palabras o frases dentro de los comentarios de la página solicitada.



Luego se crea el pipe principal, el cual es el encargado de buscar una 'n' cantidad de palabras mediante el subpipe y retornar los datos como una única lista con todos los aciertos encontrados



Finalmente indicando las palabras que el Mashup debe buscar y la ubicación de página del hotel en tripAdvisor, el Mashup verifica si esas palabras se encuentran a fin de posteriormente 'medir' la satisfacción del cliente.

textInput3	<input type="text" value="to forget"/>
textInput2	<input type="text" value="Bad experience"/>
textInput1	<input type="text" value="Awful"/>
urlinput1	<input type="text" value="http://www.tripadvisor.com"/> <input type="button" value="Run Pipe"/>

 Get as a Badge  MY YAHOO!  Google+

**List**

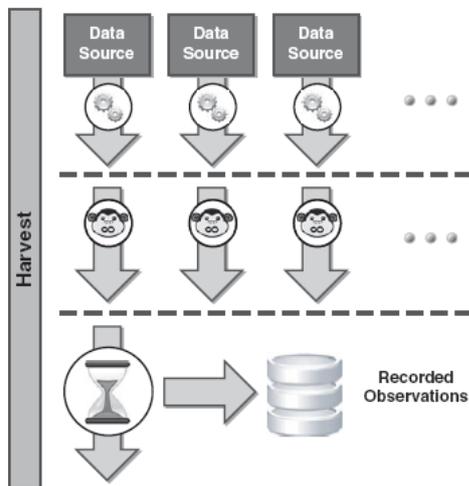
Awful!

Bad experience

Bad experience



## Time Series



### Problema

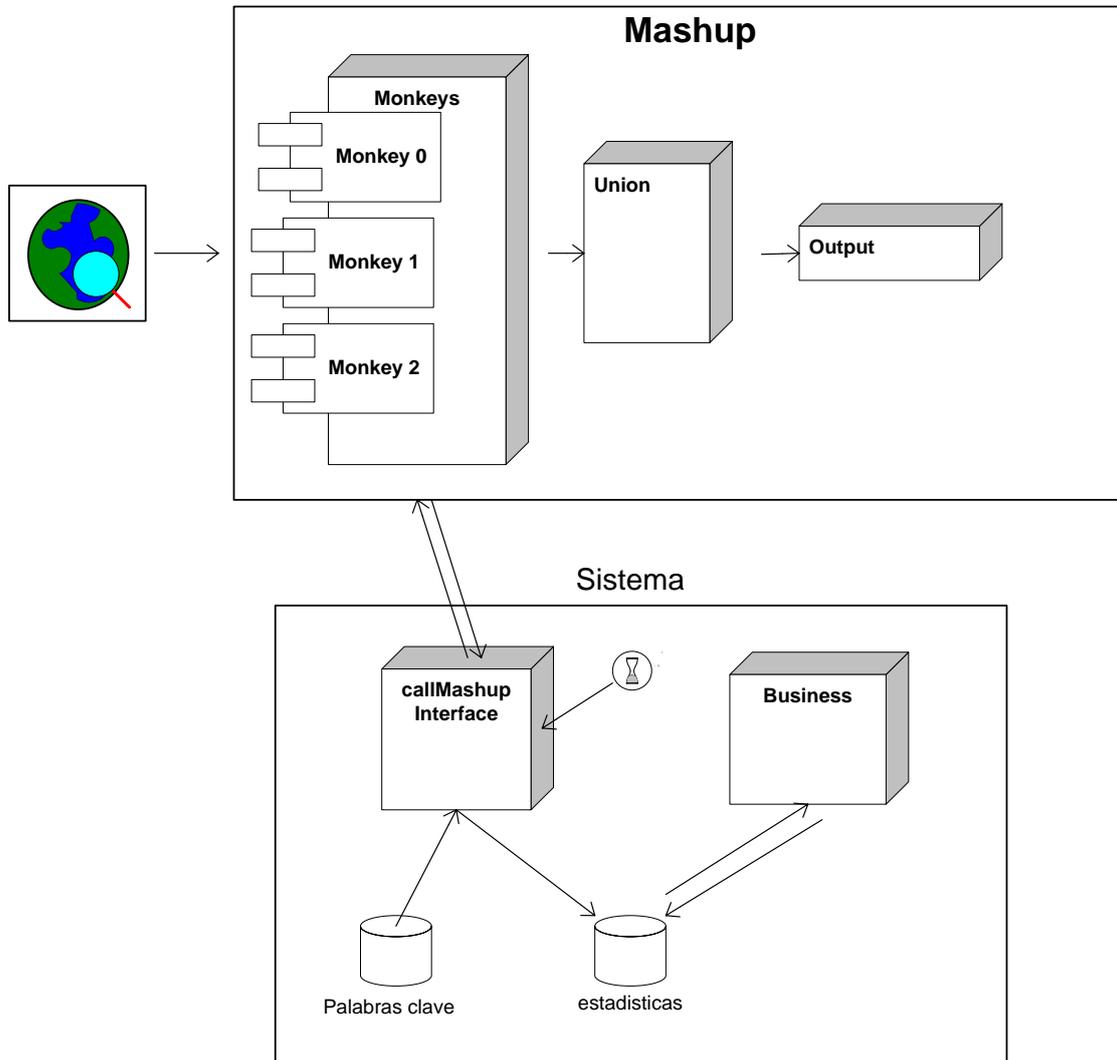
Un 'Time Serie' (TS) es una secuencia de datos observados a través de un intervalo regular. Si se asume una cierta organización interna de los datos, las mediciones pueden ser utilizadas para predecir nuevos puntos en la serie antes de que ocurran. El resultado del análisis de TS es un importante activo en el proceso de toma de decisiones de una corporación. Una empresa exitosa mantiene un ojo en las condiciones actuales del mercado, mientras se posiciona para el futuro.

### Solución

El proceso de recolección de datos TS se puede dividir en cuatro pasos distintos: identificación, recolección, transformación y almacenamiento.

- Identificación. Los recursos que contienen la información deseada se identifican.
- Colección. Como se muestra en el patrón Infinite Monkeys, de recopilación de datos Mashups se puede escalar a un nivel sobrehumano de actividad.
- Transformación. El material recogido puede pasar a través de una transformación funcionar para asegurar un formato común.
- Programación y almacenamiento. El Mashup está programado para ejecutar a intervalos regulares y guardar sus registros. Más tarde, se trabaja con estos datos puede centrarse en la predicción de los valores próximos o tratando de establecer relaciones.

**Diagrama UML:**



**Fragilidad:**

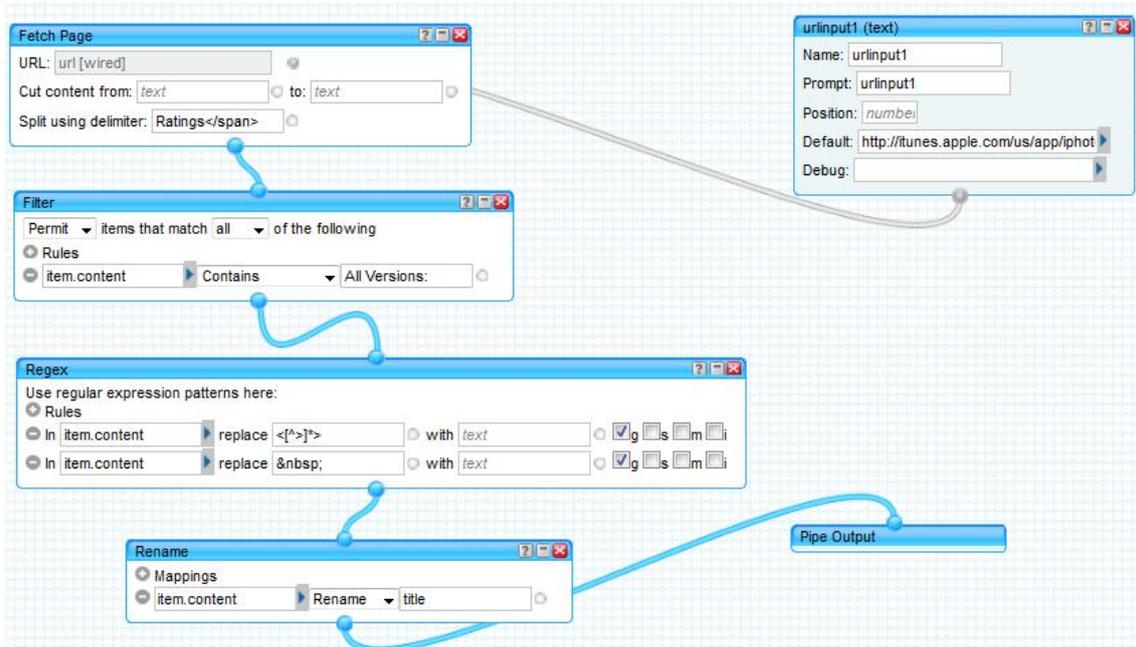
La fragilidad del patrón de Time Series está directamente determinado por la 'API Enabler' y las instancias de 'Infinite Monkey' que lo apoyan.

**Ejemplo**

Volveremos al ejemplo de Mashup anterior Leading Indicator donde empresa de desarrollo de software verifica si los clientes descargan sus productos. Sus aplicaciones están desarrolladas para iphone/ipad, por lo que utiliza AppleStore para la venta de sus aplicaciones. Como el iphone/ipad permite establecer un puntaje de satisfacción con dicha compra, la empresa quiere realizar una consulta Mashup para verificar el puntaje de cada una de sus aplicaciones desarrolladas a fin analizar posteriormente dicha información.

Dicho puntaje puede verificarse diariamente para todas sus aplicaciones, a fin de verificar si los usuarios descargan las aplicaciones y están satisfechos. Comparando la disminución en la puntuación y la rapidez en la que el usuario pierde interés en la compra de la aplicación en cada nueva versión liberada permitirá saber si distribuir los esfuerzos en liberar una nueva versión de la aplicación o crear una aplicación distinta.

[http://pipes.yahoo.com/pipes/pipe.edit?\\_id=d3750694201f56bee9791e9d0c310acb](http://pipes.yahoo.com/pipes/pipe.edit?_id=d3750694201f56bee9791e9d0c310acb)

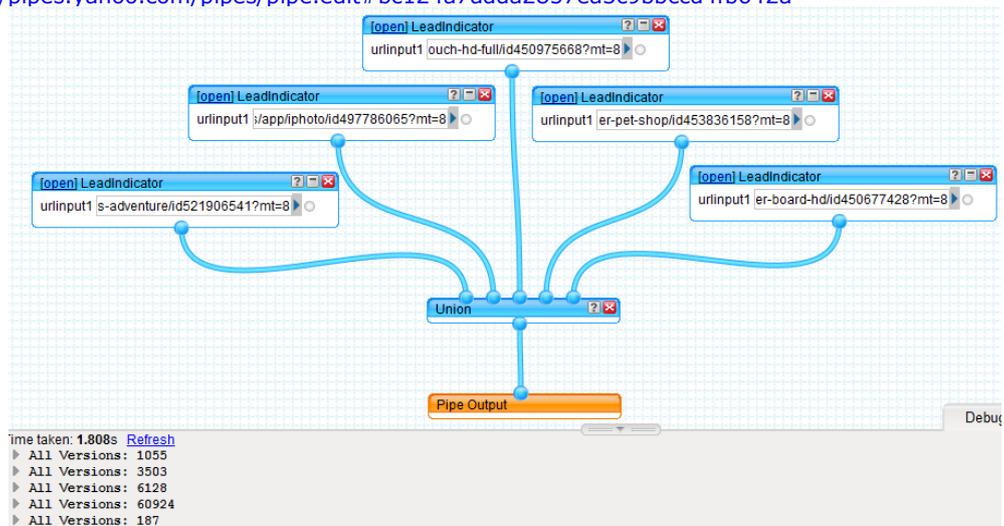


La aplicación que invoque el Mashup solo tiene que indicarle la pagina donde está la aplicación para su descarga, y el Mashup extraerá el puntaje de dicha pagina

Input:

<http://itunes.apple.com/us/app/iphoto/id497786065?mt=8>

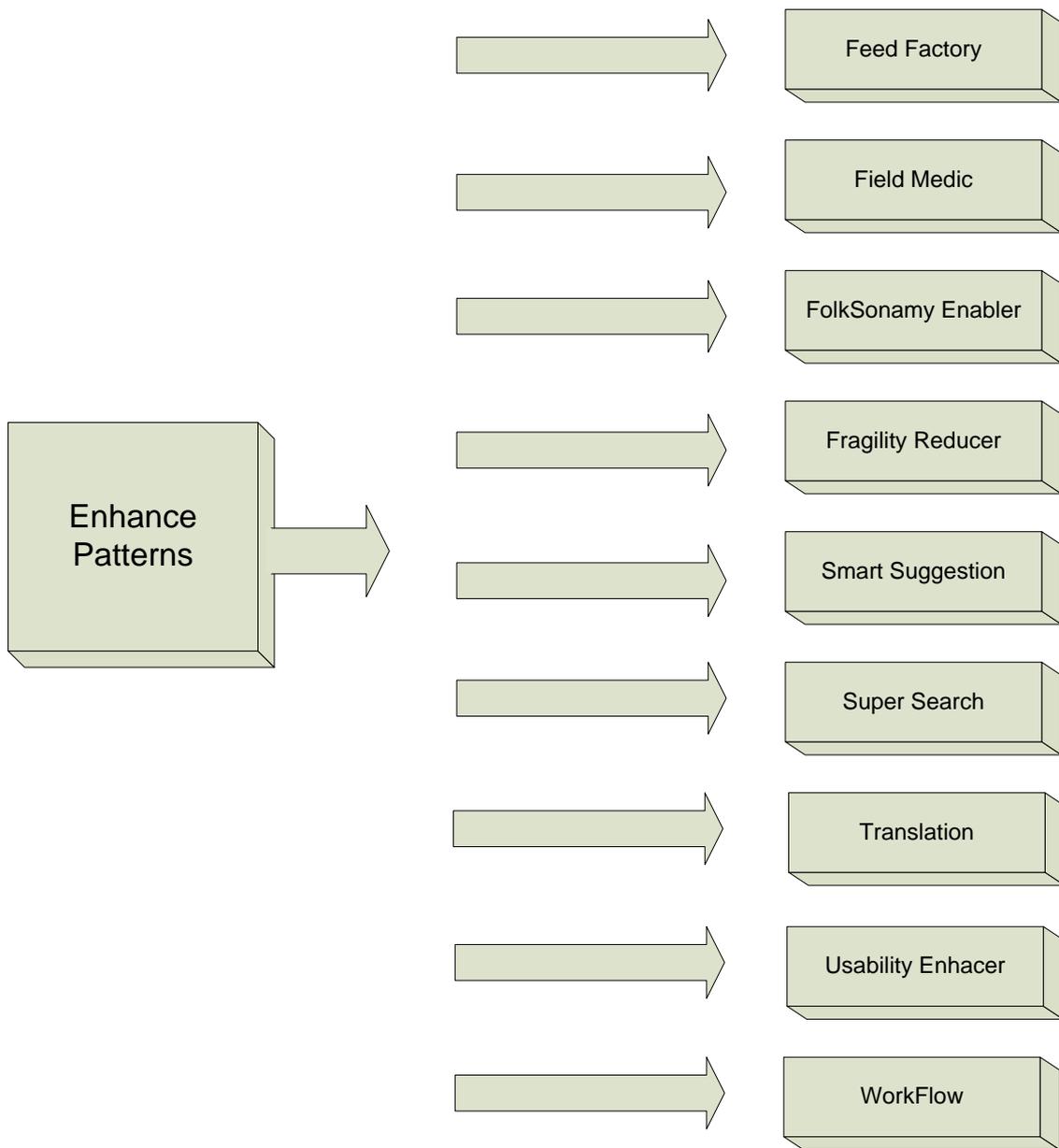
<http://pipes.yahoo.com/pipes/pipe.edit#bc124a7adda2857ea3c9bbccd4fb642a>



La información obtenida por el Mashup puede ser almacenada en una base de datos para posteriormente ser utilizada en reportes de toma de decisión

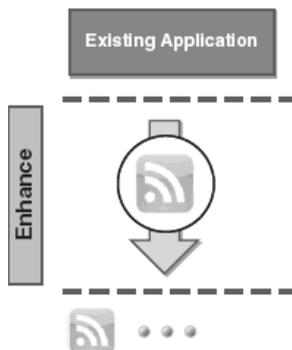
- Capítulo 4 -

# Enhance Patterns





## Feed Factory



### Problema

Mantenerse al día con los cambios de contenido en la Web puede ser una tarea difícil. El enfoque tradicional es el sitio regularmente y comprobar que no haya información nueva. Otra solución es suscribirse a un boletín electrónico con el inconveniente de que las actualizaciones sólo se informan cuando el webmaster decide publicarlas.

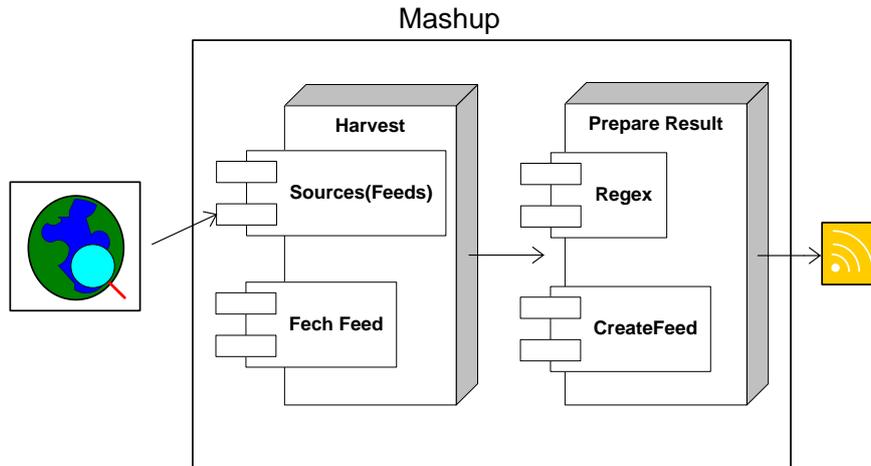
Una manera de publicar el contenido modificado es mediante feeds\*. Los usuarios ven el material desde una única ubicación en un método no diferente a cómo los periódicos presentan los artículos. En lugar de forzar a los usuarios a comprobar actualizaciones o las noticias importantes, esta información se puede publicar como un feed.

### Solución

El modelo API Enabler demuestra cómo funcionalidad de aplicaciones existentes se puede organizar y exponer en una interfaz de desarrolladores. El patrón 'Feed Factory' utiliza la extracción de datos y funciones básicas de entrada para navegar por las páginas de interés y recoger el material para publicar. En el tradicional feed RSS / Atom, el contenido se deriva de una sola fuente. El patrón de 'Content Integration' analiza cómo diferentes fuentes se pueden combinar con los Mashups.

\***RSS Feed** son las siglas de Really Simple Syndication, un formato XML para indicar o compartir contenido en la web. Se utiliza para difundir información actualizada frecuentemente a usuarios que se han suscrito a la fuente de contenidos.

## Diagrama UML:

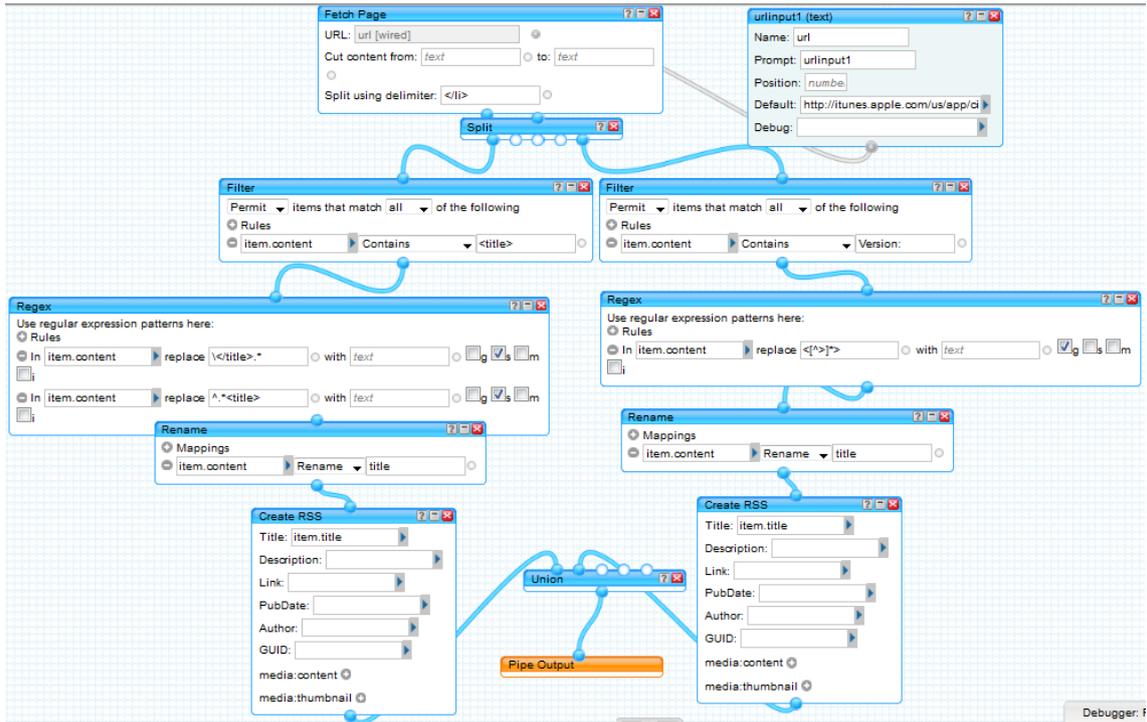


## Fragilidad:

Si bien el 'Feed Factory' es técnicamente similar al patrón 'API Enabler', la diferencia radica en cómo el Mashup devuelve la información. La razón por la que el patrón 'Feed Factory' recibe una puntuación más alta en fragilidad se refiere a que si este falla los resultados saltan a la vista del usuario.

## Ejemplo

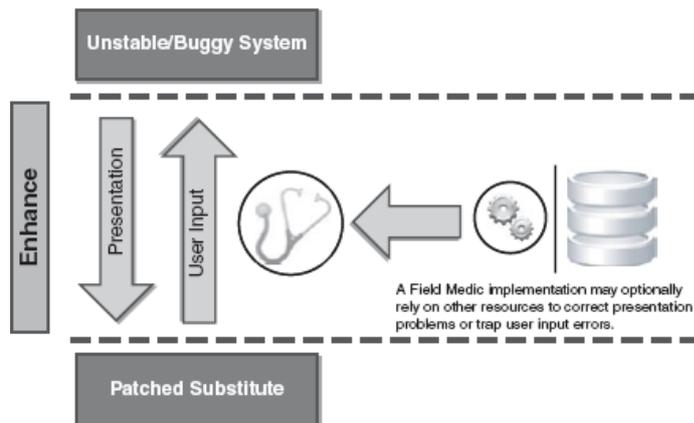
Una empresa que ofrece la posibilidad de conectarse remotamente por celular mediante el producto de un tercero, puede necesitar conocer si surge una nueva versión de dicha aplicación. Como el proveedor no posee un sistema de notificación, la empresa puede crear un Mashup que se conecte a su página de descarga y les notifique cuando hay una nueva versión.



Obteniendo como resultado

Time taken: 0.222s [Refresh](#)  
▶ Citrix Receiver. for iPhone 3GS, iPhone 4, iPhone 4S, iPod touch (3rd generation), iPod touch (4th generation) and iPad on the iTunes App Store  
▷ Version: 5.5.3

## Field Medic



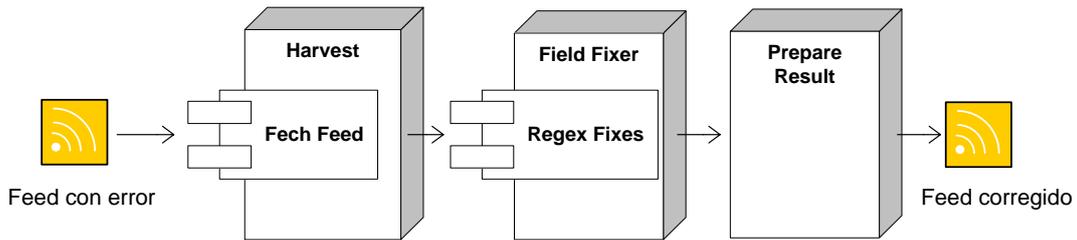
### Problema

Con mucha frecuencia actualizamos y damos nueva funcionalidad a nuestro software. En algunos casos desafortunados el software mal testeado llega a manos de los usuarios finales. Una vez que el software se libera, las limitaciones de la interfaz que impida su uso puede ser o no bloqueante. Algunas cuestiones como molestias menores pueden ser sufridas hasta la liberación de la próxima versión., pero cuando un error es crítico el curso de acción consiste en corregir el error y lanzar un parche. En algunos casos esto no es tan simple. Por ejemplo, si un defecto se descubre en un producto de software de terceros, el cliente debe esperar hasta que el parche sea emitido por el vendedor. [17].

### Solución

El patrón 'Field Medic' puede ser utilizado para añadir temporalmente los controles para capturar la entrada no deseada. Se puede crear un duplicado de la interfaz existente para que actúe como una 'fachada' del sistema existente. La 'fachada' es una envoltura que esconde un sistema existente detrás de una presentación alternativa [5]. El objetivo es replicar la experiencia existente del usuario, pero sin los errores. Si un error se captura en la entrada de datos, la nueva "aplicación" la complementa con el código adicional para atrapar las condiciones de error antes de que alcancen al sistema. La entrada inválida se puede corregir, descartar, o enviada al usuario para que realice cambios.

