

Sistema multiagentes para búsqueda de recursos de Datos Abiertos enlazados indexados



**UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS**

Camilo Valencia

Universidad Distrital Francisco José de
Caldas

Noviembre 2017

Necesidad

Se necesita un software que facilite la búsqueda e identificación de recursos abiertos bajo principios de niveles de conanza sobre Datos Abiertos Enlazados. El sistema debe contemplar desde la indexación y clasicación, hasta la búsqueda y consulta de recursos por parte del usuario.

Alcance

Por la complejidad del proyecto y teniendo en cuenta que está enmarcado dentro de un proyecto de investigación más grande, el trabajo presentado para la asignatura Ingeniería de software I desarrolla una implementación parcial del sistema propuesto que evidencia el uso de los conocimientos adquiridos durante el curso. Esta implementación parcial contiene la publicación de dos servicios con arquitectura Restful utilizando los métodos GET y POST del protocolo Http que permiten realizar búsquedas de recursos de datos abiertos enlazados en una base de datos de índices. Los servicios web están conectados al sistema multi agentes a través de una interfaz provista por la plataforma JADE. Los agentes se encargan de recoger la información de la búsqueda consultar la base de datos y devolver los resultados al usuario.

Requerimientos Funcionales

- El sistema debe permitir realizar búsquedas a un usuario por término de búsqueda
- El sistema debe rastrear automática y periódicamente la nube de datos enlazados para indexar los recursos disponibles
- El sistema debe clasificar los recursos según niveles de conanza teniendo en cuenta métricas de calidad de conanza
- El sistema debe registrar los datos de procedencia de los recursos rastreados y permitir denir niveles de conanza para las entidades de procedencia.
- El sistema debe permitir a los usuarios calicar los recursos encontrados para evaluar las métricas subjetivas de las dimensiones de conanza.
- El sistema debe permitir consultar el estado de rastreo e indexación de recursos.
- El sistema debe permitir añadir fuentes de rastreo de recursos



Requerimientos no funcionales

- El sistema debe responder las consultas de los usuarios en menos de 5 segundos.
- El sistema debe presentar los resultados de búsqueda de forma amigable, el usuario debe llegar al recurso solicitado en máximo de 2 clics.
- El sistema de consulta debe estar disponible a través de Servicios Web con arquitectura Restful.

Productos resultantes

- Aplicación publicada en AWS (Amazon Web Services):
<http://ec2-34-215-6-165.us-west-2.compute.amazonaws.com:8080/BuscadorAgentes>
- Documentación Java <http://ec2-34-215-6-165.us-west-2.compute.amazonaws.com:8080/BuscadorAgentes/doc>
- Repositorio Git con el contenido completo del proyecto:
<https://github.com/CamiloValencia/BuscadorAgentes.git>



Contenido del repositorio GIT

- Proyecto de Eclipse para verificar el código
- Archivo de respaldo de la base de datos para ejecutar el servicio localmente
- Librerías compiladas para poder ejecutar los agentes en la plataforma Jade
- Documento, presentación y anexos con la descripción del proceso desde el análisis hasta la implementación local y despliegue en la nube.
- Proyecto de Enterprise Architect con los modelos realizados

Herramientas utilizadas

- Ambiente de desarrollo Java EE con Eclipse y servidor de aplicaciones JBOSS Wildfly para ejecutar la aplicación web
- Plataforma JADE (Java Agent DEvelopment Framework) Para el despliegue de los agentes
- Enterprise Architect para la elaboración de los modelos UML.
- <http://draw.io> Para la elaboración de modelos inherentes a la metodología y sus componentes no se encuentran en Enterprise Architect
- Sharelatex para elaboración de documento y presentación
- Git Hub para el manejo de versiones
- Ms. Sql server como manejador de base de datos
- Amazon aws para soportar el servidor de aplicación y el de base de datos
- Git bash como consola para conexión ssh
- Citavi para la gestion de bibliografía.

Aporte del proyecto

El proyecto realiza un aporte significativo al curso ya que presenta el desarrollo paso a paso de una arquitectura poco común como lo es el sistema multiagentes que tiene muchos años, integrada con otra que es muy utilizada REST. Estos procesos no cuentan con mucha documentación y sirven como punto de partida para el desarrollo de nuevas soluciones. Además de esto el proyecto se ejecuta completamente en la nube estando acorde con las tendencias actuales de las organizaciones.