## Diagrama UML:



**Fragilidad:**

El patrón 'Field Medic' tiene la puntuación de fragilidad más alta posible. Este patrón solo debería utilizarse en circunstancias de suma importancia, y luego retirarse una vez que el proveedor corrigió su software

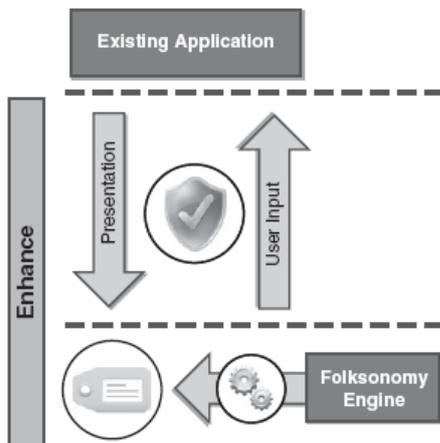
## Ejemplo

Un proveedor puede tardar en liberar nuevas versiones o correcciones de errores. El proveedor puede liberar una versión de un feed con un cambio erróneo del formato de una fecha. Dicho cambio puede resolverse fácilmente con un Mashup sin que sea necesario esperar una nueva versión del sistema del proveedor.

The screenshot shows a Mashup configuration interface with the following components:

- URLforRSS (text):** Name: URLInput, Prompt: URL for RSS, Position: numbe, Default: [empty], Debug: http://qlo.jp/index.xml
- Fetch Feed:** URL, uri [wired]
- Regex:** Use regular expression patterns here:
  - Rules:
    - In item.pubDate replace [a-zA-Z][a-zA-Z][a-z] with text
    - In item.pubDate replace Jan with 01
    - In item.pubDate replace Feb with 02
    - In item.pubDate replace Mar with 03
    - In item.pubDate replace Apr with 04
    - In item.pubDate replace May with 05
    - In item.pubDate replace Jun with 06
    - In item.pubDate replace Jul with 07
    - In item.pubDate replace Aug with 08
    - In item.pubDate replace Sep with 09
    - In item.pubDate replace Oct with 10
    - In item.pubDate replace Nov with 11
    - In item.pubDate replace Dec with 12
    - In item.pubDate replace \s(\d\d:\d\d:\d\d) with TS1
    - In item.pubDate replace (\d\d) (\d\d) (\d\d)\d\d with \$3-\$2-\$1
    - In item.pubDate replace \s+(\d\d)(\d\d)\$ with +\$1:\$2
- Pipe Output:** A yellow box at the bottom right.

## Folksonomy Enabler



### Actividades principales:

#### Problema

Las etiquetas sociales se han convertido en una tendencia actual ya que es una de las mejores maneras de asociar metadatos a objetos Web [19]. Con esta herramienta de organización, los usuarios asocian cierto contenido con un determinado nombre o etiqueta designado por uno o múltiples usuarios. Dichas etiquetas se suman para crear un sistema ad-hoc de clasificación.

Sistema básico de clasificación:



El término 'Folksonomía' es una combinación de palabras (folk: en referencia al aspecto de la comunidad) y 'taxonomy' (refiriéndose a la clasificación) [20]. La práctica de asociar detalles descriptivos (metadatos) con fuentes existentes no es nueva. Lo que hace especial el 'social tagging' es que permite que la sabiduría de la multitud a emerja. La mayoría de las aplicaciones empresariales no fueron diseñadas con este concepto. Los sistemas que carecen de cualquier tipo de reglas de categorización existentes podrían beneficiarse aplicando las capacidades de Folksonomy.

El etiquetado puede ser aplicado a documentos, imágenes, música, vídeos y otros artefactos. Otro de los beneficios de las etiquetas es que se puede suministrar información adicional fuera del contexto de los elementos a los que están unidos.

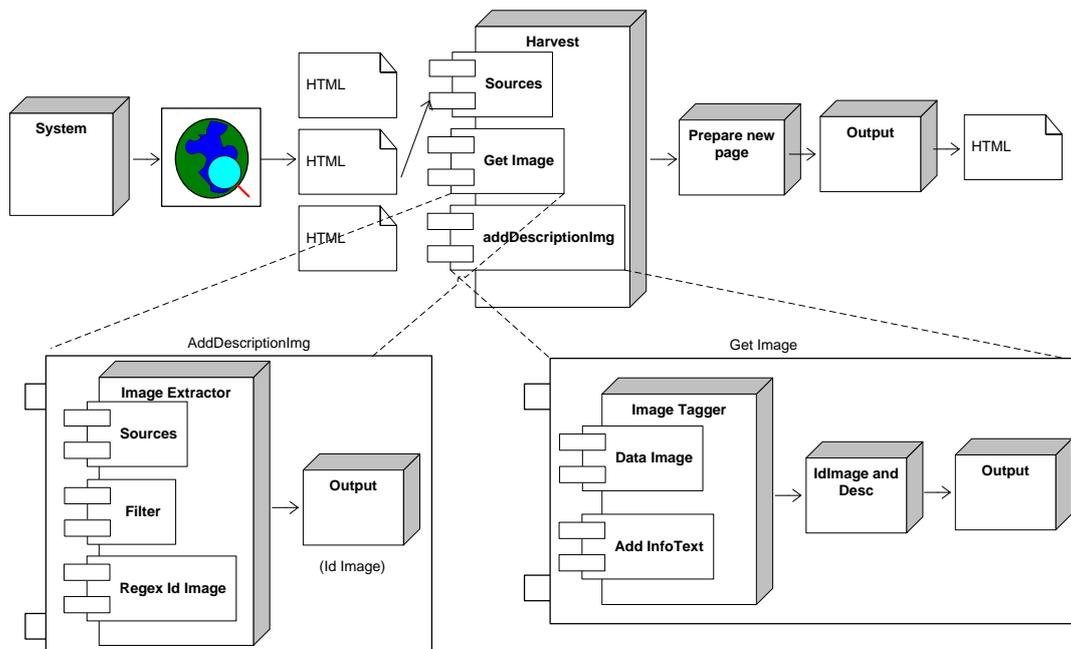
## Solución

Considerando que el sistema no fue diseñado originalmente para soportar 'social tagging', la pregunta obvia es: "¿Cómo sabemos que tags pertenecen a que elementos?" Para asociar tags con los registros existentes, es necesario extraer algún tipo de identificador de la interfaz del sistema de origen para servir como una clave única. Si no hay un código de identificación obvio se tiene que crear uno por la combinación de cualquier otro valor que lo identifique.

El enfoque más simple es ver las etiquetas como otro campo especificado por el usuario similar a "comentar" o "Notas". Tomando este enfoque simplista conduce inevitablemente a un problema. Una vez que el tamaño de las etiquetas de un vocabulario variado crece demasiado esta pierde relevancia. La solución es que la etiqueta debe ser administrada para reducir cualquier desorden potencial. Se sugieren las siguientes técnicas [18]:

- A medida que un usuario escribe un 'tag', se debe buscar a través del vocabulario existente para sugerir palabras apropiadas.
- Establecer convenios de 'tags' como el uso de la forma singular de una palabra en lugar del plural.
- Conectar sinónimos dentro del repositorio de 'tags'.

## Diagrama UML:



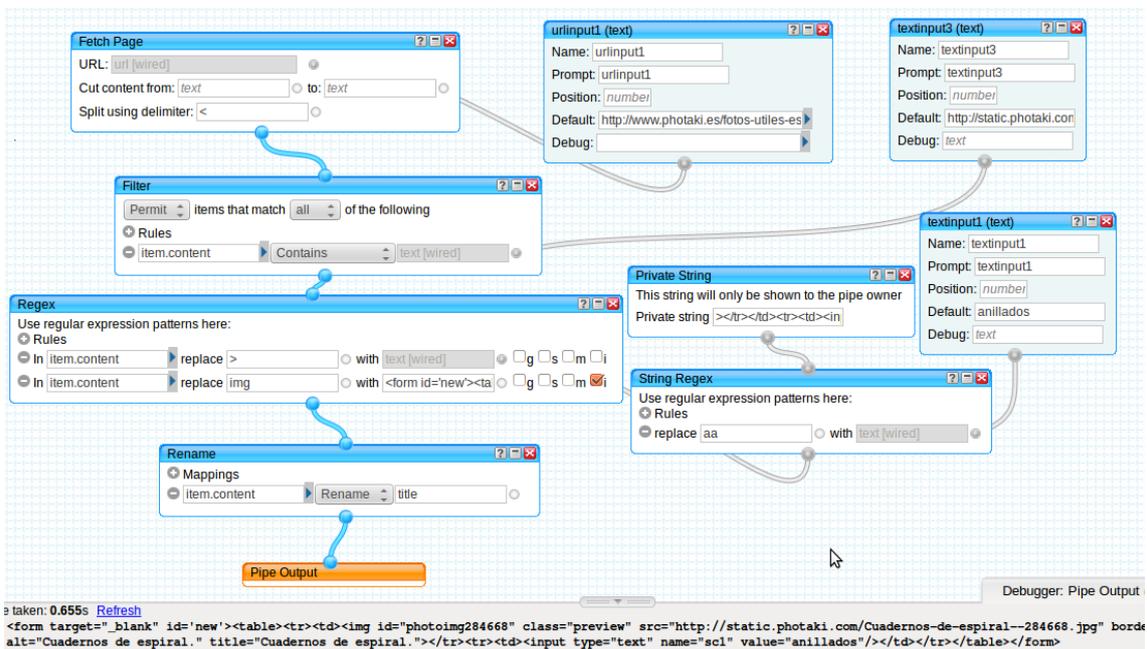
## Fragilidad

Debido a su dependencia del patrón de Usability Enhancer, el patrón 'Folksonomy Enabler' recibe una puntuación de fragilidad igual. Los riesgos inherentes en el 'tag' social en sí mismo son en realidad relativamente bajos. Este modelo describe la adición de tags a un sistema que no fue diseñado originalmente para utilizarlos.

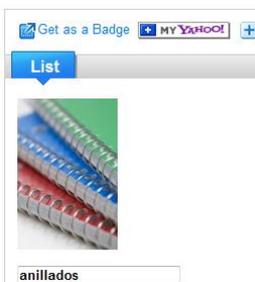
## Ejemplo

En este ejemplo se utiliza yahoo pipes para añadir la funcionalidad de un textbox asociados a las fotos de la pagina <http://www.photaki.es/fotos-utiles-escolares-p1>. Dicho textbox se puede utilizar para añadir información acerca de la foto

<http://pipes.yahoo.com/pipes/pipe.edit?id=72f65be5cc486086ec44ab586767ee79>

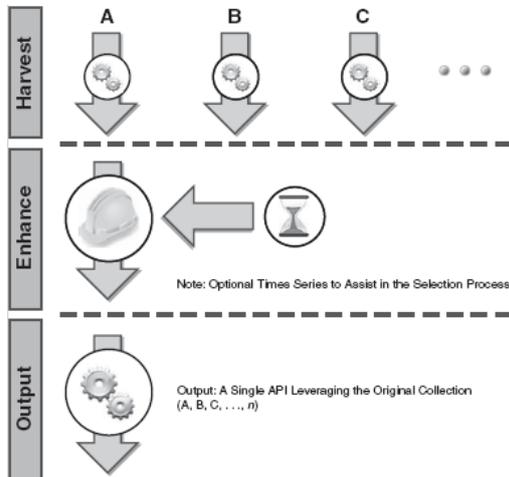


De esta manera creamos el textbox y obtenemos el id de la imagen para finalmente asociarle el valor del tag. Obteniendo como resultado lo siguiente:





## Fragility Reducer



### Problema

Los Mashups están continuamente en riesgo de romperse. Este inconveniente explica por qué cada patrón se presenta acompañado a un índice de fragilidad. Por supuesto, el riesgo real tiene que ser equiparado con los beneficios que estos proporcionan.

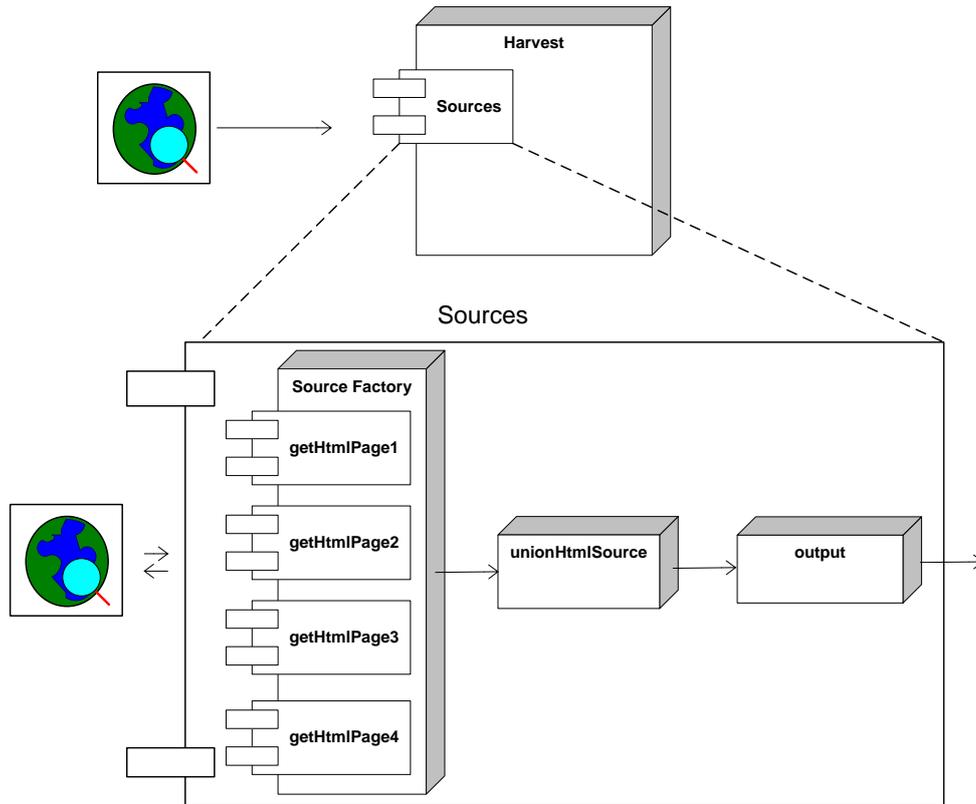
Cuanto más control se tenga sobre los recursos involucrados en un proyecto, más fácil es coordinar los cambios y evitar interrupciones en el servicio. Esta premisa básica es válida para los enfoques tradicionales de desarrollo, así como Mashups.

### Solución

Por desgracia, los Mashups cualquier interrupción en el servicio puede dar lugar a daños consecuentes, por lo que es crucial reducir las posibles averías. El patrón 'Fragility Reducer' aborda esta cuestión al tratar de proporcionar nivel de redundancia interna de las etapas más frágiles de la operación de un Mashup. Es decir, se refiere a los puntos de contacto donde los sistemas externos fuera del control del ejecutor de repente podrían retardar de estar disponibles. Con este patrón, el enfoque preferido es no depender de un único recurso, cuando más un candidato está disponible. Los siguientes beneficios se pueden lograr:

- Estabilidad. Cuando un recurso no está disponible, el Mashup cambia al reemplazo siguiente y sigue trabajando sin interrupciones.
- Rendimiento. El balanceo de carga se puede utilizar la lógica para dirigir el tráfico a los recursos que proporcionan una respuesta más rápida.
- Precisión. Comparando los resultados de múltiples fuentes proporciona un medio para determinar la precisión de los datos.

## Diagrama UML:



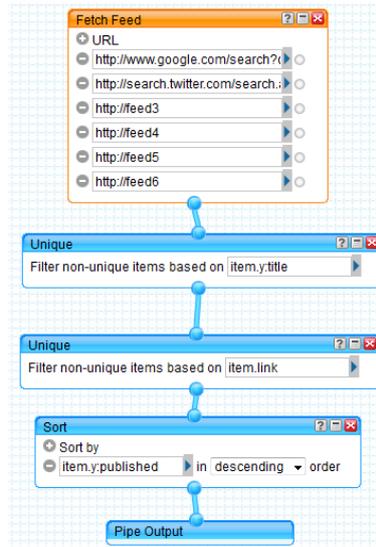
## Fragilidad:



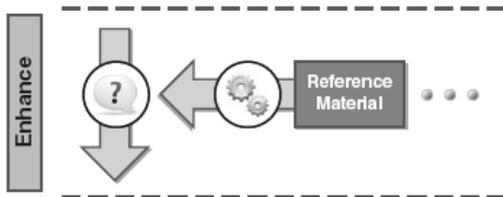
Lamentablemente, 'Fragility Reducer' no puede hacer desaparecer sus propios problemas de fragilidad por completo. Si se consulta varias fuentes para encontrar la respuesta acordada por la mayoría, existe la posibilidad de que los resultados individuales sean tan variados que no haya consenso. Si las fuentes de una colección más grande se utilizan individualmente en forma rotativa, la misma entrada puede dar resultados diferentes en distintos llamados.

## Ejemplo

El patrón 'Fragility Reducer' puede jugar un papel importante en la solución de un Mashup que de repente deja de funcionar. Este enfoque proporciona al investigador con un medio para confirmar o descartar con rapidez el papel del Mashup en el corte de su funcionalidad.



## Smart Suggestions



### Problema

Human-Computer Interaction (HCI) se refiere a la comprensión de cómo la gente hace uso de los dispositivos y sistemas que lleven incorporados o embeban computación, y cómo tales dispositivos y sistemas puede ser más útiles y usables. [42]. El software en general no es "útil" por naturaleza. Para alcanzar el siguiente nivel en usabilidad de software, una aplicación debe ser capaz de predecir lo que el usuario quiere hacer y guiarlo a través de los pasos necesarios para lograr ese objetivo. El fallecido, Clippy fue un notable intento en el software de apoyo.

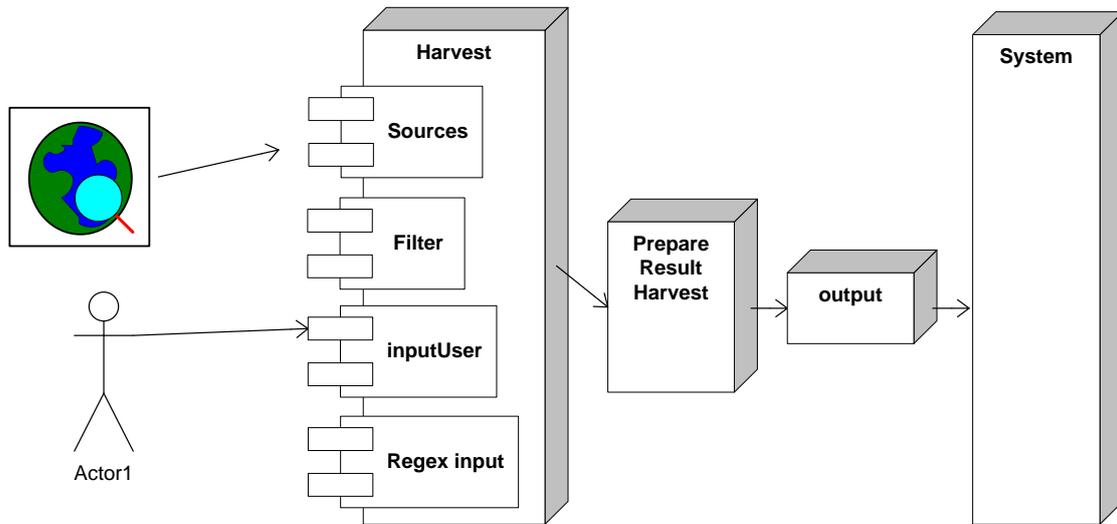
Clippy debutó en Microsoft Office 97. Cuando un usuario iniciaba uno de los programas de la suite de Office, Clippy aparecía automáticamente junto con el área de trabajo actual. A medida que el usuario escribía, Clippy trataba de averiguar qué tareas se estaban realizando ofreciendo sugerencias.

A pesar de su carácter innovador, Clippy fue universalmente 'minimizado' por los usuarios y finalmente retirado. Google Suggest<sup>15</sup> fue lanzado en 2004 como una mejora a la base de búsqueda de Google. En lugar de esperar a que un usuario que introduzca toda la consulta y haga clic en Enviar se realizan búsquedas incrementales de fondo con cada nueva pulsación. Los resultados se muestran en un cuadro desplegable debajo de la entrada del usuario. Google Suggest se hizo tan popular que se incorporó en última instancia, en la página principal de Google. El campo de HCI tiene todavía un largo camino por recorrer antes de que los problemas de usabilidad y productividad estén completamente resueltos [21,42].

### Solución

Un sistema que permite realizar una tarea es de gran ayuda, pero hay una gran diferencia a decir que esta es realmente útil. Un sistema útil reconoce comportamiento específico y se adapta para ayudar al trabajo del usuario tan eficientemente como sea posible. Los Mashups pueden desempeñar un papel importante en la funcionalidad de adición directa de asistencia a un programa. Cuando un usuario proporciona la entrada a un sistema, los Mashups pueden ser utilizados para pasar esta información a otras aplicaciones y servicios.

**Diagrama UML:**



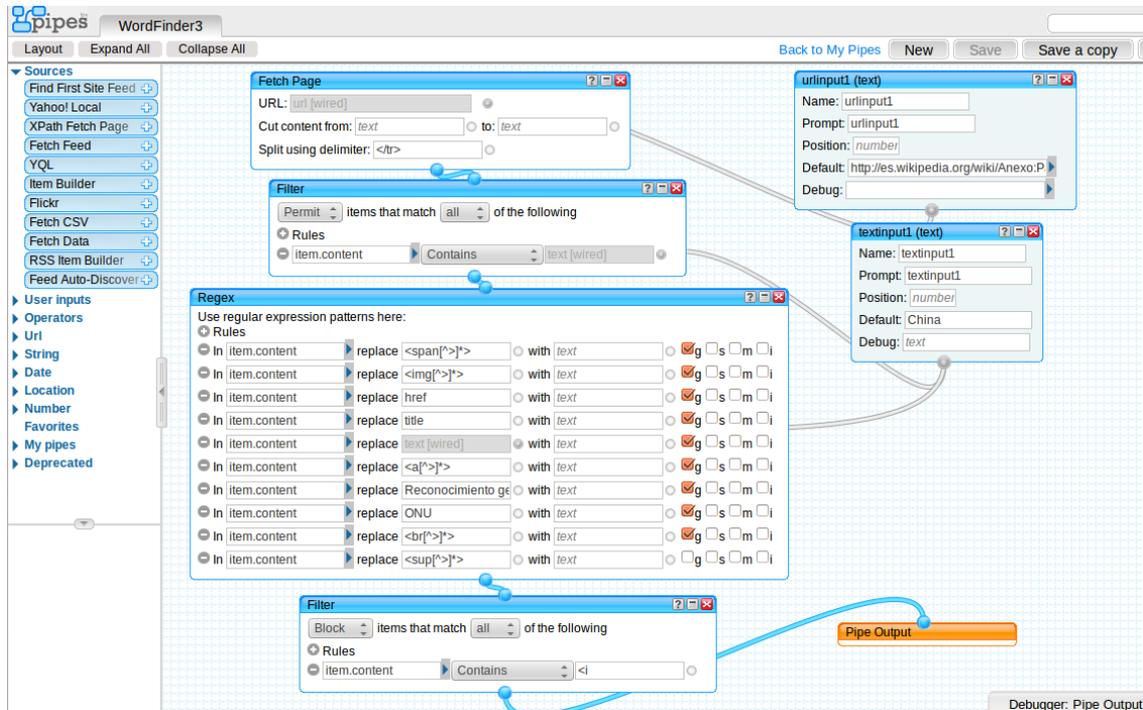
**Fragilidad** 

La principal debilidad del modelo 'Smart Suggestions' no es la falta de asistencia. El problema se produce cuando se administra información incorrecta o irrelevante. Técnicamente esto no es un fracaso de la tecnología Mashup. Tal vez nunca sea posible eliminar por completo las malas sugerencias. Así pues, la clave está en hacer que sea no molesto y aún así accesible.

## Ejemplo

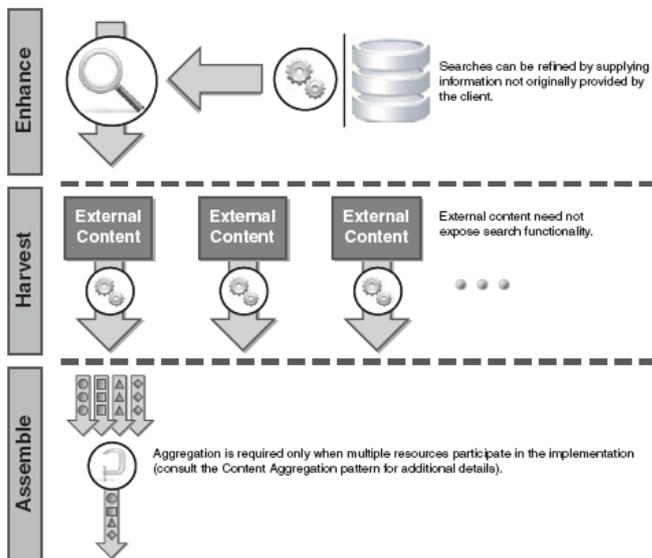
En este ejemplo del patrón 'Smart Suggestion', se utiliza Mashups para sugerir la descripción de un país a partir de su nombre.

[http://pipes.yahoo.com/pipes/pipe.edit?\\_id=c228cc3e3dd7703707be790c0ace39ca](http://pipes.yahoo.com/pipes/pipe.edit?_id=c228cc3e3dd7703707be790c0ace39ca)



El Mashup tiene como entrada la pagina de wikipedia con la lista y descripción de todos los países 'http://es.wikipedia.org/wiki/Anexo:Pa%C3%ADses', y como segunda entrada el nombre del país tipeado por el usuario. Dicho Mashup parséa la pagina filtrando la lista de valores por el valor ingresado por el usuario. Finalmente obtiene la línea seleccionada de la tabla y toma la descripción de dicho país.

## Super Search



### Problema

Los motores de búsqueda como Google, y Yahoo proporcionan servicios de búsqueda para rastrear e indexar grandes porciones de la Web. En una sesión típica, un usuario introduce una serie de palabras en el campo de entrada del motor de búsqueda y recibe una página HTML que contiene una lista de documentos web que coinciden con la consulta del usuario. Sin embargo, frecuentemente surgen situaciones en las que el proceso para la localización de los datos se dificulta:

- Es necesario buscar a través de múltiples sistemas internos.
- La empresa ha adoptado múltiples proveedores y se necesita una función de búsqueda unificada que funcione en todos sus proveedores.
- Se tiene acceso a servicios externos cuya funcionalidad de búsqueda se desea ampliar o incorporar en las aplicaciones internas.
- Si desea utilizar criterios de búsqueda que las herramientas actuales no soporta.
- El producto de búsqueda existente no es muy bueno en la corrección de errores de ortografía o sinónimos.
- Es necesario la búsqueda de varios elementos diferentes al mismo tiempo y traer de vuelta una respuesta única que represente el mejor éxito para todo el conjunto.

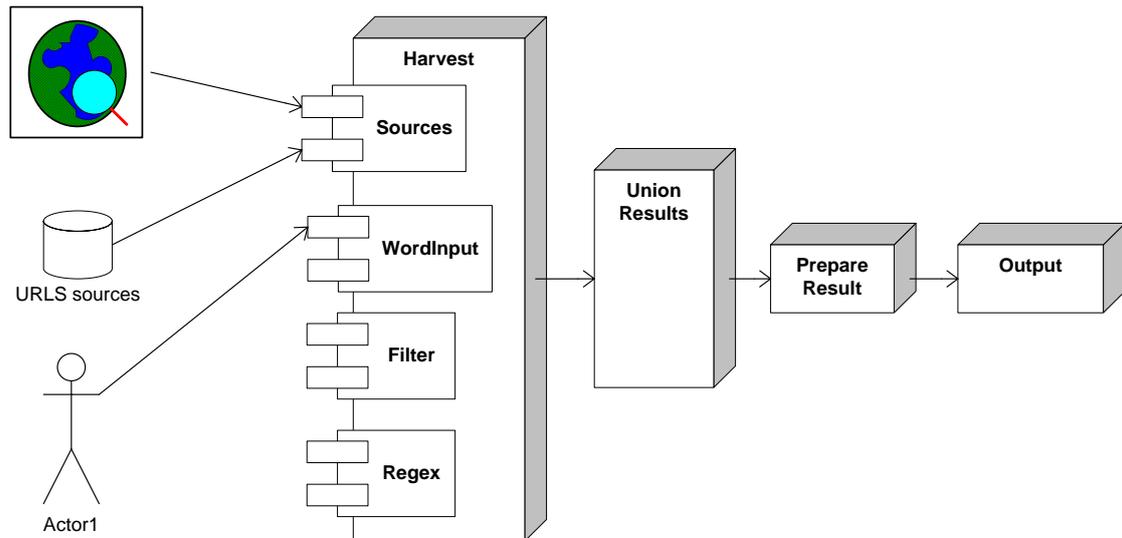
Puede que incluso se quiera añadir capacidades de búsqueda a una aplicación que no tiene ninguna.

### Solución

Un Mashup puede hacer más que simplemente unificar la funcionalidad de búsqueda existente, sino que puede imponer nuevas características de búsqueda

en los recursos existentes. ¿Cómo? Mientras una aplicación proporciona un método para navegar por su contenido, incluso si se trata de un simple botón 'Siguiente', un Mashup pueden aprovechar el patrón de Infinite Monkeys para visitar cada artículo y extraer sus detalles. Todo el proceso puede ser expuesto como un servicio. Estos nuevos servicios pueden superar los beneficios que ofrecen sus partes constituyentes. La utilidad de estas implementaciones pueden impresionar a los usuarios tanto que vean los Mashups como aplicaciones.

## Diagrama UML:



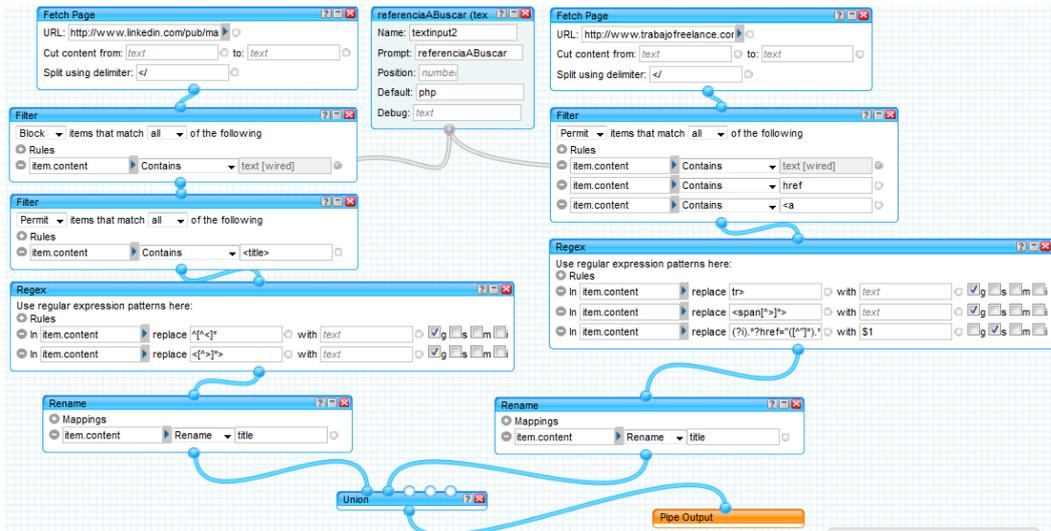
## Fragilidad

Una aplicación de Super Search incluye las interacciones entre varios sistemas, cuantos más sistemas haya involucrados en un Mashup, mayor será la probabilidad de falla potencial. Estos problemas pueden surgir cuando uno de sus sitios se rediseña radicalmente o esta temporalmente fuera de línea.

## Ejemplo

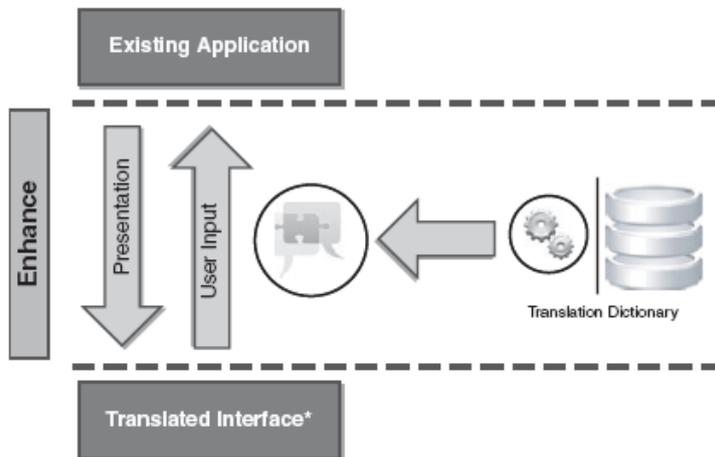
Anteriormente se vio el patrón Mashup 'Infinite Monkeys', el cual buscaba las páginas de trabajo que aplicaban a un posible recurso a contratar. Podemos extender dicho enfoque permitiendo ingresar el parámetro de búsqueda y los distintos 'sources' para la búsqueda. De esta manera podremos invocar al Mashup y obtener como respuesta un listado de links navegables.

# Trabajo Final Mashups Patterns | 2013





## Translation



\* Translation can occur in either or both directions (system to user, or user to system).

### Problema

Muchos de los productos quedan relegados a los usuarios de determinadas regiones o nacionalidades, ya que están aislados por las lenguas de los creadores del producto. Cuando los empleados están aislados por el lenguaje, puede afectar tanto la productividad como a la reputación local de su empresa [23]. Una corporación global puede manejar el reto de la comunicación multilingüe de varias maneras:

- No hacer nada. Cada empleado habla con fluidez un lenguaje estándar así como su lengua local. Muchos empleados y clientes potenciales pueden ser pasados por alto porque no son bilingües.
- Apoyar a múltiples interfaces. Crear una traducción diferente para cada sistema que será inmediatamente familiar a cada público local.
- Proporcionar a la traducción limitada.

En informática, la internacionalización y localización son los medios de adaptación de software a diferentes idiomas, las diferencias regionales y requisitos técnicos de un mercado objetivo. La localización no se limita solo a la lengua. Países diferentes utilizan diferentes formatos de fechas, monedas, signos de puntuación, y mucho más (22).

### Solución

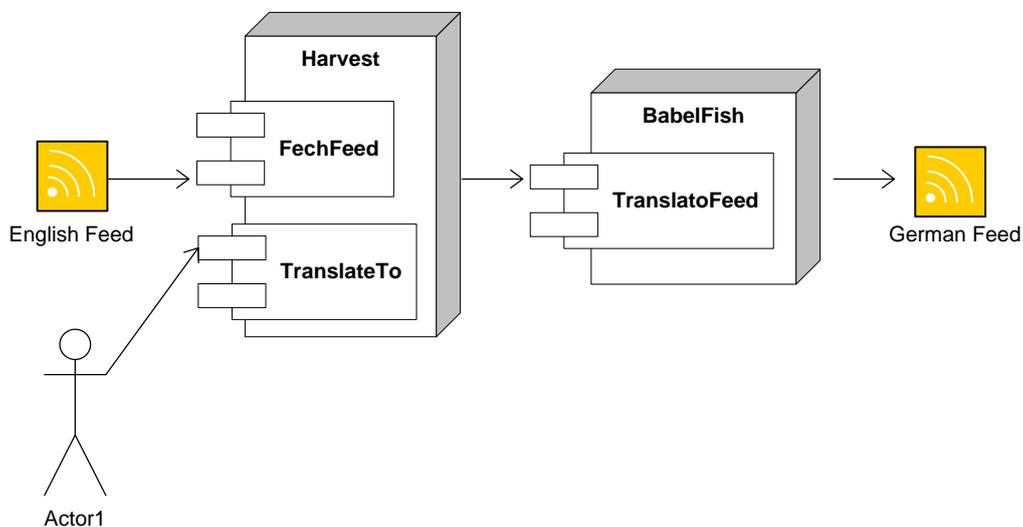
El patrón 'Translation' describe una técnica para permitir la 'internacionalización' y la 'localización dinámica'. Dos métodos pueden ser utilizados en una aplicación para la ampliación a un nuevo público:

- Construir una nueva interfaz por idioma. La adición de características de internacionalización de un producto ya existente por lo general requiere la pericia

del equipo de desarrollo original. Con Mashups, sin embargo, una interfaz nueva puede ser construida. Esta nueva capa transforma los datos y formatos numéricos a través de equivalentes culturales.

- Utilizar traducción sobre la marcha. Muchos servicios de traducción están disponibles en la Web; productos comerciales están disponibles para la compra también. La exactitud de la traducción automática va desde 50% a 85%, dependiendo de los idiomas y la materia en cuestión. Un Mashup puede utilizar un producto de traducción automática detrás de escenas, una fachada temporal puede ser creada que proporcione la interfaz alternativa e incorpore la funcionalidad de traducción.

## Diagrama Uml:



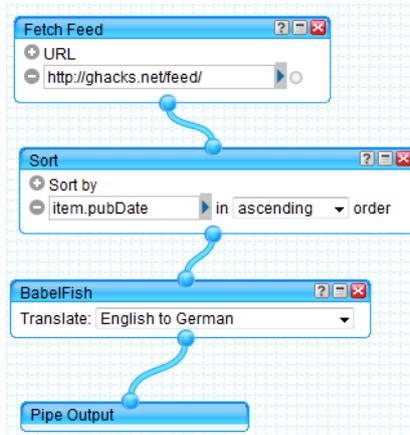
## Fragilidad:

El patrón 'Translation' utiliza las técnicas 'Usability' y 'Field Medic'. Como consecuencia, es susceptible a los mismos problemas potenciales y también merece la clasificación más alta de fragilidad. El patrón Translation también usa Mashups adicionales para acceder de forma dinámica a los recursos de traducción de idiomas.

## Ejemplo

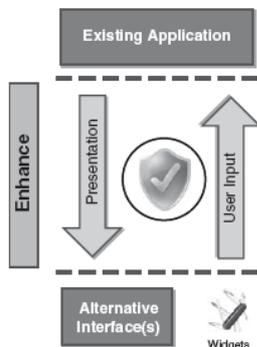
La mayoría de los usuarios de Internet proceden de países que de no habla inglesa. Es evidente que sería absurdo limitar las búsquedas de información a una gramática. A pesar que algunos motores de búsqueda ya devuelven los resultados a través de varios idiomas, la calidad de los resultados es probable que sea baja.

Un Mashup de traducción puede añadir claridad a una búsqueda para que los resultados sean más útiles. En este ejemplo permite que un feed sea traducido a un idioma en particular.





## Usability Enhancer



### Problema

Los desarrolladores de software definen la usabilidad como la medida de lo fácil que es utilizar una aplicación para completar una tarea en particular. Es irónico que el término en sí sea en gran parte desconocido por los usuarios finales, pero eso no quiere decir que no sea importante para ellos.

La mayoría de los desarrolladores de software se concentran demasiado en los aspectos técnicos de su profesión y no desarrollan una buena apreciación de la perspectiva de visión del usuario.

Otros factores entran en juego, por supuesto. Un vendedor que está creando un producto utilizado por miles de clientes no es capaz de adaptarse a los deseos de cada cliente. Como dijo la famosa frase de Bill Cosby, "No sé la clave del éxito, pero la clave del fracaso es tratar de complacer a todo el mundo." Los diseñadores de aplicaciones pueden centrarse en la usabilidad de las características más populares incluso si eso significa hacer sacrificios en otras áreas.

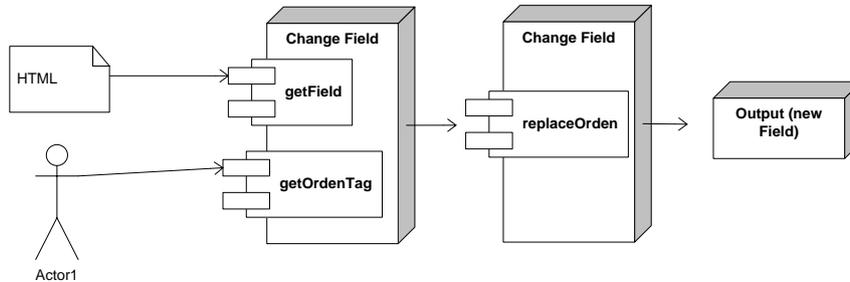
### Solución

Dado un sistema con una interfaz complicada, un Mashup "wrapper" (o facade) puede construirse para hacer frente a problemas de usabilidad [5]. Debido a que el Mashup se encargará de la comunicación entre el sistema existente y la nueva interfaz, no se requiere arreglos de la aplicación original. Los siguientes cambios pueden mejorar la facilidad de interacción con el sistema [24,25]:

- Cambiar el orden de entrada de datos de los campos para que coincida con el flujo de trabajo del usuario.
- Cambiar el nombre de las etiquetas y otros textos para reflejar el negocio.
- Añadir validaciones para evitar los errores más comunes.
- Ocultar las características innecesarias que sólo causan confusión.
- Aplicar automáticamente configuración y valores de uso común.
- Exponer un proceso de varios pasos, como una sola tarea o, por el contrario, romper el proceso en pedazos más pequeños

- Añadir la imagen corporativa, cambiar colores de la pantalla, y emplear otras medidas para volver a empaquetar un conjunto de aplicaciones dispares como un conjunto unificado.
- Acepta formatos adicionales de entrada.

## Diagrama Uml:

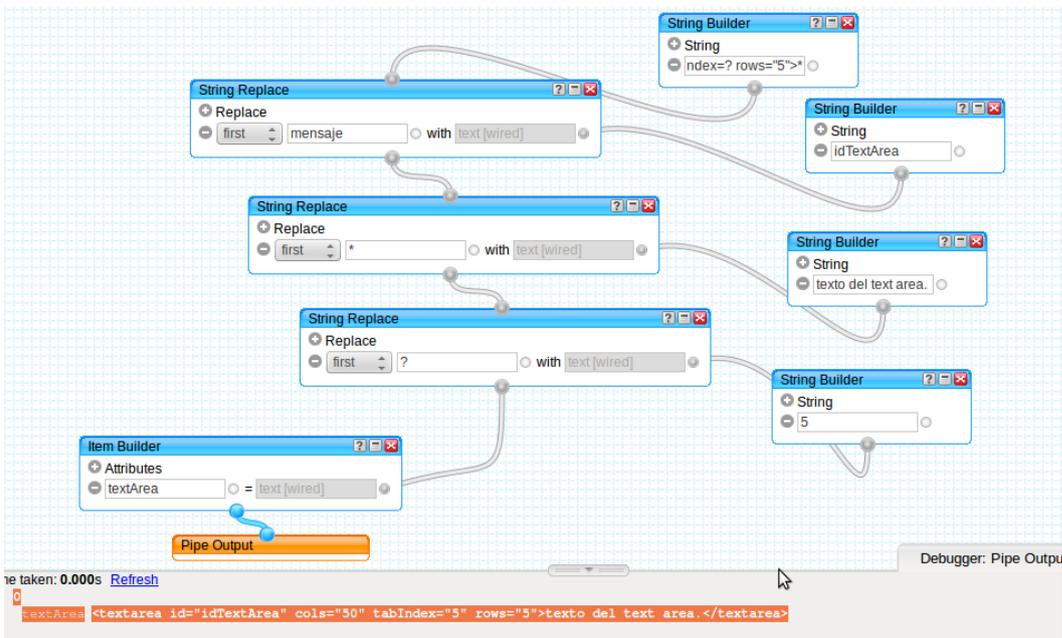


**Fragilidad:**

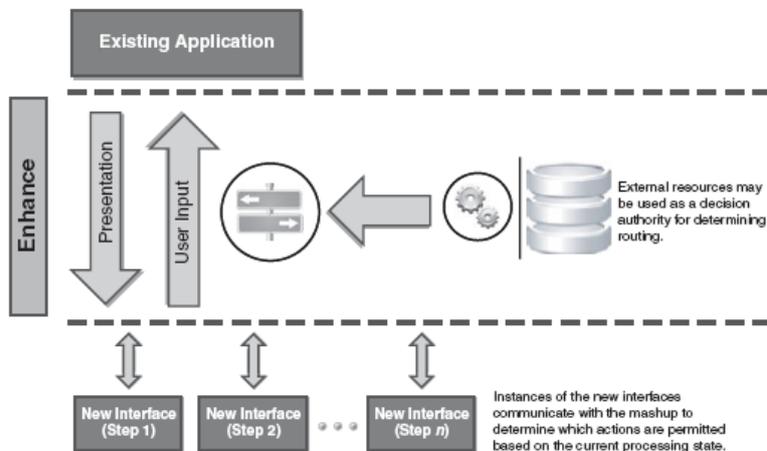
Al igual que el patrón Field Medic, Usabilidad Enhancer implica la creación de una fachada que utiliza los Mashups para interactuar con una aplicación de base. La fragilidad potencial es menor, ya que es sólo parcialmente 'envuelto'.

## Ejemplo

En este ejemplo se utiliza un Mashups para modificar el orden de un textArea.



## Workflow



### Problema

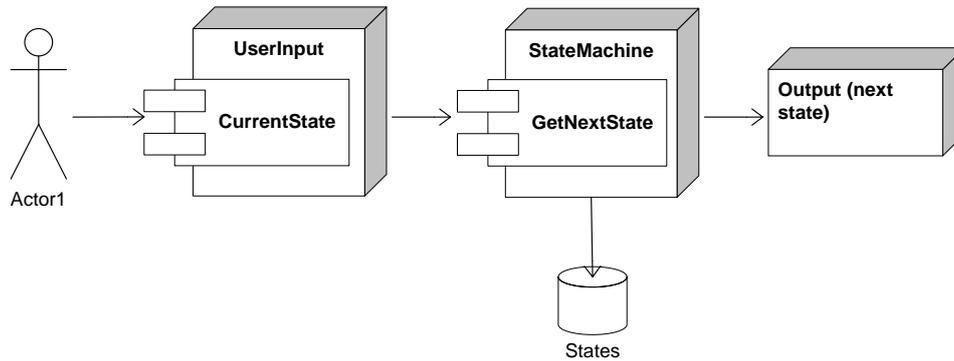
Casi todas las empresas tienen un conjunto de procedimientos predefinidos que enmarcan las acciones y responsabilidades de sus empleados. Estos flujos de trabajo permiten que la tarea se complete de manera organizada y repetible. Cuando los resultados no son deseados o no tiene éxito, un flujo de trabajo bien documentado, permite que se verifique el proceso para localizar el punto de falla. ¿Cómo pueden los programas adaptarse rápidamente a las necesidades de flujo de trabajo nuevo? ¿Qué puede hacerse cuando una aplicación que no fue diseñada para soportar los de flujo de trabajo de repente tiene que hacerlos cumplir?

### Solución

El patrón 'WorkFlow' se refiere específicamente a los problemas recurrentes y soluciones comprobadas relacionadas con el desarrollo de aplicaciones de flujo de trabajo en particular y, en general, a las aplicaciones orientadas a procesos [26]. Los Mashups también pueden ser usados para agregar capacidades de flujo de trabajo a un sistema que no admite esta funcionalidad de forma nativa. El primer paso consiste en definir el proceso y determinar los sistemas existentes que participarán en la solución. Si un flujo de trabajo existente se va a modificar, a continuación, los puntos donde se insertan las actividades nuevas o a eliminar tienen que ser capturados.

En el 'Workflow', el 'API Enabler' realiza la función adicional de la exposición de métodos que permiten monitorear los estados específicos. El siguiente paso es aplicar las técnicas descritas en el patrón Usability Enhancer. Se construye una nueva interfaz que utiliza la información de estado obtenido a partir de las implementaciones de la 'API Enabler' para permitir o restringir ciertas operaciones adicionales.

## Diagrama Uml:



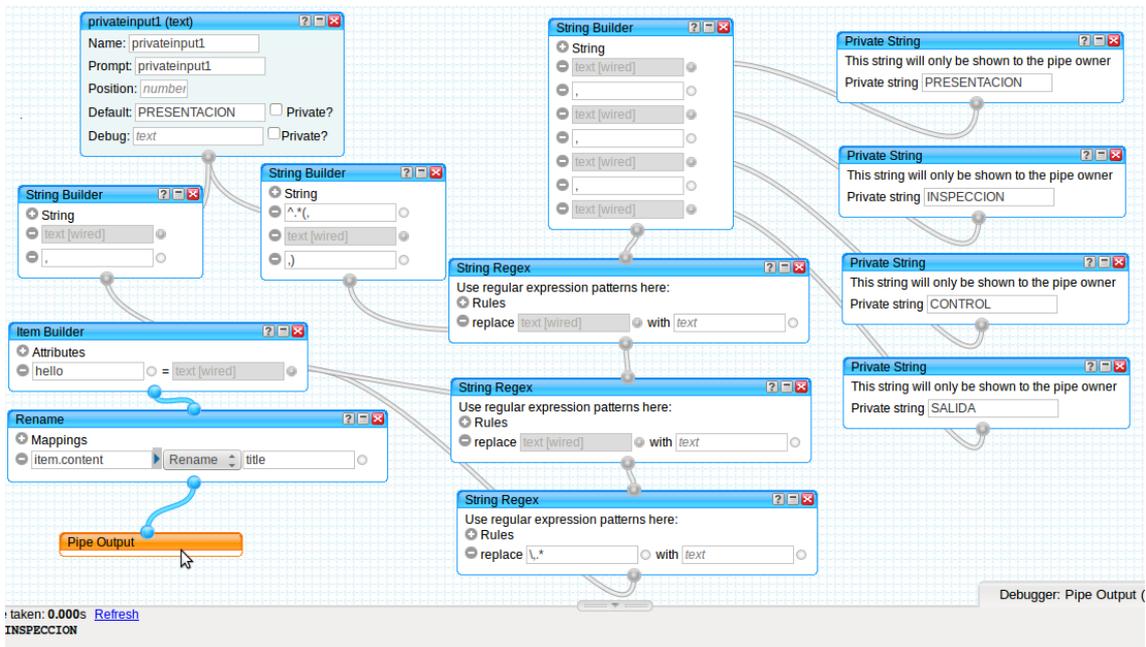
## Fragilidad:

El patrón 'Workflow' tasa la puntuación más alta fragilidad debido a que una avería en cualquier parte del proceso puede dejar los elementos en un estado donde no pueden pasar a la siguiente etapa de procesamiento.