Análisis - Modelo de casos de uso

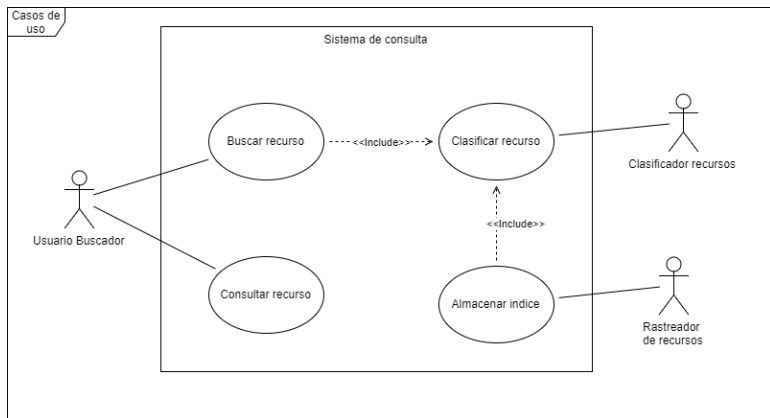


Figure: Casos de uso ed la aplicación

Análisis - Identificación Agentes

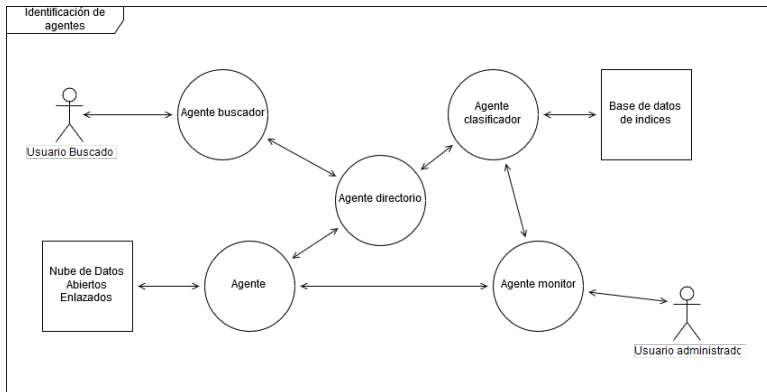


Figure: Identificación de Agentes y soluciones

Identificación Responsabilidades

Tipo de Agente	Responsabilidades
Agente rastreador	Rastrear la nube de datos abiertos enlazados a partir de un listado de url semilla y obtener recursos Informar la obtención de un recurso al agente clasificador pertinente
Agente clasificador	Clasificar recursos indexados Recuperar recursos solicitados del repositorio central Presentar recursos a agentes buscadores Actualizar información de consulta del recurso Modificar clasificación según uso del recurso Almacenar índices de recursos en un repositorio central
Agente buscador	Obtiene la consulta del usuario Devuelve resultados de búsqueda Devuelve recurso al usuario Solicitar recurso a un agente clasificador Informar al agente clasificador acerca de la efectividad de la búsqueda
Agente directorio	Registrar los servicios de agentes conectados al sistema
Agente monitor	Monitorear el estado de indexación, clasificación y aparición de errores

Table: Responsabilidades agentes

Análisis - Despliegue de los agentes

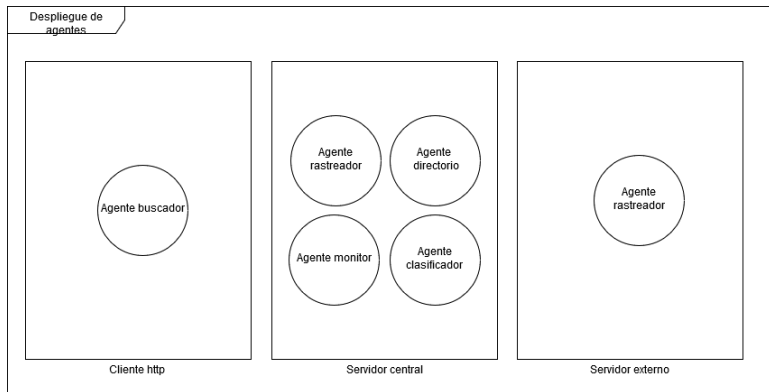


Figure: Despliegue agentes en dispositivos

Diseño - Interacción

Interacción	Resp	Protocolo Int	Role	Con	Cuando
Buscar termino	1	FIPA RE- QUEST	I	Agente bus- cador	El usuario envía un término de búsqueda
Obtener resulta- dos búsqueda	2	FIPA IN- FORM	R	Agente clasi- ficador	El agente recibe un término para búsqueda
Devolver recursos a agente buscador	3	FIPA IN- FORM	I	Agente clasi- ficador	El agente clasificador recupera la información