## Ejemplo

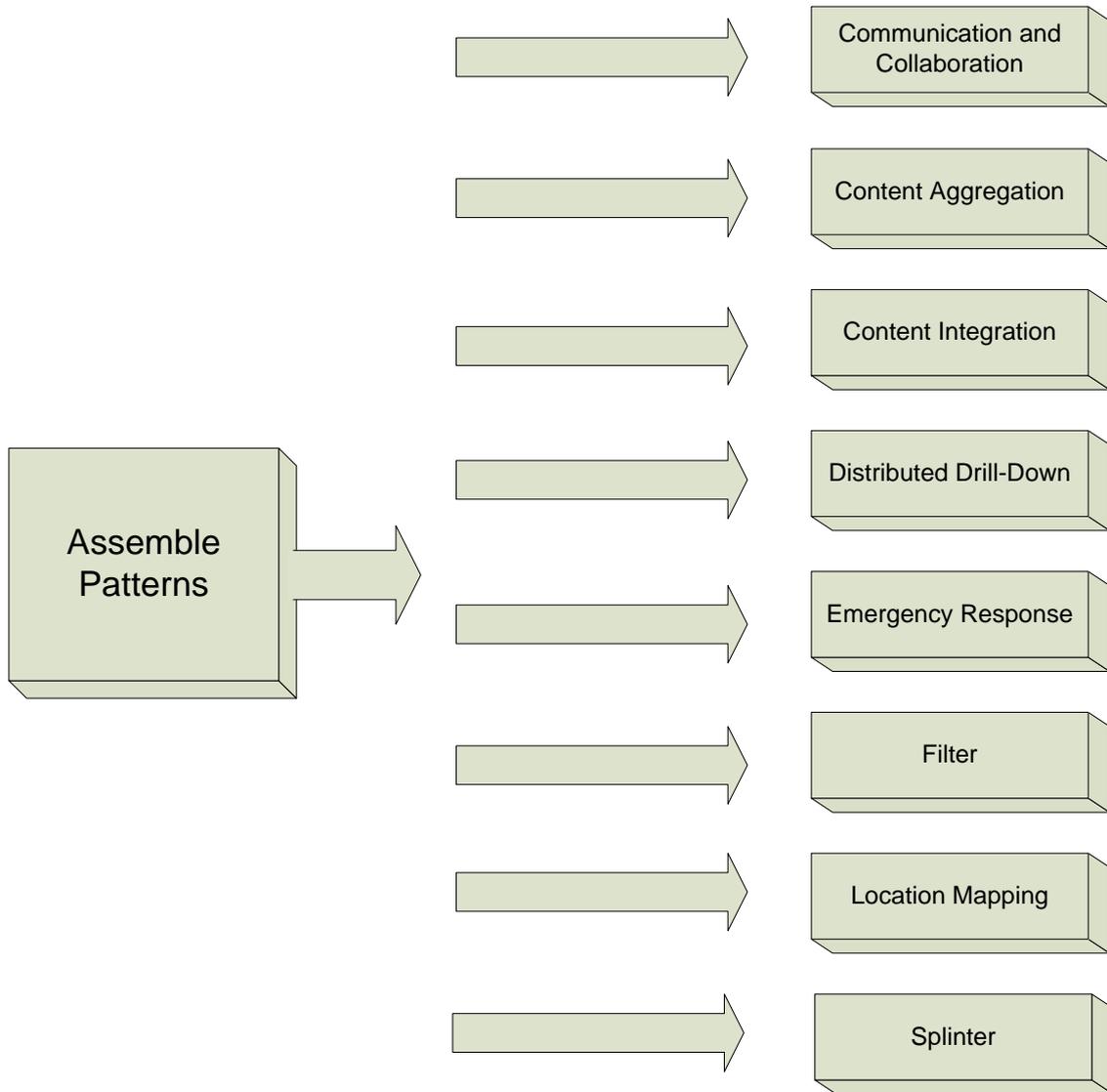
En este ejemplo se utiliza yahoo pipes para crear un workflow simple. En dicho ejemplo pertenece a un workflow en la entrada-salida de un camión a un puesto de vigilancia. El flow está compuesto por los siguientes pasos: Presentación->Inspección->Control->Salida. Como entrada el Mashup recibe el estado actual, dando como salida el próximo estado posible.

<http://pipes.yahoo.com/pipes/pipe.info?id=1707f68673d887fd2970b4b2524841e2>



- Capítulo 5 -

# Assemble Patterns



## Filter



### **Problema**

La sobrecarga de información se refiere a la dificultad que una persona puede tener en la comprensión de un problema y en la toma de decisiones dado por la presencia de demasiada información [27].

El volumen de una empresa típica, el correo electrónico está creciendo a un ritmo de casi un 32% por año. El simple hecho de almacenar estos mensajes se está convirtiendo en un problema enorme.

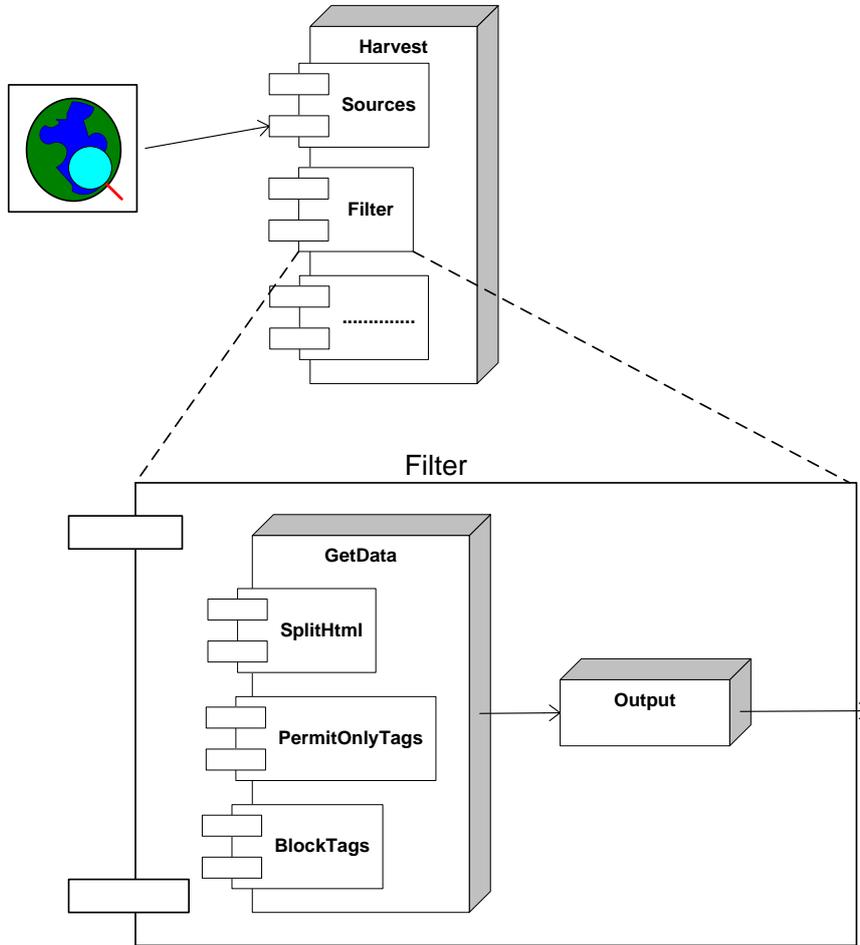
Si los casos de pérdidas de oportunidades causadas por las distracciones pudieran ser medidas, el resultado del costo real mostraría lo alarmante de la situación.

Estos sucesos aleatorios pueden interrumpir el trabajo de un empleado en cualquier momento durante el día y puede conducir a un estado mental donde la anticipación de las distracciones hace que sea difícil concentrarse en la tarea. Los filtros de spam pueden bloquear mensajes no deseados de correo electrónico. Los motores de búsqueda proporcionan miles de resultados, incluso para la consulta más simple. RSS proporciona una corriente constante de interrupciones.

### **Solución**

Los Mashups están preparados para ayudar a resolver el problema de sobrecarga, ya que interactúan con muchas fuentes de información. Mediante la organización de los datos este esfuerzo ofrece "seguridad de información" mediante una sensación de seguridad personal cuando ya no hay que preocuparse de prestar atención a todo de una vez.

## Diagrama Uml:



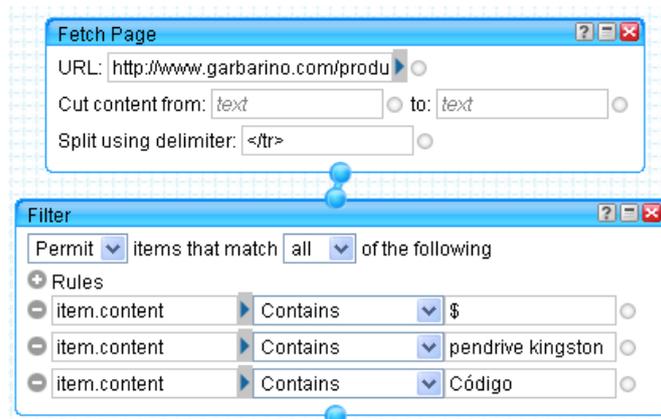
## Fragilidad



El peligro obvio en el filtrado de información es que puede que podemos haber perdido algo importante. El patrón del filtro se aplica igualmente a la información que se recibe y se crea. Otra forma de disminuir el riesgo de pérdida de información es simplemente crear menos información en el primer lugar.

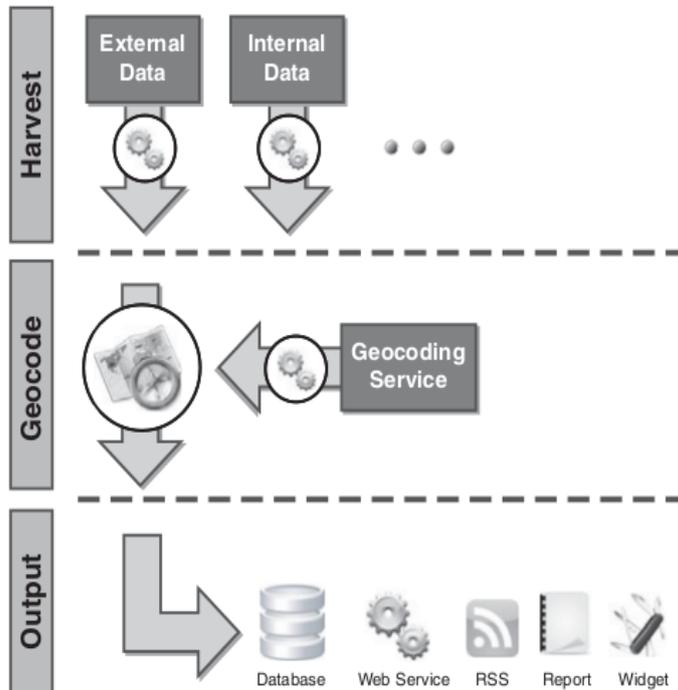
## Ejemplo

Para este ejemplo regresaremos al ejemplo de la búsqueda de precios para un pendrive en particular en la página de productos de garbarino. La aplicación del modulo filter permite filtrar los elementos de la pagina que no se desean y distraen al usuario. Este filtro permite obtener los tags que contengan '\$', 'kingston' y código, filtrando todos los productos que no lo cumplan por tener precio en dólares o no ser 'kingston'. También elimina los tags con propagandas, banners e imágenes que puedan desviar la atención del usuario.





## Location Mapping

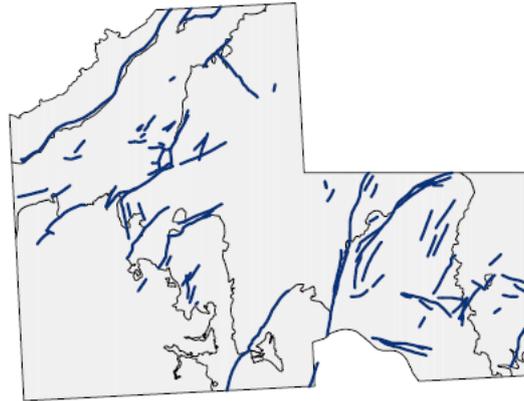


### Problema

Muchos sistemas contienen información sobre la posición global, pero esta no se encuentra en un formato útil. Una aplicación puede registrar los domicilios de su personal pero no ubicarlos geográficamente, a pesar de que esta información podría ser útil para fines de planificación de riesgos. Una tienda al por menor pueden ofrecer productos a la venta, pero no anunciarlo lo suficiente como para llegar al mayor número de clientes potenciales. La identificación de recursos mediante su ubicación ha sido una actividad importante desde mucho antes del advenimiento de la tecnología moderna.

La posibilidad de mostrar estos datos en un mapa es útil, sin embargo, puede ser igualmente importante calcular las posiciones relativas de ciertos artículos. Pero antes de que cualquiera de estas tareas pueden llevarse a cabo, toda la información geográfica deben ser recogida y convertida a un formato común.

Ingenieros geólogos pueden querer identificar las mejores ubicaciones para construcción de viviendas en un área con terremotos regulares mediante la identificación de las características de las rocas

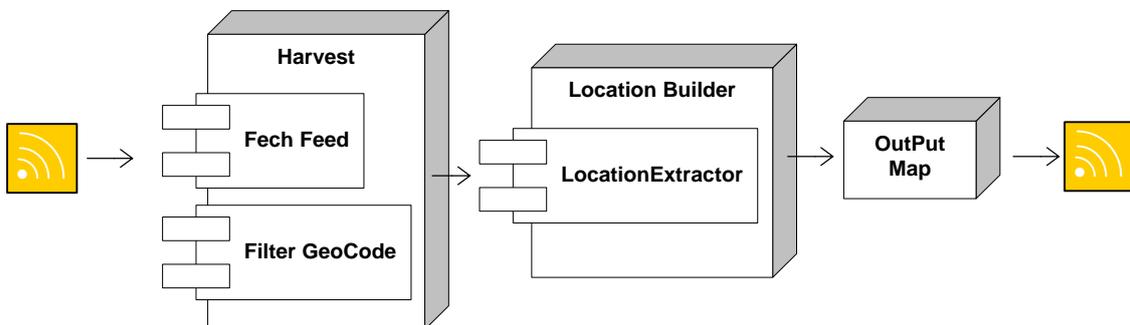


En la figura anterior se muestra un mapa indicando en azul las fallas de la zona.

## Solución

Un Sistema de Información Geográfica es una integración organizada de hardware, software y datos geográficos diseñada para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión geográfica [28]. Muchos productos Web pueden resolver la información de ubicación con diversos grados de precisión. Una implementación de mapeo de ubicación puede mezclar y combinar las diferentes regiones que sirven para crear un servicio de validación global. Los patrones de 'Smart Suggestions', 'Field Medic', and 'Usability Enhancer' sugieren diferentes técnicas para mejorar las aplicaciones con esta capacidad adquirida. Los servicios de validación de direcciones se pueden utilizar con las bases de datos existentes para marcar registros específicos para posterior investigación o eliminación. Un segundo uso del patrón de mapeo de la ubicación implica el uso de direcciones como un medio para establecer la ubicación relativa de los objetos de interés. El último y más popular el uso del patrón de mapeo de ubicación es visualizar los datos espaciales mediante un sistema de información geográfica (SIG).

## Diagrama Uml:



## Fragilidad

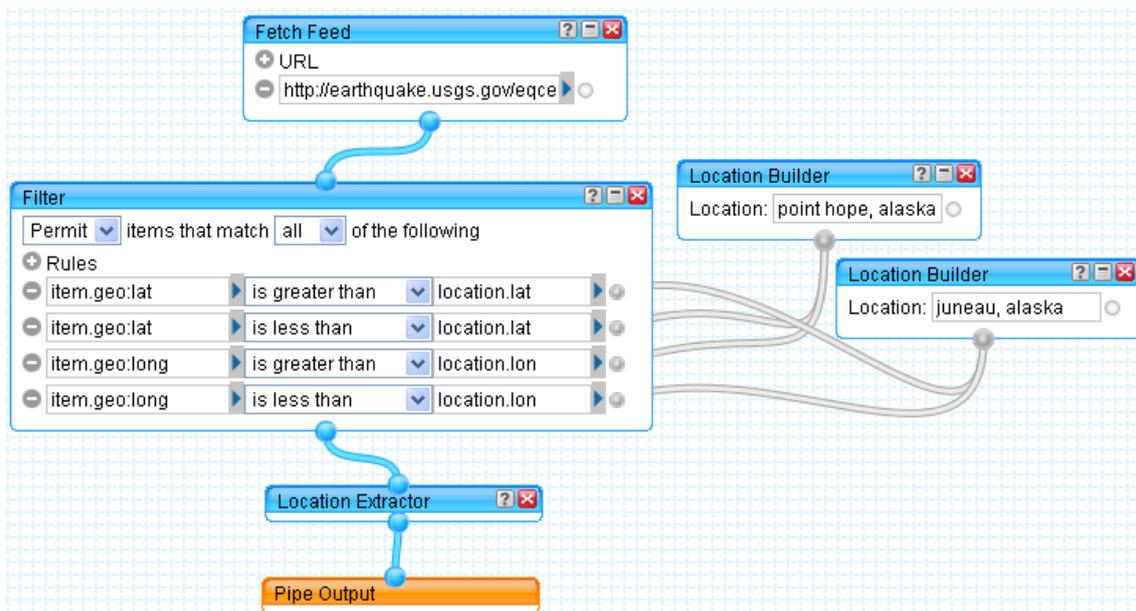


Aunque el patrón se manifiesta en tres maneras diferentes (verificación, visualización y localización), la geocodificación está en el corazón de cada una de estas implementaciones. El riesgo se origina en las situaciones en las que la información de dirección no puede ser establecida. Este fracaso puede llevar al almacenamiento de información incorrecta o indicar puntos de datos fuera de un mapa. En cada uno de estos casos es sencillo atrapar la condición de error y notificarla al usuario antes de que cualquier daño se produzca. El mapeo de ubicación recibe la clasificación más baja de fragilidad por esta razón.

## Ejemplo

<http://pipes.yahoo.com/pipes/pipe.edit?id=b62e741f3229069b3687fe27bbfeaba0>

En el siguiente ejemplo se utiliza el módulo 'Location Extractor' y 'Filter' para obtener la ubicación de sectores con terremotos que se encuentren entre un rango de latitudes



Mostrando como resultado del Mashup



## Usos conocidos

Housingmaps es el Mashup prototípico utilizando Google Maps + Craigslist crea una página de búsqueda de viviendas a través de superposiciones visuales dinámicas de listas de inicio de Craigslist en Google Maps.

APIs

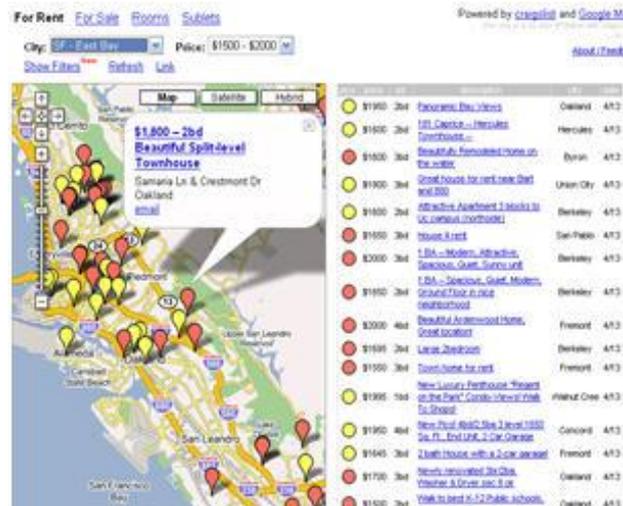
Google Maps

Creado:

20 Nov 2005

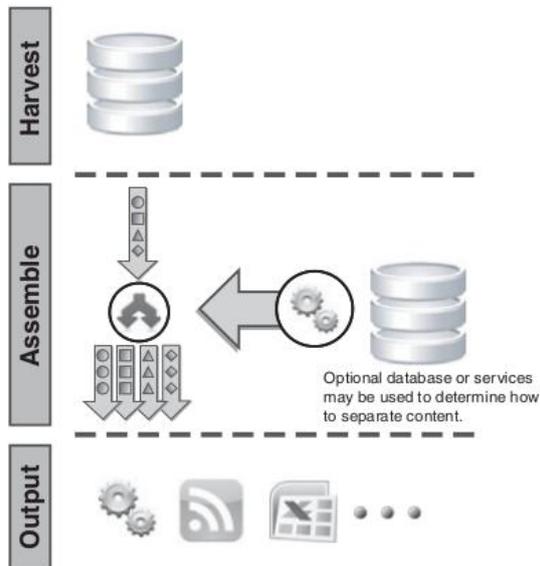
URL

<http://www.housingmaps.com/>





## Splinter



### Problema

Muchas empresas han adoptado el concepto de DataWarehouse (DW) a fin de gestionar sus datos empresariales. En informática, un Datawarehouse es una base de datos utilizada para la presentación de informes y análisis de datos. Se trata de un repositorio de datos central que se crea mediante la integración de datos de múltiples fuentes dispares [29]. El entorno está destinado a permitir a los usuarios finales un acceso fácil a los datos corporativos. Debido a que sus esfuerzos son costosos y lentos, algunas empresas optan por centrarse en la agregación de inteligencia de negocio específica o información de Supply-Chain.

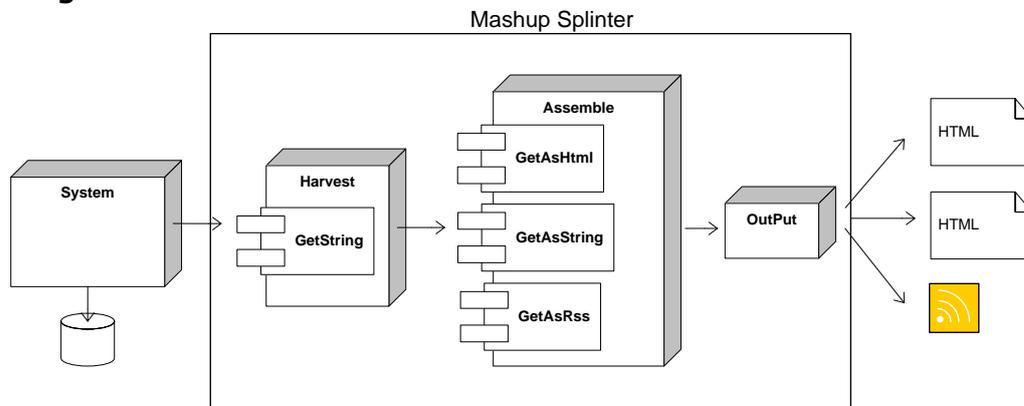
### Solución

Mediante el patrón 'Content Integration' se puede combinar la información empresarial. Este proceso puede ser un paso clave en la creación de un Data Warehouse. Además mediante las técnicas para la eliminación de información innecesaria o redundante aplicando el patrón 'Filter'. El componente final es a la partición de los datos en los incisos pertinentes. El patrón Splinter toma una sola fuente de datos y hace su contenido disponible en nuevas formas de fuentes de datos. A medida que se ilustra el "problema", este requerimiento es común en donde se utiliza un único repositorio de datos. Una aplicación de Splinter no se limita a la creación de nuevos feeds. Se puede considerar esta capacidad como una variación del patrón del 'API Enabler'.

El valor del patrón 'Splinter' no se limita solo a hacer un DW fácil de usar. Si se va a construir un 'Dashboard' para diversas actividades de compra, esas métricas tendrán que ser extraídas de forma individual de diferentes porciones de la base de

datos. En esta situación se puede aplicar Splinter para proporcionar a cada métrica la información necesaria para su visualización.

## Diagrama Uml:



## Fragilidad

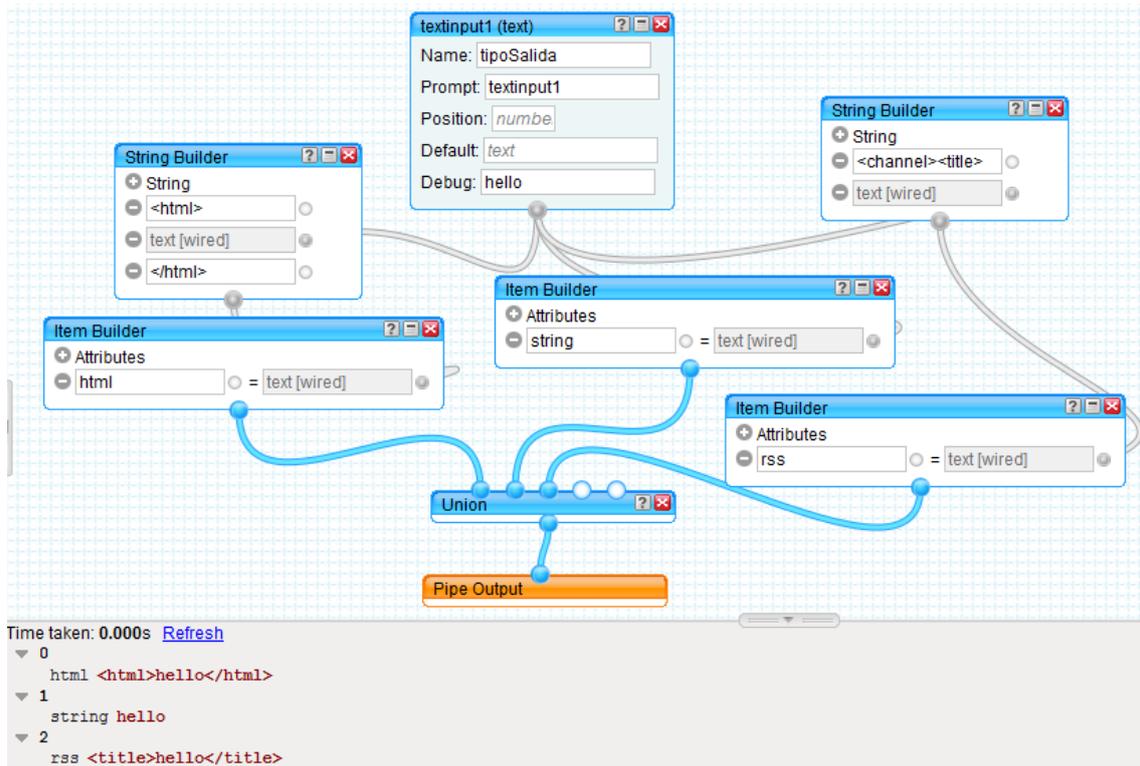


Al igual que todos los Mashups, el patrón de Splinter no está exento de riesgos. El principal riesgo es dejar algo de lado en los resultados finales. También existe la posibilidad de que el movimiento de datos a un formato de salida diferente cambie su valor. El objetivo del patrón de Splinter es la creación de múltiples artefactos los cuales puedan ser utilizados por una aplicación o usuarios.

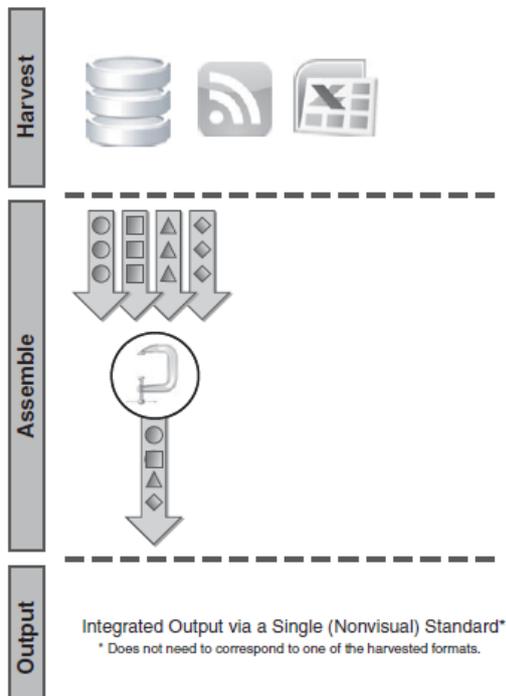
## Ejemplo

En este ejemplo simple se muestra como los Mashups pueden obtener los datos y transformarlos en distintos formatos para que sean utilizados por otra aplicación.

[http://pipes.yahoo.com/pipes/pipe.info?\\_id=475d92b25c5f624e2e694aa12159946d](http://pipes.yahoo.com/pipes/pipe.info?_id=475d92b25c5f624e2e694aa12159946d)



## Content Integration



### Problema

Un Mashup puede extraer datos de cientos o miles de sitios Web por lo que la tarea de combinar todo el material obtenido es abrumadora. Los valores más simples tales como fechas y números de teléfono se pueden presentar en diferentes formatos.

El departamento de IT ya gasta una cantidad significativa de tiempo al administrar un flujo transparente de datos desde y hacia otras aplicaciones (SOA) [30]. Hoy en día las técnicas incluyen bases de datos relacionales, servicios Web, XML, RSS, JSON, y muchos más. Una aplicación diseñada para procesar XML no puede consumir los detalles de una base de datos sin cambios en su código fuente. La proliferación de tantos almacenamientos de datos distintos y métodos de transmisión ha creado una Torre de Babel tecnológica.

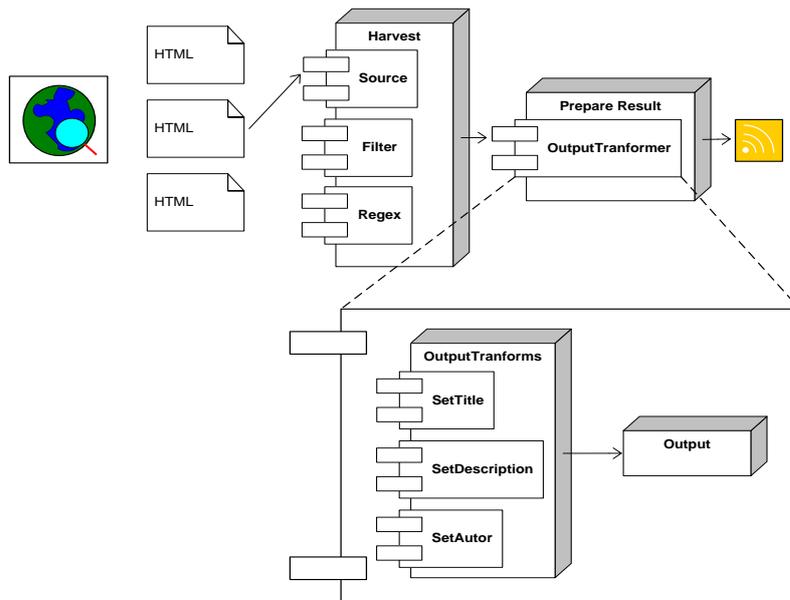
### Solución

Las aplicaciones de una empresa pueden ser hechas a medidas, adquiridas de terceros o legacy. En orden de soportar procesos de negocio común y el libre compartimiento de datos a través de aplicaciones, estas aplicaciones necesitan ser integradas. El patrón de integración proporciona un método rápido para la combinación de flujos de datos de múltiples sistemas. Este nuevo flujo puede ser integrado en los procesos de negocio y aplicaciones existentes. A diferencia de las implementaciones del 'Aggregation pattern', las implementaciones de 'Content

'Integration' no tienen un componente visual.

Una característica del modelo de 'Content Integration' es que puede ser implementado por usuarios que estén fuera del dominio de flujo de datos. El patrón 'Content Integration' puede servir como una medida provisional para crear nuevos datos compuestos y alimentar sistemas. Aunque este patrón debe usarse con precaución para no crear dependencias del sistema que estén ocultas o no estén bien comprendidas.

## Diagrama Uml:



## Fragilidad



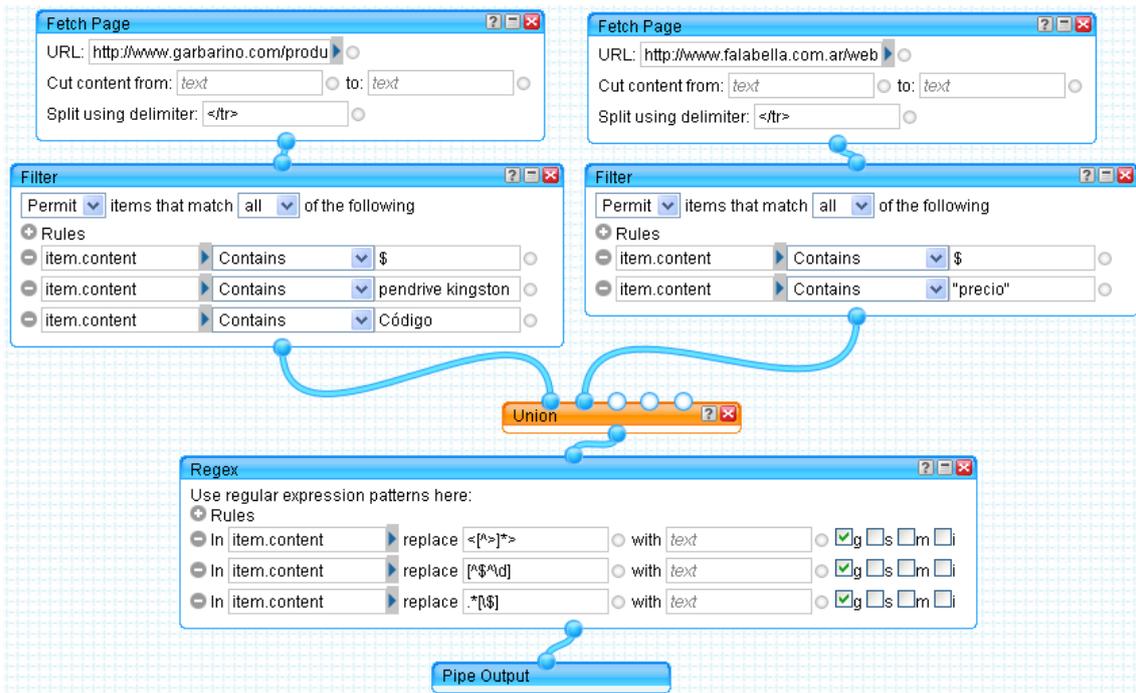
Los Mashups que manipulan los datos tienen un mayor nivel de riesgo. Un error en una de las partes constituyentes puede hacer que el recurso combinado sea inutilizable. Uno de los principales beneficios del modelo de integración de contenido es también su mayor debilidad. Una de las técnicas para hacer frente a este problema es añadir metadatos a los recursos combinados que señalan a los sistemas de origen. Estos metadatos se pueden utilizar para rastrear los problemas de vuelta a través del Mashup a la fuente original del material.

## Ejemplo

En el ejemplo de 'Competitive Analysis' se utilizó el patrón de 'Infinite Monkeys' para obtener y procesar la información de distintas fuentes. En este ejemplo se compara el precio de un pendrive de cierta marca en Garbarino contra los productos del tipo pendrive de Falabella.

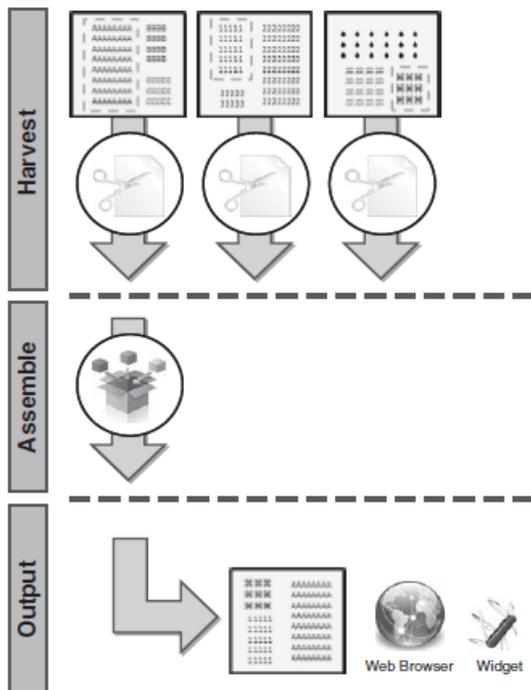
Una vez que los datos son obtenidos por los distintos 'Monkeys', se unifican para posteriormente darle un formato de salida en común.

<http://pipes.yahoo.com/pipes/pipe.edit?id=7f838b46ab03a8dfb11ddb64521e09fc>





## Content Aggregation



### Problema

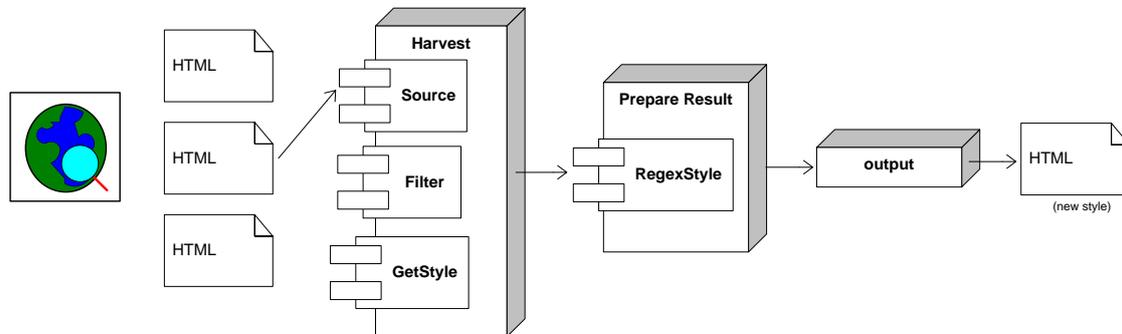
Hoy en día en las computadoras ejecutan fácilmente docenas de aplicaciones a la vez. Organizadas en ambientes ricos en gráficos llegan a un punto donde el número de herramientas al mismo tiempo a nuestra disposición puede sobrepasar nuestra capacidad de interactuar con ellas de forma simultánea. Esto no es un caso de sobrecarga de información debido a que el trabajo en sí mismo podría ser manejable. Más bien, la raíz del problema es la multitarea, la labor de cambiar entre tantas aplicaciones con interfaces diferentes con el fin de completar un solo trabajo. En la era pre-Mashup no había alternativas razonables a la multitarea en línea, por lo que no tenía mucho sentido en la investigación de este tema. La evidencia sugiere que en relación con el cambio entre sistemas dispares presenta visualmente un impedimento cognitivo momentánea para el usuario.

### Solución

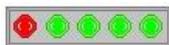
Los nuevos paradigmas de interfaz tales como RSS (ver 'Feed Factory' pattern), portales (ver 'Portal Enabler' pattern) y widgets de escritorio (ver Widget Enabler pattern) son sólo algunos de los recientes intentos encaminados a aliviar la sobrecarga de los sentidos [31]. Lo que les diferencia del patrón 'Content Aggregation' es que producen nuevos artefactos de salida que no estaban previstos originalmente. El patrón 'Content Aggregation' consiste en la recolección de los elementos visuales de uno o más sitios para ser adaptados o recombinados dentro

de un contenedor de presentación común con poca o ninguna modificación. Su complemento natural es 'Integration pattern'.

## Diagrama Uml:



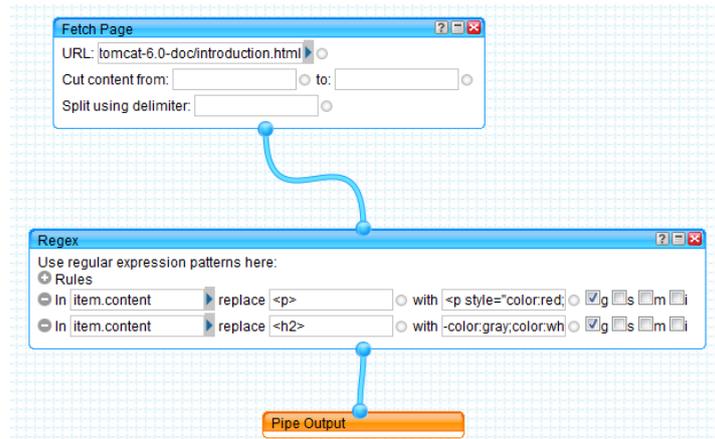
## Fragilidad



El patrón 'Content Aggregation' recibe una calificación baja de fragilidad, porque implica sólo la comunicación simple entre componentes. Si la presentación de material agregado es modificada por el Mashup, existe un riesgo potencial de que la nueva visualización pueda introducir errores. Otro riesgo es que la información fuera de su entorno natural, no pueda transmitir su importancia. También es posible que un Mashup 'Content Aggregation' incluya contenido fuera de fecha. Otros riesgos se relacionan con cuestiones de seguridad, debido a que podría exponer información que en su contexto anterior era segura.

## Ejemplo

En este simple ejemplo se muestra como Mashups puede resolver el problema de sobrecarga de los sentidos. Los estilos de las pantallas del sistema original pueden ser modificados capturando el HTML y agregando un nuevo CSS. También modificando los anchos de tablas o agregando nuevos títulos a fin de crear un único frontend homogéneo.



Reemplazando el estilo de los tags que se deseen modificar se pueden realizar los cambios que se crean necesarios.

## Links

- [Docs Home](#)
- [FAQ](#)

## User Guide

- [1\) Introduction](#)
- [2\) Setup](#)
- [3\) First webapp](#)
- [4\) Deployer](#)
- [5\) Manager](#)
- [6\) Realms and AAA](#)
- [7\) Security Manager](#)
- [8\) JNDI Resources](#)
- [9\) JDBC DataSources](#)
- [10\) Classloading](#)
- [11\) JSPs](#)
- [12\) SSL](#)
- [13\) SSI](#)
- [14\) CGI](#)
- [15\) Proxy Support](#)
- [16\) MBean Descriptor](#)
- [17\) Default Servlet](#)

## Apache Tomcat 6.0

### Introduction

#### Table of Contents

- [Introduction](#)
- [Terminology](#)
- [Directories and Files](#)
- [Configuring Tomcat](#)
- [Where to Go for Help](#)

#### Introduction

For administrators and web developers alike, there are some important bits of information you should familiarize yourself with before starting out. This document serves as a brief introduction to some of the concepts and terminology behind the Tomcat container. As well, where to go when you need help.

#### Terminology

In the course of reading these documents, you will run across a number of terms; some specific to Tomcat, and others defined by the [Servlet](#) and [JSP specifications](#).

## Después

## Links

- [Docs Home](#)
- [FAQ](#)

## User Guide

- [1\) Introduction](#)
- [2\) Setup](#)
- [3\) First webapp](#)
- [4\) Deployer](#)
- [5\) Manager](#)
- [6\) Realms and AAA](#)
- [7\) Security Manager](#)
- [8\) JNDI Resources](#)
- [9\) JDBC DataSources](#)
- [10\) Classloading](#)
- [11\) JSPs](#)
- [12\) SSL](#)
- [13\) SSI](#)
- [14\) CGI](#)
- [15\) Proxy Support](#)
- [16\) MBean Descriptor](#)
- [17\) Default Servlet](#)

## Apache Tomcat 6.0

### Introduction

#### Table of Contents

- [Introduction](#)
- [Terminology](#)
- [Directories and Files](#)
- [Configuring Tomcat](#)
- [Where to Go for Help](#)

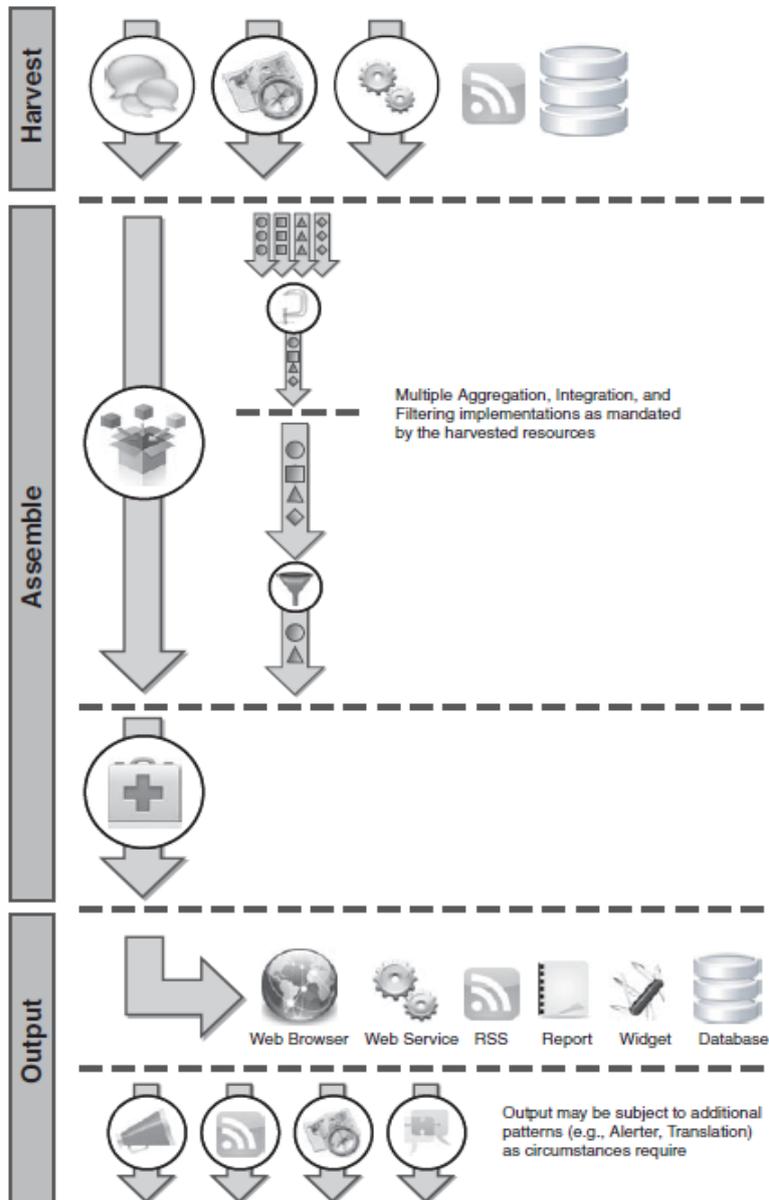
#### Introduction

For administrators and web developers alike, there are some important bits of information you should familiarize yourself with before starting out. This document serves as a brief introduction to some of the concepts and terminology behind the Tomcat container. As well, where to go when you need help.

#### Terminology

In the course of reading these documents, you will run across a number of terms; some specific

 **Emergency Response**



**Problema**

El ciclo de desarrollo de sistemas consiste en el análisis de los requisitos y el diseño de arquitectura. Incluso 'extreme programming' propone armar los test antes de la implementación. A veces "el camino correcto" es demasiado lento cuando surgen oportunidades que necesitan ser explotadas en un corto tiempo. El patrón 'Emergency Response' se ocupa de estas situaciones críticas. El "lado positivo" a estas situaciones es que el problema ya ha sido medido, aunque en términos

generales. En estas condiciones la "emergencia" se reduce a una programación planificada de los recursos y mano de obra. Nuestra preocupación aquí son las emergencias de carácter más drástico, cuando la seguridad y el bienestar humano está en riesgo. Los desastres naturales tales como terremotos, inundaciones, huracanes, tornados y tormentas de invierno son los principales ejemplos. La solución requiere reunir una colección de recursos, como organizaciones de beneficencia, corporaciones y agencias gubernamentales, todos son sistemas que nunca se han diseñado ni pensado para inter-operar entre ellos.

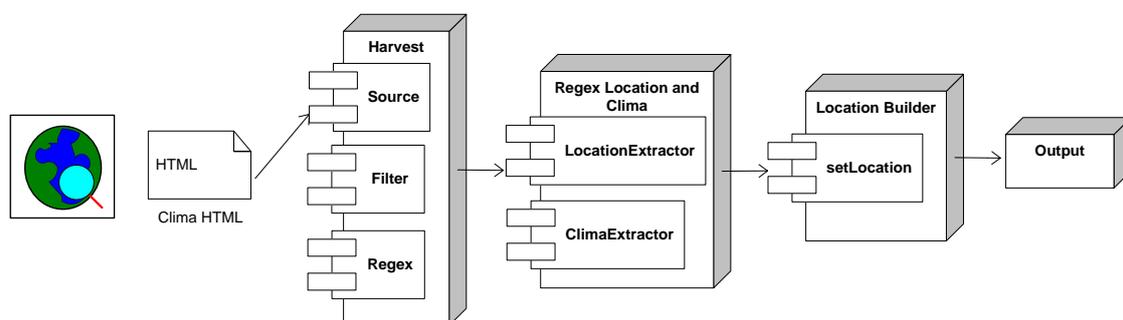
## Solución

El poder de los Mashups que cuando se crean conjuntamente por varias personas es útil durante una crisis, cuando ninguna entidad tiene toda la información que el público exige. La aleatoriedad de los desastres y sus efectos hacen que sea imposible proporcionar directrices generales para el implementador. Los Mashups son una solución natural para la coordinación de activos dispares en condiciones extremas. El primer paso en la coordinación de un esfuerzo es obtener toda la información posible. Por ejemplo Agencia de Manejo de Emergencias, Administración de Seguridad y Salud Ocupacional son buenos puntos de partida. Un Mashup 'Emergency Response' puede reunir alguno o todos los recursos que se necesiten en función de la gravedad del incidente [32]:

- Imágenes de satélite y previsiones meteorológicas.
- Ubicación de las fuentes seguras de agua potable.
- Ubicación de las fuentes de asistencia alimentaria.
- Advertencias de puentes inseguros y carreteras.
- Información médica y contactos.
- Ubicación de centros móviles de recuperación de desastres.
- Cantidad disponible y ubicación de las bombas de sumidero.
- Retrasos en transportes.
- Una wiki, o "directorio de personas" para ayudar a localizar a parientes lejanos.

Este Mashup puede permanecer en su lugar como una ayuda para emergencias proporcionando ayuda durante catástrofes futuras.

## Diagrama Uml:



## Fragilidad

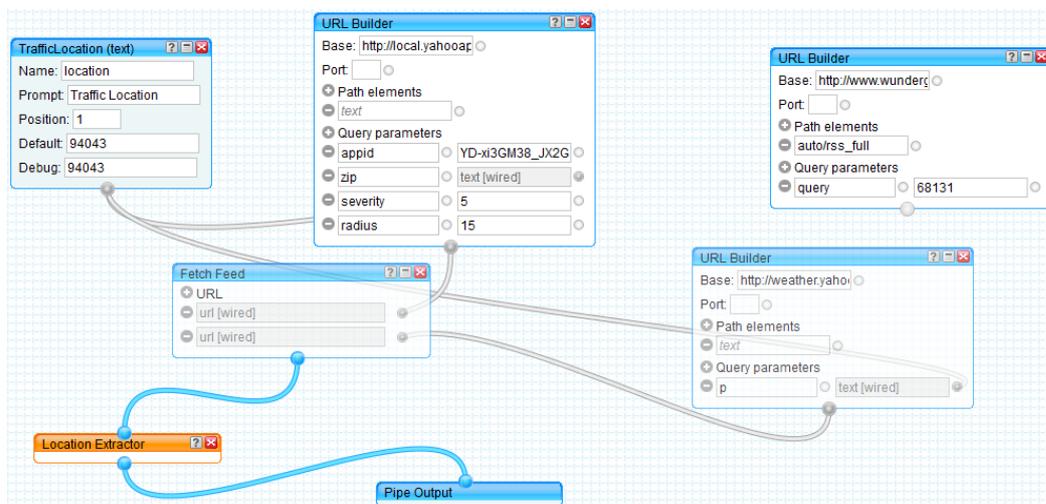


La desventaja de proporcionar rápidamente una respuesta Mashup es que puede

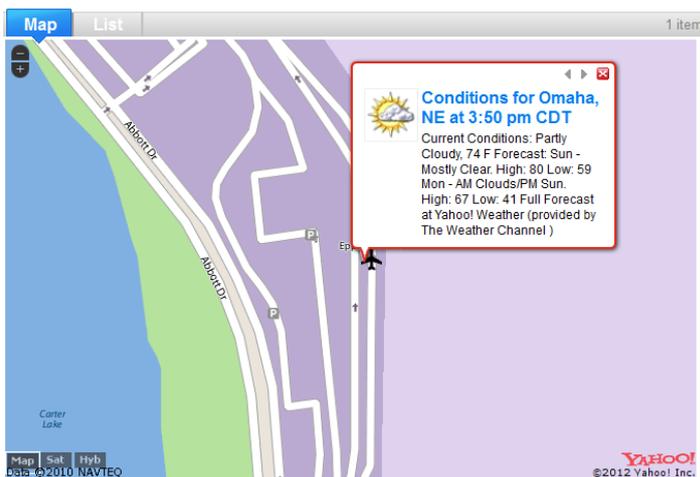
tener consecuencias negativas. Es fácil de propagar información inexacta o contradictoria. Mezclar mala información con buenas intenciones puede dar falsa credibilidad. Mínimamente, algún tipo de calificación simple o mecanismo de retroalimentación se tiene que incorporar para aprovechar el conocimiento de las personas cercanas a la situación de emergencia.