Table: Tabla de interacción

Diseño - Mensajes

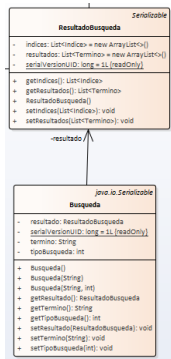


Figure: Mensajes

Diseño - Interacción Agentes recursos

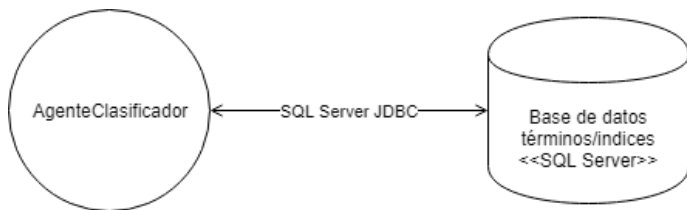


Figure: AgenteClasificador - Base de datos

Diseño - Interacción Agentes recursos

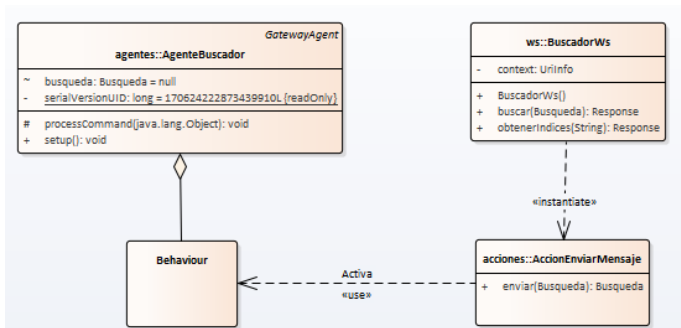


Figure: AgenteBuscador - Webservice

Diseño - Comportamientos internos agentes

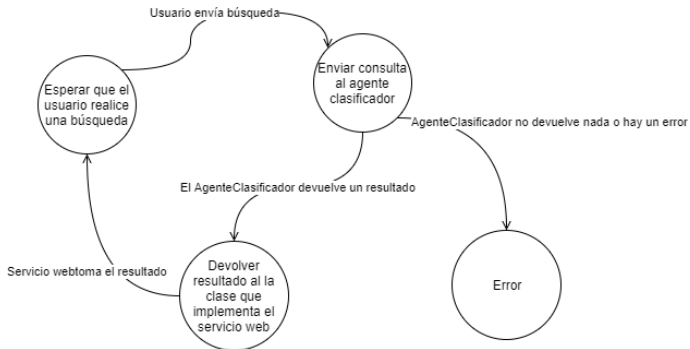


Figure: Comportamiento AgenteBuscador

Diseño - Comportamientos internos agentes

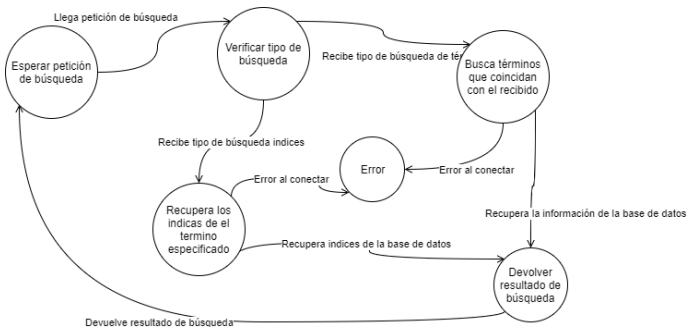
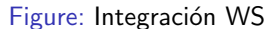


Figure: Comportamiento Agente Clasificador



Diseño - Modelo de base de datos

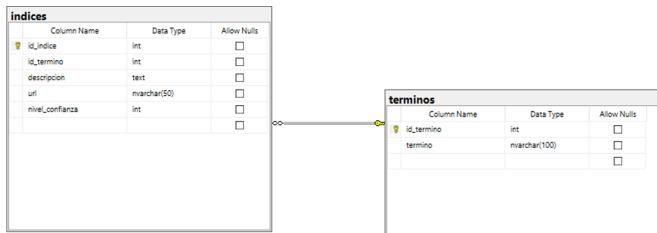


Figure: Base de datos

Conclusiones

- En la actualidad se presentan nuevas tecnologías todos los días pero estas pueden trabajar juntas para la creación de soluciones que una sola no podría ofrecer
- La agilidad y flexibilidad de la nube para soportar aplicaciones la hace atractiva a las organizaciones al reducir el time to market , los costos generados al adquirir una infraestructura y los costos de mantenimiento de un equipo de IT.
- No todas las soluciones del mercado en el tema de software son completas pero se pueden mezclar para obtener las funcionalidades que necesitamos
- Los sistemas multiagentes tienen una arquitectura difícil de digerir y diferente a las tradicionales, es necesario profundizar mas para sacar el mejor provecho de estos sistemas