## Ejemplo

Este ejemplo de Mashup permite informar a las personas indicadas si el clima no está propicio para realizar un vuelo. El input del Mashup es la ubicación del aeropuerto donde se quiere obtener el clima. Posteriormente el Mashup retorna la ubicación en el mapa con el estado del clima

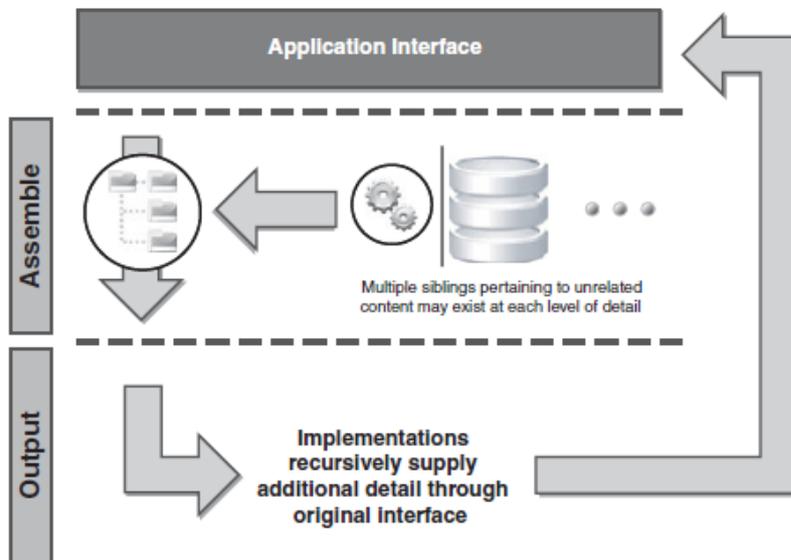


El resultado del Mashup es el siguiente:





## Distributed Drill-Down



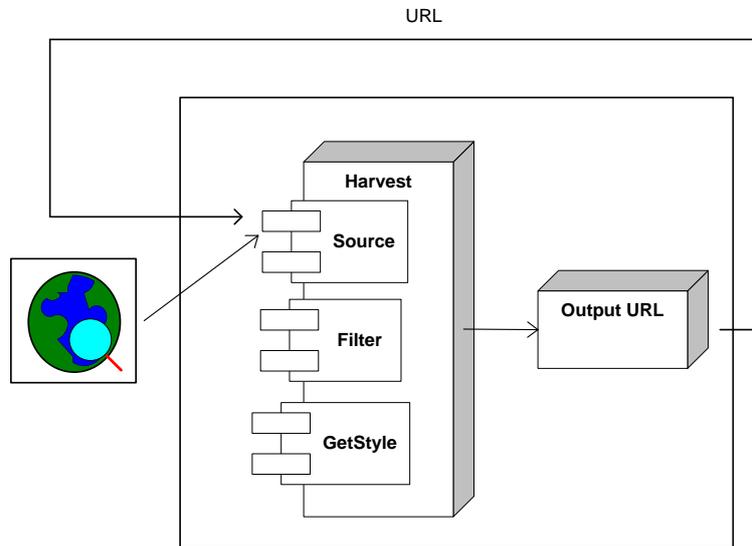
### Problema

Para determinados elementos de la pantalla es necesario ofrecer al usuario posibles links de navegación hacia otras secciones relacionadas con dicho elemento [33]. La idea es que con cada nuevo paso a través de una relación 'vista principal/detalles' un usuario está caminando cada vez más profundo en los detalles de un objeto específico. En esta técnica la interfaz de usuario se compone de dos partes separadas. La primera parte presenta una visión principal de los datos conocidos. La vista principal muestra sólo la información suficiente para que cada tema que interpreta pueda ser identificado individualmente. La segunda parte de la interfaz de usuario presenta información más específica (conocida como la vista de detalle). Cuando se selecciona un elemento de la vista principal, la vista de detalle muestra material adicional para la edición o revisión. El proceso de navegar a través de múltiples 'vista principal/detalle' se conoce como 'Drill-Down'.

### Solución

El patrón 'Drill-Down' (DDD) proporciona información 'vista principal/detalle' a través de la interfaz de datos que no se almacena en el mismo sistema. Por esta razón, el patrón DDD se refiere a veces como 'agregación de contenido con navegación' (CAN). Esta tarea requiere que la aplicación DDD vaya un nivel debajo de cada elemento que se selecciona para determinar si existen otros materiales relacionados [33]. Las conexiones entre los sistemas pueden ser real o puramente conceptual.

## Diagrama Uml:



## Fragilidad



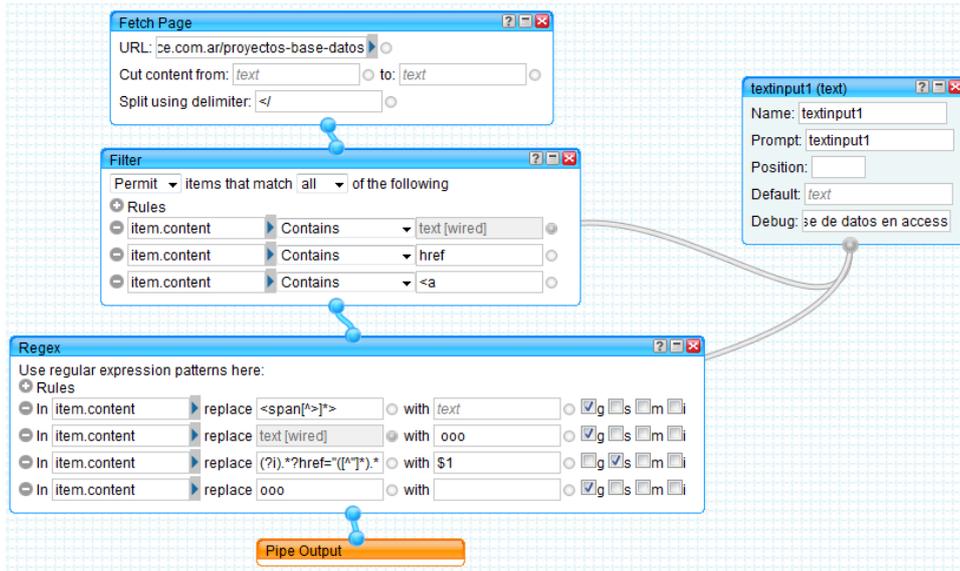
El patrón DDD recibe una puntuación baja de fragilidad. Suponiendo un evento que rompe la conexión entre los sistemas automatizados, todavía es posible llevar a cabo operaciones Drill-Down hasta que el Mashup es corregido. El riesgo principal es que una implementación falle al indicar la presencia de más detalles de los que realmente existen.

## Ejemplo

Este ejemplo permite navegar 'http://www.trabajofreelance.com.ar/' buscando por los links de 'Base de Datos', retorna las url que aplica a dicha búsqueda. En el ejemplo el Mashup retorna la url "http://www.trabajofreelance.com.ar/proyectos-base-datos' . Luego si se reingresa al Mashup la url obtenida y el string de búsqueda 'Recuperar base de datos en access' el Mashup retorna 'http://www.trabajofreelance.com.ar/proyecto-456110/' el cual tiene toda la información del proyecto necesitado.

url del Mashup:

[http://pipes.yahoo.com/pipes/pipe.edit?\\_id=940403a918f839bbd21e36b2f738b573](http://pipes.yahoo.com/pipes/pipe.edit?_id=940403a918f839bbd21e36b2f738b573)



## Usos Conocidos

Youcalc visualiza oportunidades, casos y cuentas en un informe de gráfico interactivo. Permite drill-down y agrupación por tipo, origen, estado, estatus, industria, país. Genera informes con la lista de clics en su cuenta salesforce.com.

APIs

Salesforce.com

Creado

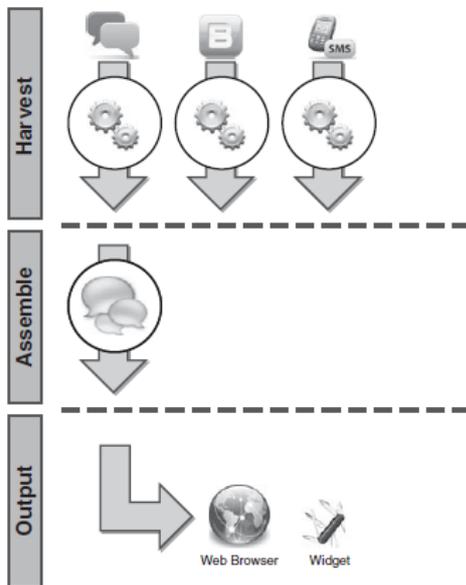
26 Apr 2009

URL

[http://www.youcalc.com/pubapp ...](http://www.youcalc.com/pubapp...)



## Communication and Collaboration



### Problema

La sobrecarga de información hace referencia a la dificultad de una persona en entender un problema y en tomar una decisión que puede ser causada por la presencia de demasiada información [27]. Como consecuencia de la gran cantidad de tecnologías que han surgido, nuestras redes personales se amplían, y nos inundan con alertas de los múltiples dispositivos, productos, y plataformas a las que todos estamos conectados.

En la empresa la información es el producto natural de un negocio. Cuando la gente intercambia datos entre ellos, los principios generales detrás de crowdsourcing ayudan a prevenir que el material útil sea sofocado por el volumen de las comunicaciones (34). Estas situaciones conducen a la información de baja carga, lo cual también puede ser perjudicial.

### Solución

Los Mashups se pueden utilizar para proporcionar un entorno en la que la persona que necesita la información y la persona (o personas) que tiene la información estén conectadas a través de comunidades concretas y dinámicas [34]. Este sistema reduce al mínimo las ocasiones en las que una persona pregunte a una amplia audiencia. Este resultado es una de las principales diferencias entre el patrón 'Communication and Collaboration' (C & C) y el patrón de Portal Enabler. Desafortunadamente, el correo electrónico, la mensajería instantánea y otros servicios se han convertido en productos en sí mismos, cuando en muchos casos se debe girar en torno a proyectos específicos. El patrón de C & C se describen los métodos para organizar estos recursos en canales de situación.

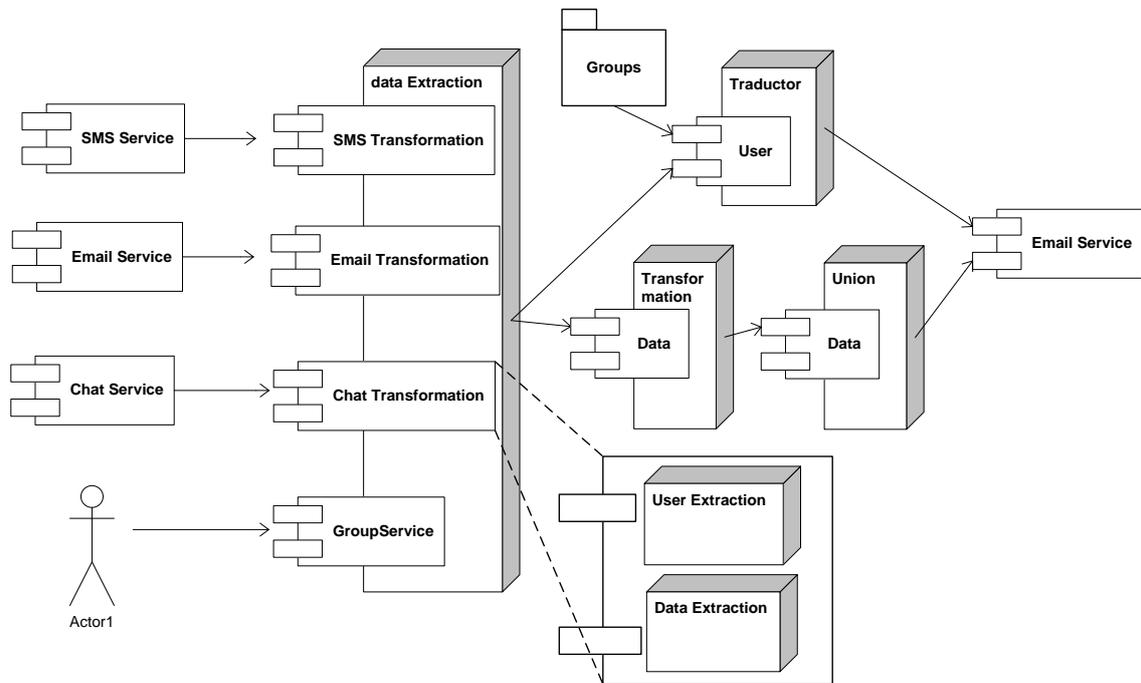
## Fragilidad:



El incumplimiento a las comunicaciones importantes puede tener consecuencias lamentables como la propagación de mensajes no deseados a un público amplio. La situación es análoga a cómo el cerebro humano maneja la información. Las sinapsis son el mecanismo del cerebro de la comunicación entre las neuronas. De acuerdo a esto es igualmente importante que el cerebro a veces 'pode' conexiones de uso poco frecuente.

## Ejemplo

Como se mencionó anteriormente se puede utilizar Mashup como herramienta de integración de mensajería. De este modo los Mashups podrían ser una posible solución a la sobrecarga de información de todos los posibles canales de comunicación de una empresa. Lamentablemente en yahoo pipes todavía no hay componentes que permitan la integración con mensajería tipo mails/chats, aunque una posible alternativa sería la utilización de webservices para poder integrarlos. Este modelo representa una modelo del patrón 'Communication and Collaboration':



## Usos Conocidos

ChatStat es un software de chat en vivo que utilizan las empresas para conectarse directamente con sus clientes y prospectos. Al integrar la API de Jigsaw, ChatStat proporciona el nombre, cargo, empresa y datos de contacto de los visitantes del sitio en cada charla.

APIs:

Jigsaw

Creado

23 Sep 2009

URL:

<http://www.chatstat.com/>

**Engage. Interact. Succeed.**  
Use ChatStat to enhance your website.

**Free Download**  
Try ChatStat free for 30 days!

**\$20 Pricing Guide**  
Live Chat for as low as \$20/mo

**Demonstration**  
Watch a quick demo of ChatStat

**Live Chat** **Web Analytics** **Remote Access** **Translation** **VoIP Callback** **IM Support**

**SELF-HOSTED ENTERPRISE EDITION**  
Large-scale customers, you can now host ChatStat on your servers!

A quick overview of ChatStat's main features:

**Live Chat Software**  
Live chat with visitors on your website using an intuitive instant messenger interface. Boost customer confidence, customer satisfaction, and customer loyalty.

**Web Analytics**  
ChatStat will collect and graph website data allowing you to get specific reports you need. Search Engine Keywords, Operator Productivity, Page Views & more!

**Remote Desktop Screen Sharing and Control**  
Provide effective assistance to your customers by viewing or controlling their computer screen

Let us call you! Just enter your phone number below and click to call.

+1 [Phone Number Field]

**Click To Call**

**How It Works**  
A step-by-step walkthrough about what ChatStat does.

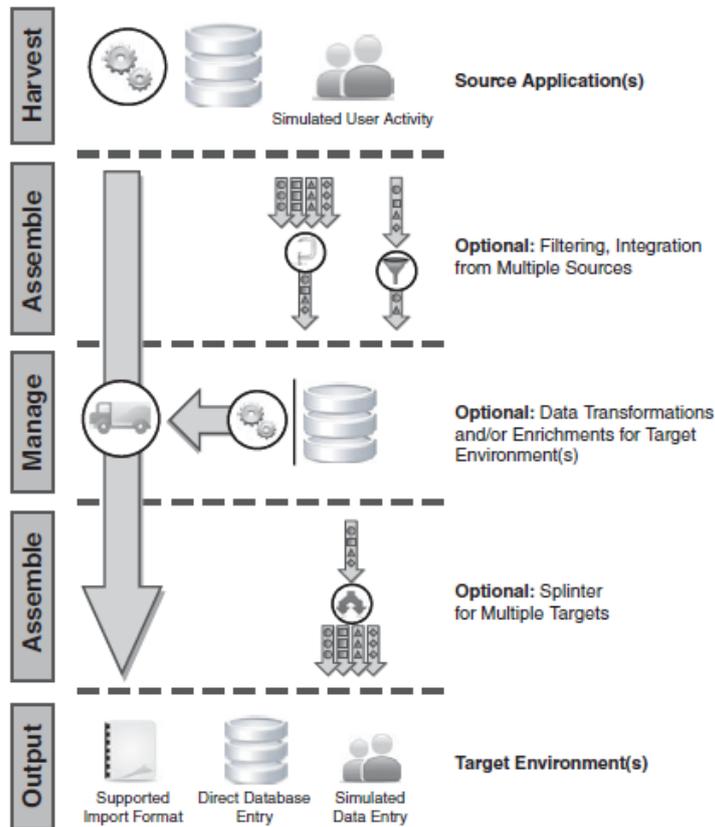
**The ChatStat Blog**  
Read about what's going on with our Development team.

- Capítulo 6 -

# Manage Patterns



## Content Migration



### Problema

Una ventaja competitiva constituye una destreza o habilidad que logra desarrollar una empresa y que la coloca en una posición preferencial a los ojos del mercado [35]. Cuando llega una herramienta innovadora al mercado, esta rápidamente puede llegar a ser muy popular. El responsable de la innovación se dice que tiene la "ventaja del primer movimiento." Si la empresa realiza una estrategia exitosa, puede establecerse como fuerza dominante sobre sus competidores. Es sólo cuestión de tiempo antes de que surjan opciones alternativas. Sus productos gozan la ventaja de ser el "segundo movimiento" porque, pueden aprender de los errores de sus antecesores y utilizar este conocimiento para proporcionar ofertas superiores.

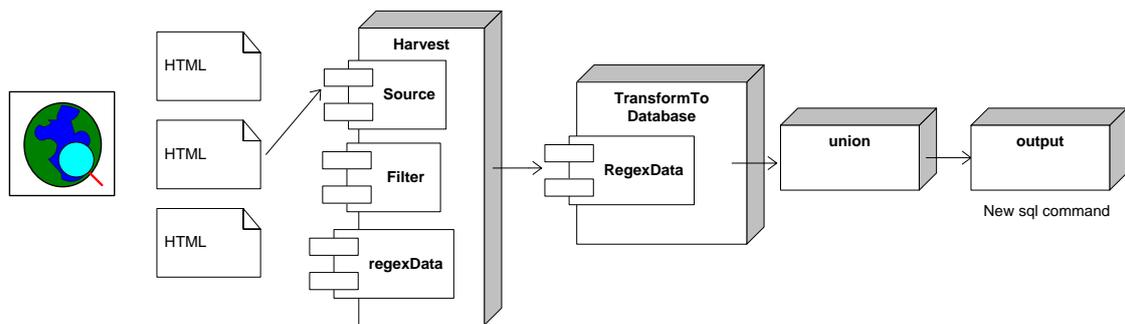
Si el usuario invirtió una gran cantidad de esfuerzo para abastecer con datos el sistema original, los costos en cambiar a un nuevo producto es un problema. Para que una ventaja competitiva sea realmente útil es necesario que sea sostenible; es decir, que la empresa pueda mantenerla durante cierto tiempo [35]. Una técnica que utilizan muchas empresas para mantener su cuota de mercado es dificultar el cambio de su producto al de los rivales.

## Solución

Un vendedor puede construir su herramienta de tal manera que sea difícil migrar a los productos del competidor. Una herramienta de integración de información puede ser capaz de extraer los registros necesarios de la base de datos de una aplicación, siempre que sea posible descifrar la estructura del sistema original. En el peor de los casos se pueden revisar los datos en el sistema original y reingresarlos manualmente.

Los Mashups pueden ofrecer una mejor solución. No importa qué función realice la aplicación original, la información que contiene debe estar accesible a sus usuarios. El patrón 'Content Migration' busca explotar este vacío legal. Mediante el patrón 'Infinite Monkeys' los datos se extraen de una forma metódica. El siguiente paso es copiar este contenido en su nuevo hogar. Si hay suerte, el sistema de destino proporcionará algún tipo de funcionalidad de "importación". Si no lo hace, los Mashups una vez más pueden dar la respuesta mediante el reingreso "manual" de los recursos extraídos.

## Diagrama Uml:



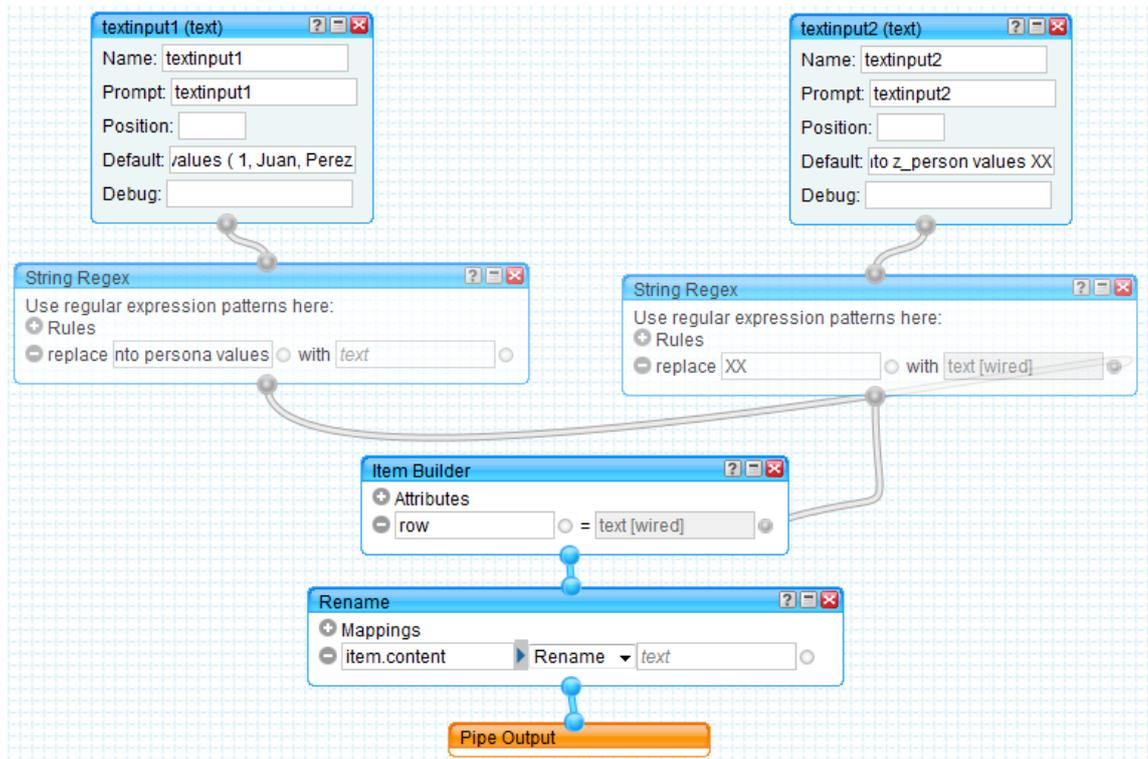
## Fragilidad



El problema principal del patrón es que se pierda información. La traducción incorrecta es otro problema. Luego de la migración se tiene que realizar una investigación para asegurar que la información esperada ha sido recibida correctamente en su nueva ubicación.

## Ejemplo

Una ejemplo de un Mashup 'Content Migration pede ser el siguiente'. Este Mashup permite traducir los inserts de la tabla PERSONA y transformarlos a inserts de otra tabla llamada Z\_PERSON



## Usos Conocidos

Amanda Enterprise es una versión certificada, probada, y con el apoyo de la herramienta de recuperación Amanda. Amanda puede utilizar Amazon S3 para copias de seguridad, archivar y recuperar cualquier cantidad de datos, en cualquier momento y desde cualquier lugar en Internet.

APIs

Amazon S3

Creado

02 Jan 2008

URL

<http://www.zmanda.com/backup-Amazon-S3.html>

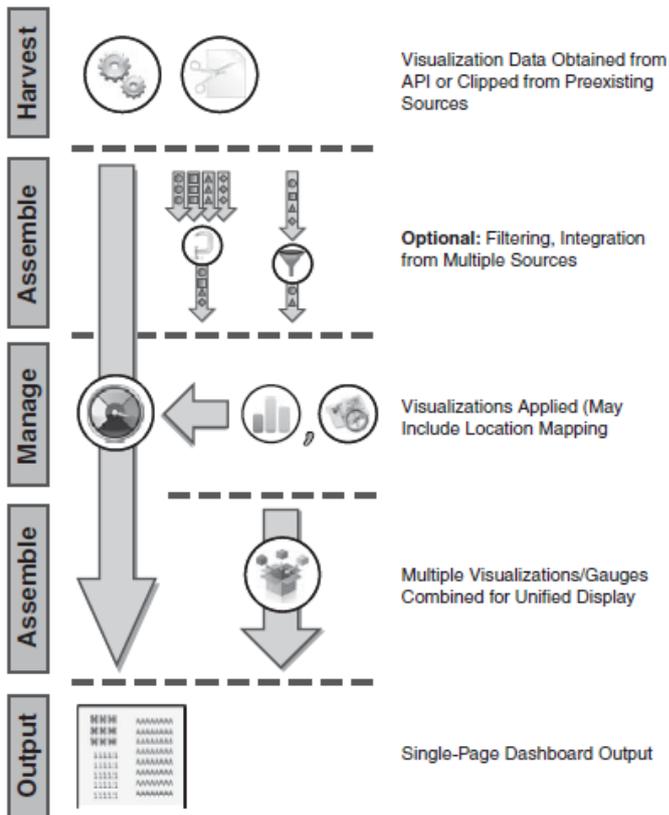
Amanda Enterprise Version 3.1	
<b>Supported OS for Amanda Enterprise Backup Server</b>	
<b>Linux</b>	
<ul style="list-style-type: none"> <li>• Red Hat Enterprise Server 3, 4 (AS, ES, and WS)</li> <li>• Red Hat Enterprise Linux 5, 6 (server and desktop)</li> <li>• CentOS 3, 4, 5, 6</li> <li>• Fedora 6, 7, 8, 9, 10, 11, 12, 13</li> <li>• Oracle Enterprise Linux 3, 4, 5, 6</li> <li>• SUSE Linux Enterprise Server 9, 10, 11</li> <li>• OpenSUSE 10.x</li> <li>• Debian 3.1, 4.0, 5.0</li> <li>• Ubuntu 6.10. 8.04.1. 8.10. 9.04. 9.10. 10.04. 10.10</li> </ul>	
<b>Solaris</b>	
<ul style="list-style-type: none"> <li>• Solaris 10 (SPARC or x86/ x64)</li> <li>• Solaris 9 (SPARC)</li> <li>• Solaris 8 (SPARC)</li> <li>• OpenSolaris (x86/ x64)</li> </ul>	
<b>Mac OS (on Intel and PowerPC)</b>	
<ul style="list-style-type: none"> <li>• Mac OS X Server</li> <li>• Mac OS X 10.4</li> <li>• Mac OS X 10.5</li> <li>• Mac OS X 10.6</li> <li>• Mac OS X 10.7 (Intel only)</li> </ul>	
<b>MS Windows (32 bit and 64 bit)</b>	
<ul style="list-style-type: none"> <li>• Microsoft Windows 7</li> <li>• Microsoft Windows Vista</li> <li>• Microsoft Windows XP (Pro and Home editions)</li> <li>• Microsoft Windows 2008 Server R2</li> <li>• Microsoft Windows 2008 Server</li> <li>• Microsoft Windows 2003 Server</li> <li>• Microsoft Windows 2000 Server **</li> <li>• Microsoft Windows 2000 **</li> </ul>	
<b>Zmanda Client for VMware vSphere and ESXi</b>	
<ul style="list-style-type: none"> <li>• VMware vSphere 4, 5</li> <li>• VMware ESXi 3.5 or higher (Licensed versions only)</li> </ul>	

Amanda Enterprise Version 3.1	
<b>Supported Application versions for Zmanda Agents</b>	
<b>Oracle Agent</b>	<ul style="list-style-type: none"> <li>• Oracle Server 10g on Linux</li> <li>• Oracle Server 10g on Solaris</li> <li>• Oracle Server 11g on Linux</li> <li>• Oracle Server 11g on Solaris</li> <li>• Oracle Server 11g on Windows</li> </ul>
<b>MS Exchange Agent</b>	<ul style="list-style-type: none"> <li>• Microsoft Exchange Server 2003</li> <li>• Microsoft Exchange Server 2007</li> <li>• Microsoft Exchange Server 2010</li> </ul>
<b>MS SQL Agent</b>	<ul style="list-style-type: none"> <li>• Microsoft SQL Server 2000</li> <li>• Microsoft SQL Server 2005</li> <li>• Microsoft SQL Server 2008</li> <li>• Microsoft SQL Server 2008 R2</li> <li>• Microsoft SQL Server 2012</li> </ul>
<b>MS Office SharePoint Server (MOSS) Agent</b>	<ul style="list-style-type: none"> <li>• SharePoint 2010</li> <li>• MOSS 2007</li> <li>• WSS 3.0</li> </ul>
<b>Postgres Agent</b>	<ul style="list-style-type: none"> <li>• PostgreSQL 8.x, 9.x</li> <li>• PostgreSQL Plus 8.3</li> </ul>



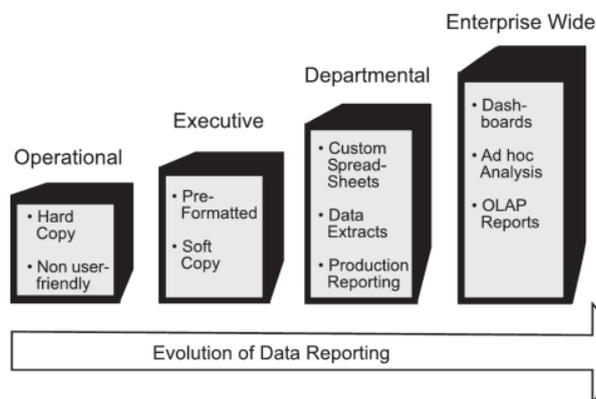
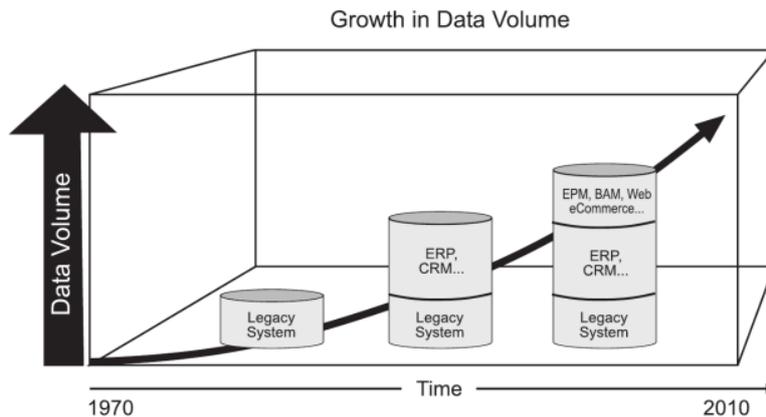
## Dashboard



### Problema

Un principio de gestión bien establecido es que no se puede gestionar lo que no se puede medir. También es verdad que no se puede controlar bien lo que no se puede monitorear. Los 'Dashboards' deben proveer una visibilidad clara para poder dirigir a través de las espesas nubes de sobrecarga de datos y la falta de visión. En los años tempranos del siglo 21 convergieron varios pensamientos que complementaron la vieja búsqueda de 'la información correcta en el tiempo correcto'. Los 'Dashboards' son la nueva cara del emergente campo de la gestión de información.

En retrospectiva, Los 'Dashboards' parecen reflejar el curso natural de progresión en la búsqueda de mejor información y mejor toma de decisiones (36).



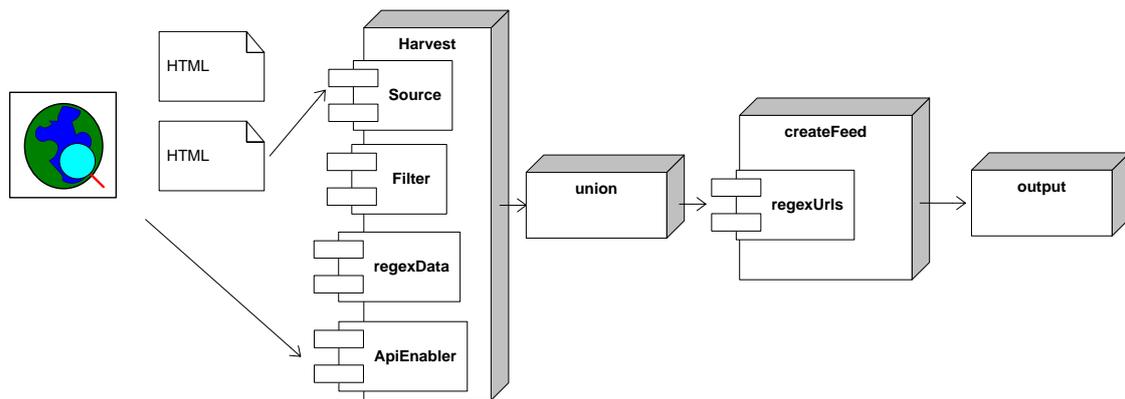
## Solución

Los Dashboards de negocio ofrecen una visión concisa de la información importante que se puede controlar de un vistazo. El contenido exacto varía según la aplicación, pero los retos son claros. Los hechos del poder del Dashboard provienen en de una gran cantidad de sistemas potencialmente relacionados. Una vez obtenidos los datos, estos deben ser visualizados, analizados e incorporados en la pantalla final. Un Dashboard debe tener las siguientes características [36]:

- Intuitivos: Los usuarios finales no tienen que requerir de una gran curva de aprendizaje.
- Basados en Web: Los usuarios deben ser capaces de acceder al Dashboard a través de la web.
- Seguros: Los administradores de sistemas deben poder administrar la seguridad más fácilmente y reducir y trackear el acceso incorrecto.
- Escalable: El sistema debe ser escalable para añadir nuevas aplicaciones.
- Obediente a la industria: El Dashboard debe ser capaz de integrarse con los bases de datos actuales y trabajar con diferentes servers.
- Tecnología abierta: El software no debería ser propietario.
- Soporte: Debería ser fácil de mantener por personal de IT con poca experiencia en Dashboards.
- Rentable: El total del costo de mantener el Dashboard debería ser justificado por los beneficios de sus resultados.

Entonces, ¿dónde entra el Mashup? Desde una perspectiva de adquisición de datos en general, una variedad de patrones de Mashup se pueden utilizar para obtener la información que es imposible de obtener de otro modo, especialmente si el Dashboard controlará recursos externos. El siguiente paso es preparar la información para su análisis. La capacidad de un Mashup para convertir los datos a un formato común hace que este trabajo sea más fácil. El tratamiento posterior puede ser realizado uniéndolo en un producto analítico. El paso final es crear la visualización que se presentará al usuario.

## Diagrama Uml:

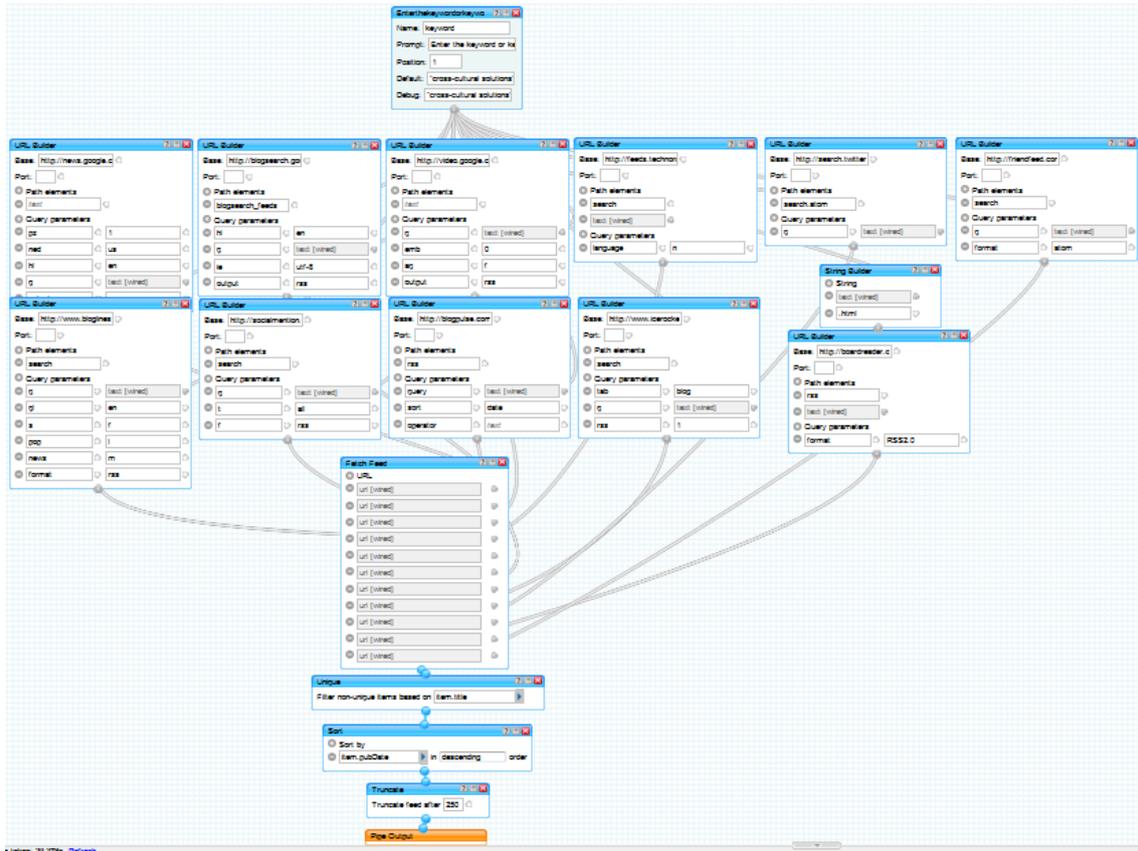


## Fragilidad

Este patrón comparte muchos de los mismos riesgos asociados con una instancia de 'Content Migration'. Otro riesgo que debe tenerse en cuenta es que si se presentan datos incorrectos o inexactos se conduce a malas decisiones o mala gestión. Un Dashboards que señale incorrectamente "Todo está bien" puede causar daños graves.

## Ejemplo

A partir de este Mashup se pueden controlar las noticias u otras palabras clave importantes para un usuario. Ingresando los términos de búsqueda se consigue la información a partir de diarios, blogs, wikis, foros de discusión y sitios para compartir videos.



## Usos Conocidos

2012twit.com es un dashboard de Twitter para las elecciones presidenciales del 2012. El sitio utiliza la API de Twitter para recolectar mensajes de Twitter por el presidente Obama, 32 candidatos del Partido.

APIs

Twitter

Creado

04 Apr 2011

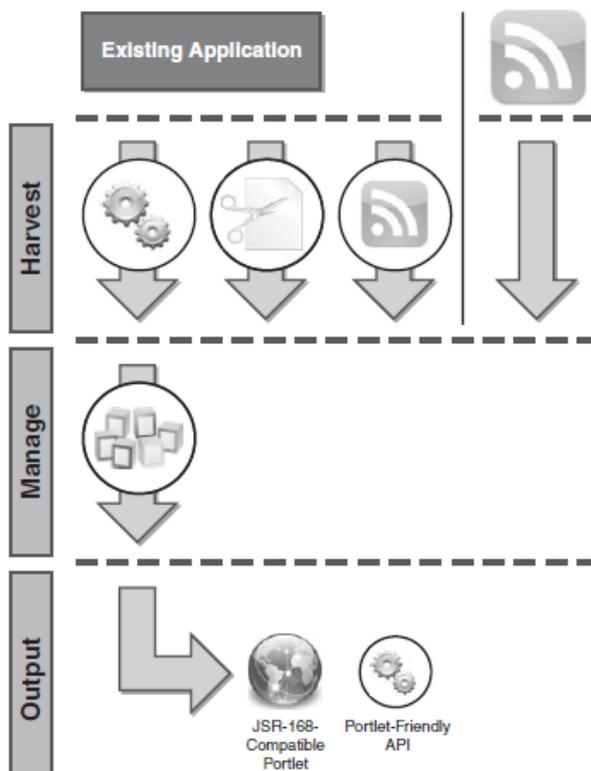
URL

<http://uniteblue.com/>





## Portal Enabler



### Problema

El mercado de los portales ha evolucionado en un amplio segmento de la industria. Un portal es un sitio web que ofrece al usuario, el acceso a una serie de recursos y servicios relacionados a un mismo tema. Incluye: enlaces, buscadores, foros, documentos, aplicaciones, compra electrónica, etc.

Principalmente un portal resuelve necesidades de información específica.

Un portal es el punto de partida de un usuario que desea entrar y realizar búsquedas en la web u obtener información importante de él.

Un portal de Internet puede ser, por ejemplo, un Centro de contenido intermediario entre compradores y vendedores de rubros específicos. Toda esta información no necesariamente está contenida dentro del mismo portal, porque el portal, normalmente, se encarga de centralizar enlaces en una forma fácil y organizada que facilite la navegación dentro de un tema [37].

### Solución

Los Mashups pueden mejorar y extender la funcionalidad del portal permitiendo incorporar nuevos contenidos que de otra forma seria inaccesible.

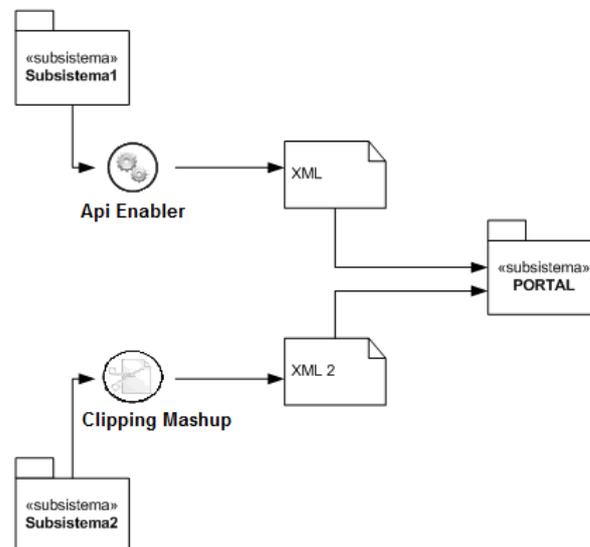
Se puede emplear varios métodos diferentes para la migración de contenido existente a un portal mediante Mashups. Una de las técnicas consiste en crear un

XML o RSS feed. Muchos frameworks de portales aceptan estos formatos. Cuando a contenido estático se refiere, ésta puede ser la forma más fácil de incorporar. Otra manera se basa en el uso directo del patrón 'API Enabler' para hacer los datos disponibles para los desarrolladores IT. Luego este equipo puede crear código que se ajuste al estándar del portal. Una tercera técnica depende de las capacidades de las herramientas Mashup que se tengan disponibles. Tanto el personal de IT como los usuarios finales pueden crear Mashups que sean fáciles de instalar.

## Fragilidad

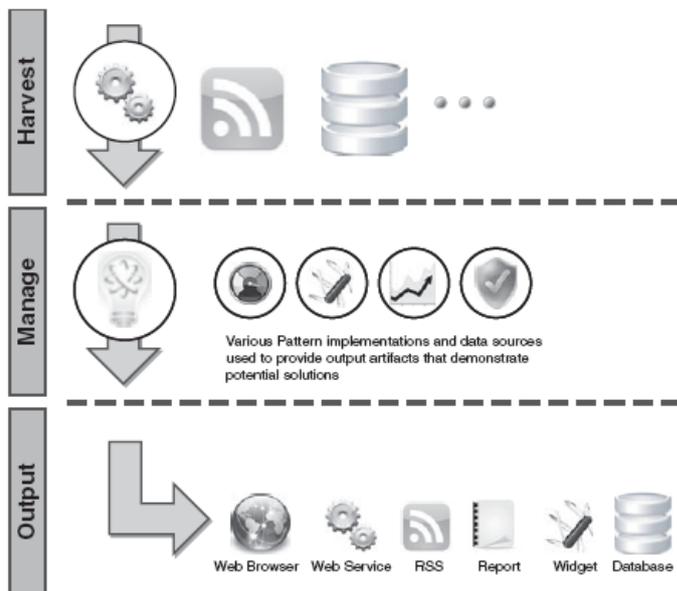
Mediante el recorte, varias porciones de un sitio puede extraerse y ser integradas en un marco de portal. La arquitectura subyacente es un poco delicada, sin embargo, un servidor en algún lugar del entorno de despliegue ejecuta un agente de Mashup que suplanta a un navegador Web. Nadie se observa este proceso, ya que el agente recupera y procesa el HTML puro sin intervención del usuario. En última instancia, las secciones requeridas son "cortadas" fuera y alimentadas al marco de portal. Pero... ¿Y si el contenido ha contiene JavaScript? , ¿los vínculos o los forms apuntan al servidor original? Ahí es donde surge la fragilidad de este enfoque. A pesar de que se muestra al usuario como si estuviera interactuando con el sitio original, en realidad una capa Mashup intermediaria maneja todas las peticiones y las respuestas.

## Ejemplo





## Quick Proof-of-Concept



### Problema

A pesar de décadas de estudio todavía no es posible estimar de forma fiable el esfuerzo desarrollo de software. Si se está construyendo una nueva pieza de software idéntica a uno que ya se ha escrito, entonces debería ser capaz de predecir cuánto tiempo va a tomar desarrollarla. Por supuesto, las empresas pocas veces reescriben completamente los sistemas que ya tienen. Podemos utilizar nuestra experiencia para medir la cantidad de esfuerzo que se gasta, pero en última instancia, solo es una conjetura.

La creación de prototipos es un método para abordar los desafíos de la estimación. Un prototipo puede variar desde una de baja fidelidad, o representar todo o parte de una interfaz de usuario final. El prototipo puede representar el simple comienzo de un sistema complejo [38].

Una versión inicial de un sistema puede dar una idea de los posibles obstáculos que saldrán a la superficie permitiendo refinar las especificaciones y estimaciones. Esto nos lleva de vuelta a donde empezamos:

¿Cómo se puede maximizar la cantidad limitada de recursos?

¿Cómo se puede determinar qué soluciones son dignas de exploración inicial y posterior financiación?

### Solución

Los Mashups no deben proporcionar apoyo continuo a tareas críticas, pero es perfectamente aceptable el uso de Mashups para validar una idea que requiere una inversión importante. Algunos ejemplos son los siguientes:

- Se puede crear un Mashup prototipo para obtener retroalimentación de los usuarios sobre la funcionalidad propuesta.

- Los usuarios finales dan soluciones basadas en el Mashup.
- Se pueden utilizar los Mashups para medir el esfuerzo de integrar aplicaciones dispares.
- Los usuarios pueden testear flujos de trabajo creando de forma temporal Mashups basados en agentes de enrutamiento.

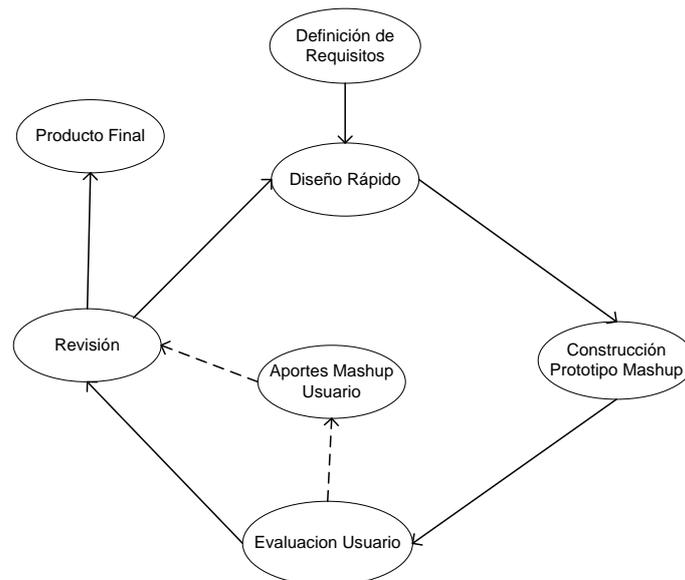
El producto final no debería utilizar Mashups, o este pueden jugar un papel más limitado. El personal de IT y los usuarios suelen trabajar juntos para construir el primer prototipo de la aplicación. Por desgracia, esto puede tomar mucho tiempo y consumir muchos recursos afectando al producto final. Este es un estado final casi inevitable, dada la brecha que separa a los desarrolladores y usuarios. Los Mashups pueden llenar este vacío de comunicaciones. Debido a la facilidad de los Mashups de uso, los usuarios pueden construir sus propias soluciones (o partes de la misma) y proporcionarlos al departamento de IT como una "porción real" de diseño. Debido a que los usuarios finales toman un papel más activo en la solución final, se debe esperar una mayor satisfacción en el producto final.

## Fragilidad

Por definición, no se pretende que una prueba de concepto sea un producto totalmente funcional o una solución de largo plazo. Una vez que el Mashup PoC ha cumplido con las metas, el trabajo debe realizarse la aplicación formal. Teniendo en cuenta su vida útil prevista, es tentador dar al PoC la puntuación más baja fragilidad. Pero en muchas circunstancias la fragilidad de un PoC se ve ensombrecida por la emoción creada por tener una nueva solución rápida.

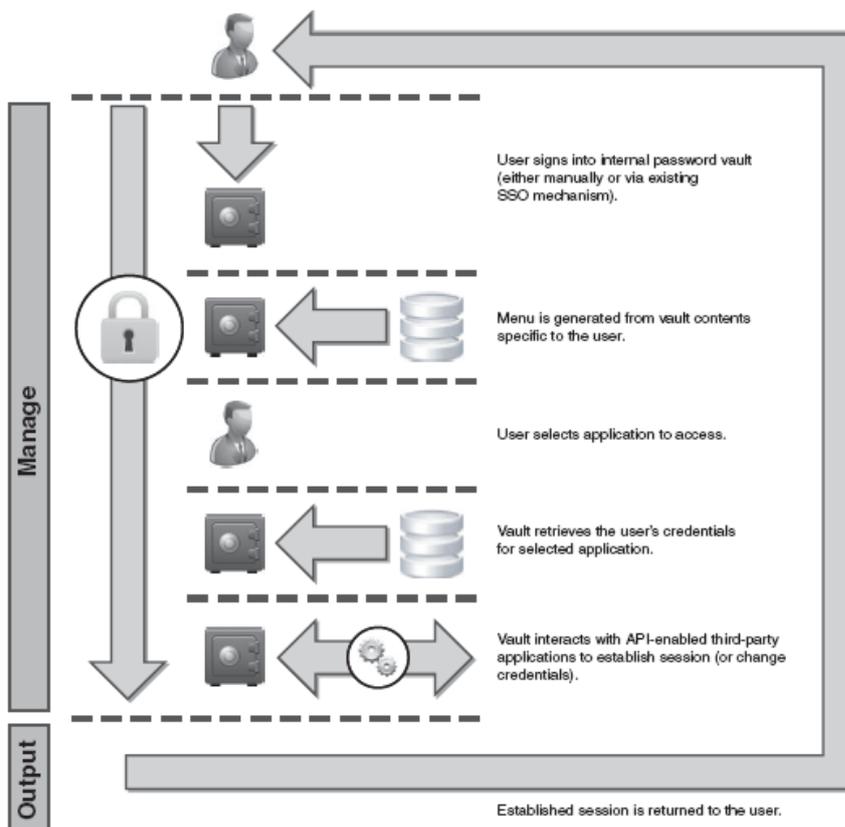
## Ejemplo

El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles para el cliente o el usuario final. Este diseño conduce a la construcción de un prototipo, el cual es evaluado por el cliente para una retroalimentación:





## Single Sign-on



### Problema

Una línea de defensa en los sistemas de una empresa es la autenticación. De esta manera un usuario requiere proporcionar un inicio de sesión para ingresar a los datos confidenciales, y cada cierto tiempo, el sistema solicita un cambio de password a fin de mantener seguros los datos.

Un usuario que requiere con frecuencia proporcionar información de inicio de sesión puede sufrir fatiga de passwords . Esta dolencia incluye escribir las contraseñas para que no se olviden, o la elección de passwords sencillas.

Estudios han demostrado que las personas tienden a reutilizar passwords a través de sistemas. Muchas empresas han tratado de equilibrar las fuerzas mediante un procedimiento de autenticación que habilita al usuario para acceder a varios sistemas con una sola instancia de identificación (SSO).

Las soluciones de Single Sign-on nos permiten proporcionar un registro único de credenciales para múltiples sistemas. Esto permite tener a la gente autenticada al producto de inicio de sesión el cual dará inicio sesión automáticamente a los otros productos si lo requieren otros productos [39]. Desafortunadamente, este enfoque no hace nada para hacer frente a los problemas de autenticación de los productos y servicios externos. Una empresa no puede imponer su solución interna de SSO en los proveedores externos, pero no puede permitirse el lujo de aceptar la debilidad

de seguridad tampoco. Un enfoque consiste en crear una "bóveda de passwords" interna, donde los usuarios puedan almacenar los nombres de cuenta y códigos de acceso que han creado para los diferentes servicios. Una bóveda de passwords puede ser capaz de generar automáticamente contraseñas seguras y recordar a los usuarios cuando cambiarlas.

## Solución

Las capacidades de entrada de datos y extracción permiten a los Mashups suplantar la actividad de los usuarios. Si se establece una bóveda, un Mashup con la contraseña correcta podría firmar en este repositorio en nombre del usuario y obtener acceso a todas las credenciales específicas del lugar que contiene.

Los beneficios de implementar una bóveda de datos son los siguientes [40]:

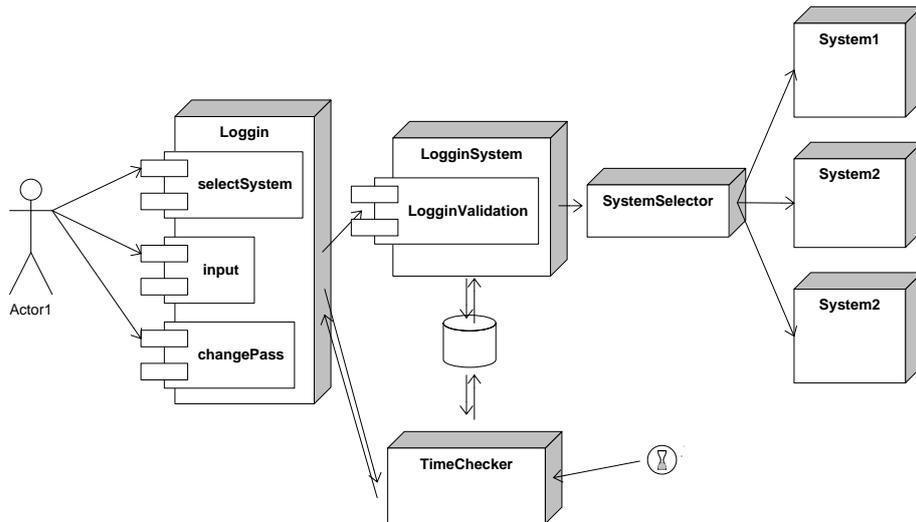
- Reduce el éxito del phishing\*, porque los usuarios están capacitados para ingresar la contraseña en todas partes sin pensar.
- Reduce la fatiga de contraseña, de diferentes nombres de usuario y combinaciones de contraseñas.
- Reduce tiempo empleado para ingresan contraseñas para volver siempre la misma identidad.
- Soporta la autenticación convencional, tal como credenciales de Windows.
- La reducción de costes de IT debido a un menor número de llamadas a la mesa de ayuda acerca de las contraseñas bloqueadas.
- Seguridad en todos los niveles de entrada / salida / y acceso a los sistemas sin el inconveniente de reingresar el usuario.
- Informes centralizados para el cumplimiento de la adhesión.

Los pasos para proporcionar acceso seguro son las siguientes:

- Establecer una bóveda de contraseñas.
- Conectar la bóveda de contraseñas a la plataforma de la empresa. Esto se puede hacer directamente o mediante el uso de un Mashup.
- Extraer de la URL de destino o la ubicación para cada elemento de la bóveda y construir un "menú de aplicaciones" para los usuarios finales.
- Cuando se hace click en un elemento del menú, el Mashup pasa las credenciales del usuario de SSO a la bóveda y extrae el inicio de sesión del usuario correspondiente.
- Finalmente el Mashup utiliza esta información para iniciar su sesión en nombre del usuario.

\***Phishing** es un término informático que denomina un tipo de delito encuadrado dentro del ámbito de las estafas cibernéticas, y que se comete mediante el uso de un tipo de ingeniería social caracterizado por intentar adquirir información confidencial de forma fraudulenta

## Diagrama Uml:

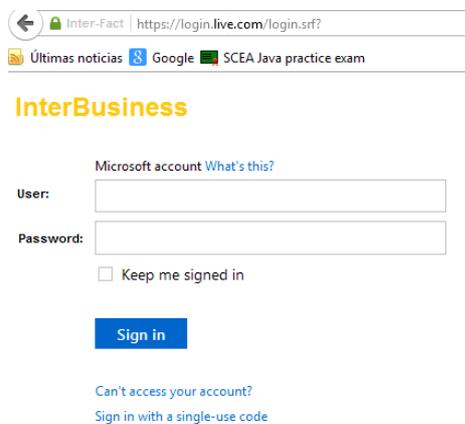


## Fragilidad

Una de las fragilidades de este enfoque es que si los usuarios necesitan tener acceso al sistema fuera de la red de la empresa, no serán capaces de conseguirlo porque sus credenciales están encerradas dentro de la bóveda de la intranet corporativa. Qué pasa si un usuario solicita al sistema externo que la contraseña "perdida" se la envíe por correo electrónico? Esto presenta otra debilidad potencial, porque si la cambia, el vínculo entre la bóveda y la aplicación se rompe. Otra desventaja del gestionado de seguridad por Mashup se produce cuando el entorno de destino cambia de repente su diseño. Ya que el Mashup podría accidentalmente introducir los datos privados de autenticación en un campo público.

## Ejemplo

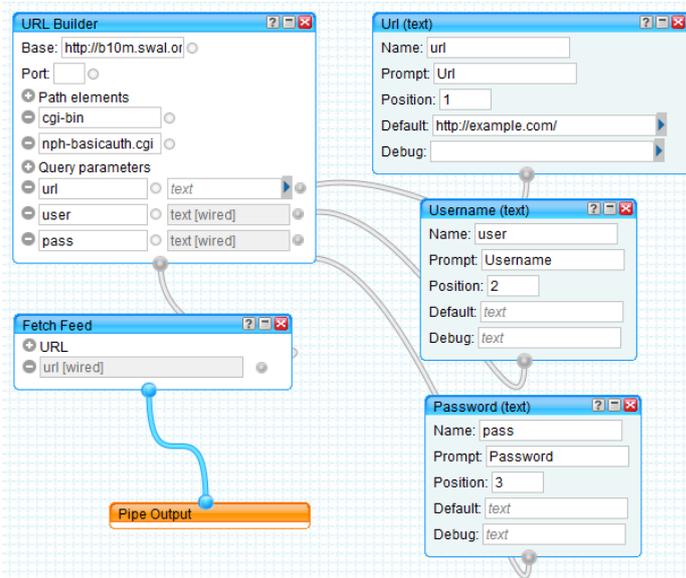
En este ejemplo se muestra como se puede aplicar Mashups para la utilización de una bóveda de contraseñas. Mediante un logueo a la página principal se ingresa a la bóveda de contraseñas. La cual administra las passwords de todos los sistemas al que el usuario puede loguearse.



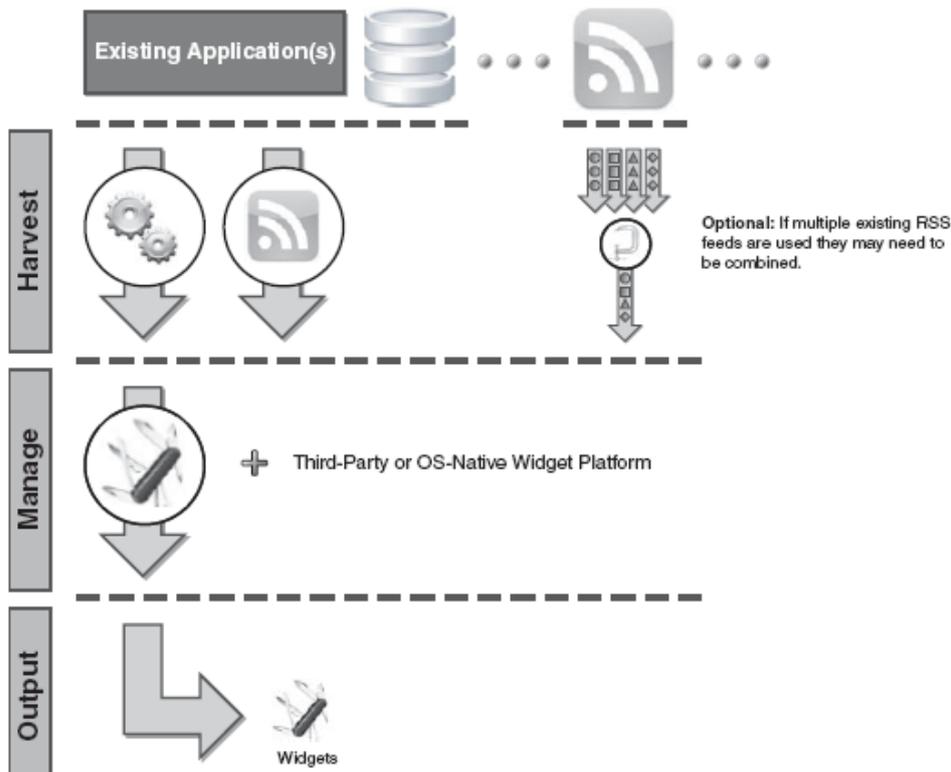
Una vez ingresado al sistema la bóveda retorna todos los sistemas al que el usuario puede loguearse.



Una vez que el usuario selecciona la empresa la bóveda retorna la url y el usuario y contraseña que administra para loguearse automáticamente a la página del sistema



## Widget Enabler



### Problema

Los widgets básicamente son mini-aplicaciones que pueden ser embebidas en una página web agregando una porción de código. Es hace al widget reusable y portable. Incluso algunos widgets, modificando el la porción de código embebido, se pueden configurar para alterar su comportamiento. El termino widget no se refiere al contenido en sí, sino al packaging. En otras palabras el contenido del widget puede ser nada. Los widget les permiten a las personas crear o modificar sitios web, perfiles sociales, o incluso agregar contenido enriquecido a blogs, permitiendo agregar ese código sin tener que haberlo desarrollado [41]. Las empresas que desean empaquetar los datos y aplicaciones en estos componentes de fácil distribución deben averiguar cómo conectar su funcionalidad existente con estas nuevas interfaces. Mientras tanto, se produjo una explosión simultánea de widgets de escritorio gracias a la aparición de plataformas como Apple Dashboard, Google Gadgets, Widgets de Microsoft Vista, y Yahoo Widgets. Los widgets de escritorio atraen a un público diferente al de sus primos de la web. Estos centran la atención en el cumplimiento de las necesidades funcionales específicas en lugar de proporcionar un medio para atraer la atención. El potencial para la comercialización todavía está allí, pero con el énfasis en las preocupaciones personales. En lugar de mantener la enorme infraestructura necesaria para apoyar el framework de un portal, una empresa debe considerar el valor de la adopción de

una plataforma de widgets. Este patrón puede ayudar a la reutilización de los portlets existentes como widgets y permite que tanto el personal de IT como los usuarios puedan crear nuevos widgets.

La naturaleza no estructurada de un contenedor widget permite una flexibilidad sin precedentes en la creación, la conexión y el intercambio de experiencias muy concretas. Usados correctamente, los widgets pueden cumplir con las promesas incumplidas de los portales. Además de reconocer su papel en una estrategia en la sustitución del portal, una empresa puede descubrir otro valor agregado mediante el uso de widgets. Por ejemplo, podría no tener sentido de seguir apoyando estructuras gigantes, si sólo un puñado de características son utilizados por los diferentes empleados. De hecho, esta política podría ser responsable de propagar los problemas relacionados con la calidad y seguridad de la información. Un enfoque mejor podría consistir en proporcionar un widget que exponga sólo el subconjunto de la funcionalidad necesaria para hacer un trabajo.

### Solución

Los "problemas" describen dos categorías principales: widgets basados en web y basado en escritorio. Ambos comparten un conjunto común de problemas.

- La adquisición de contenidos. Widgets y su contraparte más rígida, portlets, encara un desafío común en la adquisición de los datos que mostrara. Se puede obtener una fuente XML o RSS contra el entorno de origen y crear feeds de recursos. Otra opción es habilitar la API (Api-Enabler) un recurso y conectarlo manualmente con la interfaz de presentación que el framework de widgets tiene disponible. Las empresas deben tener en cuenta que los usuarios externos con experiencia pueden reempaquetar esta interfaz en un nuevo widget tan fácilmente como pueden los desarrolladores internos.
- Seguridad. Los widgets son "el vector de ataque más reciente" de software malicioso.
- Personalización. Los usuarios de hoy ya no se conforman simplemente con aceptar los productos que le dan. Un widget de diseño puede minimizar este deseo, pero no eliminarlo totalmente
- Búsqueda unificada / comunicaciones entre widget. Esta última cuestión se aplica principalmente a los widgets internos utilizados como reemplazo de portales corporativos.

Ya sea proporcionando una línea de productos nuevos o de escritorio, los beneficios de Mashups, como parte de una estrategia de widget es clara. Además de hacer los contenidos accesibles, pueden mejorar y ampliar las plataformas de widgets con nuevas capacidades.

### Fragilidad



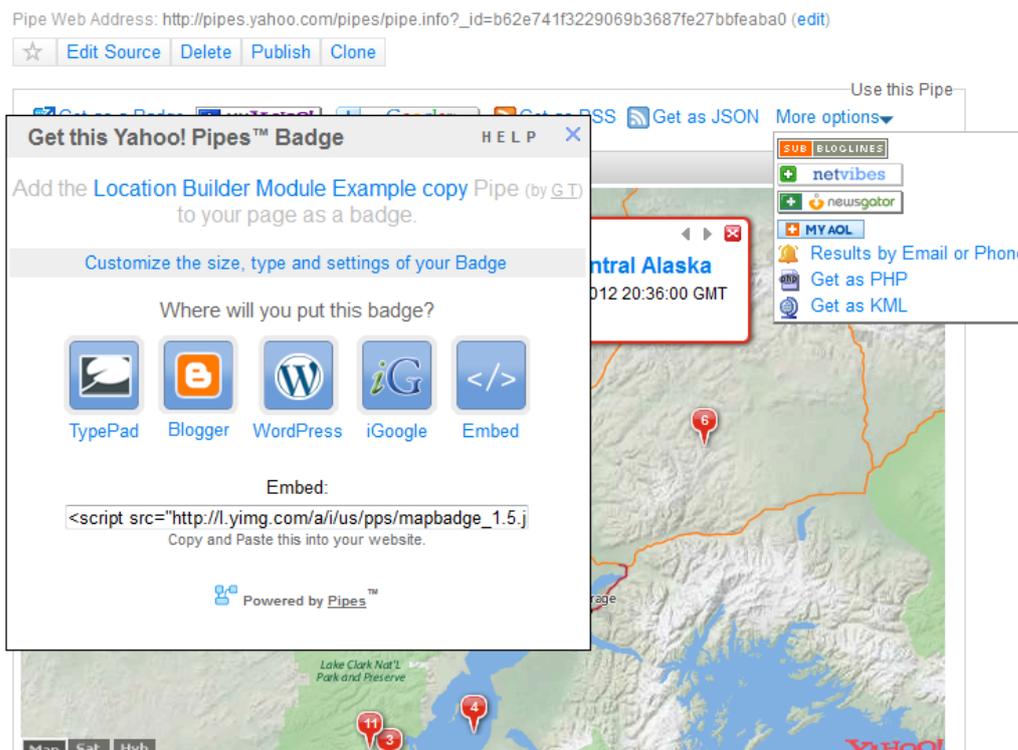
Los widgets no son más ni menos fiables que sus alternativas construidas como aplicaciones de cliente o sitios Web dinámicos. Debido a que las implementaciones del API Enabler y Feed Factory son los agentes principales detrás de las

implementaciones del Widget Enabler, su puntuación se realiza a través de fragilidad estos patrones.

## Ejemplo

Se puede realizar widgets con yahoo pipes mediante badges.

Utilicemos el siguiente pipe de sitios de terremotos. Dicho pipe se puede publicar como badge haciendo click en la opción 'get as badge' en la pantalla de publicación del Mashup. Al realizar dicha acción se presentará una ventana de donde se presentara dicho badge.



En este ejemplo lo publicaremos como 'Embed' el cual nos devolverá el script que debemos agregar en nuestra página web para poder incorporar el widget.

```
<script  
src="http://l.yimg.com/a/i/us/pps/mapbadge_1.5.js">{"pipe_id": "b62e741f3229069b3687fe27bbfeaba0", "_btype": "map"}</script>
```

Descargando la hoja de estilos de CSS para Yahoo Pipes se pueden descargar dos versiones:

- La hoja de estilo primero para usuarios nuevos en CSS - 'Rookies' CSS. Esta hoja de estilo contiene las observaciones al lado de todas las clases CSS y su propiedad.
- La segunda hoja de estilos es para usuarios avanzados de CSS - 'All-Stars'. Esta hoja de estilo sólo incluye comentarios al lado de las clases.

Creando un HTML en blanco y agregando el link haciendo referencia al CSS.

```
<link href="pipesstyle_ENTER_VERSION.css"
rel="stylesheet" type="text/css">
```

Finalmente agregando el script rodeado por tags divs haciendo referencia a la clase pipesstyle

```
<div class=pipesstyle">
<script src="http://l.yimg.com/a/i/us/ppp/listbadge_1.4.js">
{"pipe_id":"8cf4202441470398ea665ec175e73c6c",
"_btype":"list"}</script>
</div>
```

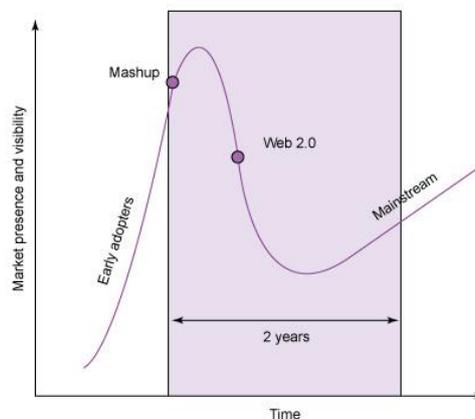
- Capítulo 7 -

# Conclusiones

Vivimos en un mundo que se ha sobre-saturado de información digital y cada vez más las empresas necesitan poder utilizarlo. "Los patrones de Mashup", describen las formas en que las tecnologías Web 2.0 son utilizadas para proporcionar el contexto a las empresas que necesitan y de manera rápida reunir y dar sentido a los contenidos.

En este trabajo se describen los patrones Mashup con el fin de encontrar una nueva forma de ofrecer servicios web y nuevas orientaciones para el diseño de herramientas de Mashup.

Como se muestra a continuación cada mes, más organizaciones abren sus fuentes de datos enriqueciendo el ecosistema Mashup. Bibliotecas de código Mashup o sitios como Yahoo! Pipes y Microsoft Popfly se orientan a los no programadores y añaden nuevas características y servicios [10].



Lo que observamos con la Web 2.0 Mashup es un cambio de paradigma cómo los usuarios utilizan la web, un desarrollo que cuestiona todo lo que se ha desarrollado y aplicado hasta ahora.

Los Mashups tienen el potencial de crear una ventaja competitiva importante al permitir la innovación rápida de procesos de negocio a través de la co-creación de nuevos sistemas que aprovechan la inteligencia colectiva de una organización. También los Mashups resultan eficaces al aportar valor a la empresa con sus beneficios y aspectos únicos. Esto fomenta la investigación que pueda mejorar nuestro conocimiento sobre las estrategias Mashup, modelos de negocio y las tecnologías de interés involucradas.

## Bibliografía y Referencias

- [1] Tim O'Reilly. What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software. Communications & Strategies, no. 65, 1st quarter 2007, p. 17.
- [2] Anderson, P. What is Web 2.0? Ideas, technologies and implications for education. JISC Technology and Standards Watch (2007) (3) Toward a New Paradigm: Mashup Patterns in Web 2.0.
- [4] Raymond Yee, Pro Web 2.0 Mashups Remixing Data and Web Services. Apress publication Date: March 7, 2008 .ISBN-13: 978-1-59059-858-0.
- [5] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, 'Design Patterns Elements of Reusable Object-Oriented Software', Publication Date: Oct 31 1994, ISBN-10: 0201633612, ISBN-13: 978-0201633610, Edition: 1.
- [6] Thomas Erl, Service-Oriented Architecture: Concepts, Technology, and Design, Pearson Education, 2006. ISBN: 813171490X, 9788131714904.
- [7] Martin Fowler, Kent Beck, John Brant, William Opdy, Don Roberts. Refactoring: Improving the Design of Existing Code. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA ©1999. ISBN: 0-201-48567-2.
- [8] Phillips, Joseph. 'PMP Project Management Professional Study Guide', McGraw-Hill, Publication Date: December 24, 2003. ISBN:978-0-07-162672-9.
- [9] Craig S. Fleisher, Babette E. Bensoussan, 'Business and Competitive Analysis'. Published Feb 27, 2007 by FT Press. ISBN-10: 0-13-216158-3
- [10] <http://www.ibm.com/developerworks/lotus/library/mashups-patterns-pt1/>, 'Mashup business scenarios and patterns: Part 1'. Holt Adams, Executive IT Architect, IBM . Published 27 Jan 2009.
- [11] Isaac, Richard E. (1995). The Pleasures of Probability. Publication Date February 24, 1995, ISBN-10: 038794415X | ISBN-13: 978-0387944159.
- [12] Mary DeRosa, 'Principles of Data Mining', editorial 'The MIT Press', año 2001). ISBN (electronic): 026208290X.
- [13] RUEY S. TSAY, University of Chicago . 'Analysis of Financial Time Series, Second Edition', Published in 2005 by Wileyblackwell, ISBN-13: 978-0-471-69074-0.
- [14] Elaine K. Harris , 'Customer Service: A Practical Approach', Published in 2006, ISBN-10: 0130978531.
- [15] Efthymios Kouloumpis, Theresa Wilson, Johanna Moore , 'Twitter Sentiment Analysis: The Good the Bad and the OMG!'. \*Work performed while at the University of Edinburgh, Association for the Advancement of Artificial Intelligence (www.aaai.org).
- [16] Shelley Powers , What Are Syndication Feeds, *Publisher*, O'Reilly Media, Inc., 2008. , ISBN 0596526970.
- [17] Pekka Abrahamsson, Outi Salo & Jussi Ronkai ISSN 1571-5035 Human-Computer Interaction Series, Agile software development methods - Review and analysis, University of Oulu 2002. ISBN 951-38-6010-8.
- [18] Smith, Gene: Tagging : people-powered metadata for the social web. Berkeley, Calif. : New Riders, 2008. - ISBN 0-321-52917-0 ; 978-0-321-52917-6.
- [19] Manish Gupta, Rui Li, Zhijun Yin, Jiawei Han. 'Survey on Social Tagging Techniques', Published in Newsletter, June 2010 ISBN: 013306526X.
- [21] John Karat & John Karat, Human-Computer Interaction Series, Editor Springer London 2010, ISSN 1571-5035.
- [22] Guy Smith-Ferrier, Addison-Wesley Professional. '.NET Internationalization: The Developer's Guide to Building Global Windows and Web Applications'. Published 7 August 2006. ISBN 0-321-34138-4.
- [23] Bert Esselink , 'Practical Guide to Localization', John Benjamins Publishing (2000), ISBN 1-58811-006-0.
- [24] Nora Koch, Matthias Pigerl, Gefei Zhang, and Tatiana Morozova. 'Patterns for the Model-Based Development of RIAs'. Ludwig-Maximilians-Universität München, Germany , Cirquent GmbH, Germany, S.CO LifeScience GmbH, Germany.
- [25] <http://designingwebinterfaces.com/designing-web-interfaces-12-screen-patterns>,

- 'Designing Web Interfaces Principles and Patterns for Rich Interaction', By Theresa Neil.
- [26] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. 'Workflow Patterns'. Department of Technology Management, Eindhoven University of Technology 2002.
- [27] Judith B. Strother, Jan M Ulijn, Zohra Fazal, 'Information Overload. An International Challenge for Professional Engineers and Technical Communicators', Wiley & Sons (2012-07-03), ISBN 978-1-118-23013-8.
- [28] Burrough, P.A. and McDonnell. Principles of geographical information Systems, Oxford University Press (1998).
- [29] C.S.R. PraBHU, M.Tech(IIT Bombay) ,Data WareHousing. Concepts, Techniques, Products and Applications , June 2006, ISBN-81-203-2068-9.
- [30] Norbert Bieberstein, Sanjay Bose, Marc Fiammante, Keith Jones, 'Service-Oriented Architecture Compass: Business Value, Planning, and Enterprise Roadmap', Published in Oct 25, 2005, ISBN: 0-13-187002-5.
- [31] [http://es.wikipedia.org/wiki/Portal\\_%28Internet%29](http://es.wikipedia.org/wiki/Portal_%28Internet%29), 'Portal (Internet)'. Categoría: Portales de Internet, Licencia Creative Commons Atribución Compartir Igual 3.0.
- [32] SE-SNPMAD. 'Sistema Nacional para la Prevención, Mitigación y Atención de Desastres'. Plan Nacional de respuesta 30/06/2001 Nicaragua.
- [33] Ingeniero Diego Oscar Campana. 'Tesis Especificación de Interfaces y Patrones RIA'. UNLP, Tesis de maestría , Fecha de publicación 2009.
- [34] James Surowiecki, 'The Wisdom Of Crowds', James Surowiecki, June 2004, ISBN 0-385-50386-5.
- [35] Dra. Avelina Koenes, Dr Claudio L. Soriano, 'Competitive advantage'. Edición Diaz de Santoz (1997). ISBN: 84-7978-283-8.
- [36] Shadan Maik. 'Enterprise Dashboards Design and best practices for It'. Published by John Wiley & Sons, Inc 2005. ISBN-13 978-0-471-73806-0.
- [37] Collins, Heidi. 'Corporate Portals Revolutionizing Information Access to Increase Productivity and Drive the Bottom Line'. Amacom 2001. ISBN: 0-8144-0593-2.
- [38] Diane Cerra, 'Effective Prototyping for software makers'. Published by Elsevier 2007. ISBN 13:978-0-12-088568-8.
- [39] Richard L. Sezov, Jr. 'Portal Administrar's Guide'. Liferay Press May 2, 2010. ISBN 978-0-615-24733-5.
- [40] Kevin Roebuck, 'Single Sign-On (Sso): High-Impact Strategies - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors'. Publisher: Tebbo (Jun 10 2011). ISBN 174304495X, 9781743044957.
- [41] John Arana, 'Creating flash widgets with Flash CS4 and ActionScript 3.0'. Published by Apress, 02/12/2008. ISBN 1430215844.
- [42] John M. Carroll, 'HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science'. Morgan Kaufmann 2003. ISBN: 9780080491417