Universidad Carlos iii de Madrid

Escuela Politécnica Superior

Ingeniería de Telecomunicación

PROYECTO FIN DE CARRERA

Extractor Web de Huella Digital

Autor: Ana María Salas Fernández
Tutor: Dr. Vicente Luque Centeno

Octubre de 2010

Agradecimientos

En primer lugar, me gustaría agradecer a mis padres todo su apoyo y su compresión, por compartir conmigo mis éxitos y mis fracasos, y estar incondicionalmente a mi lado. Sin ellos, no hubiera podido llegar donde estoy hoy. Gracias a mi hermano Nacho, por ser un gran hermano y por animarme siempre. Gracias a la gran familia que tengo.

También me gustaría dar las gracias a William, mi novio, porque gracias a él recuperé la seguridad y confianza en mí misma, y por compartir juntos los buenos y malos momentos que nos ha dado la carrera. Gracias por animarme, apoyarme y creer en mí siempre, por ser tan importante para mí en todos los momentos de mi carrera.

En este punto me tengo que acordar de mis compañeras y compañeros de estudio. Con los que he compartido largas horas de estudio y largos días de biblioteca. Gracias por ser tan buenos compañeros y amigos.

Por último, agradecer a Denodo la posibilidad de realizar con ellos el PFC y a todos mis compañeros por la acogida en mi actual empresas.

En definitiva, gracias a todos aquellos que han hecho posible este proyecto.

¡Gracias a todos!

Índice general

Lista de Figuras

Capítulo 1

Introducción y Objetivos

1.1. Introducción

La **Web 2.0** [1][2]se puede definir como la transición de las aplicaciones tradicionales hacia aplicaciones que funcionan a través de la Web enfocadas al usuario final. Se trata de aplicaciones que generan colaboración y servicios que intentan reemplazar las aplicaciones de escritorio, y en muchos casos lo consiguen.

Internet cambia constantemente y junto con él las maneras de aprovechar los datos que ahí se encuentran. Obtener y clasificar la cantidad de información que Internet brinda es un reto a conseguir. Desde hace mucho tiempo la información se ha usado para obtener ventaja sobre algún rival o competidor. Los datos hoy en día siguen siendo igualmente importantes. En la actualidad se presenta un nuevo reto, que es aprovechar el gran potencial que tiene Internet y la información que posee.

Ejemplos de la **Web 2.0** son las comunidades Web, los servicios Web, las aplicaciones Web, los servicios de red social, los servicios de alojamiento de videos, las wikis, blogs, mashups, etc... Dentro de la Web 2.0 se facilita al usuario interactuar con otros usuarios o cambiar contenido del sitio Web (Wiki), en contraste con antiguos sitios Web donde los usuarios se limitan a la visualización pasiva de información que se les proporciona.

Figura 1.1: Web 2.0

Internet deja de ser un lugar pasivo para convertirse en un espacio social dinámico.

- Las aplicaciones se integran ofreciendo al usuario múltiples posibilidades, además de darle voz dentro del mundo de Internet, ya que puede expresar su opinión, obtener opiniones de terceros, mostrarse a sí mismo...
- Las aplicaciones no son siempre de proveedor principal.
- Los usuarios pueden ser betatesters, desarrolladores...

Dentro de la **Web 2.0** han aparecido muchos tipos de nuevas aplicaciones o servicios, pero uno de los más conocidos y revolucionarios han sido las conocidas **Redes sociales**[4]. Éstas han ganado en Internet su lugar de una manera vertiginosa. Las redes sociales, actualmente, se han convertido en lugar de encuentro entre personas.

Una **red social** [4] [5] es una estructura compuesta de personas (u organizaciones u otras entidades), las cuales están conectadas por uno o varios tipos de relaciones. Estas relaciones pueden ser amistad, parentesco, intereses comunes, intercambios económicos, relaciones sexuales, compartir creencias, conocimiento o prestigio.

Existen muchos tipos de redes sociales. Existen redes dirigidas a todo tipo de usuario y sin una temática definida, que permiten la entrada y participación libre y genérica sin un fin definido. Los ejemplos más representativos del sector son **Facebook**[9], **Sonico**[10], **Twitter**[11], ...

También existen redes sociales que se desarrollan alrededor de una temática definida. Su objetivo es el de congregar en torno a esta temática a un colectivo concreto. Por ejemplo, **Linkedin**[12] están dirigida a relaciones profesionales para usuarios.

Hoy por hoy, cualquier persona tiene libre acceso a las redes sociales, dependiendo de los sistemas de registro que posean. Sin embargo, los jóvenes son los que utilizan las redes sociales de forma masiva. Por lo tanto, cualquier persona que cumpla los requisitos para el registro en una red social puede acceder a ella, independientemente de su edad, tanto para beneficiarse con sus utilidades, como para sufrir sus consecuencias.

Con la proliferación de las redes sociales tanto desde el punto de vista profesional como desde el punto de vista social, aparece el término "**Huella digital**".

En este proyecto, se definirá ese término como **Información que se encuentra en Internet de un individuo o empresa, ya pueda ser en una red social, en un blog, foro o cualquier página Web**.

En la definición anterior, no solo se introduce el término red social, sino que también se ha producido la expansión de blogs personales y la participación de personas en foros:

- Un blog[13], o en español también una bitácora, es un sitio Web que se actualiza periódicamente y recopila de forma cronológica textos o artículos de uno o varios autores. En los blogs el autor tiene el control sobre lo que se publica y comentarios que añaden terceras personas.
- Un foro[14], también conocido como foro de mensajes, foro de opinión o foro de discusión, es una aplicación Web que da soporte a discusiones u opiniones.

1.1.1. Problemática de las redes sociales

Las **redes sociales** son muy útiles, además que permiten conectar con amigos de la infancia, compañeros de facultad, profesionales de nuestro sector, gente que comparte nuestras mismas aficiones, etc., pero también existen una serie de problemas[7] [8] que pueden surgir a través de ellas. A continuación se dará un repaso a los problemas más comunes.

1. La **falta de privacidad y seguridad** en cuanto al tratamiento de los datos. Es importante informarse de donde se encuentra el alojamiento de las Redes sociales, es decir, si el almacenamiento de datos se produce en nuestro país o en países externos, ya que la regulación cambia totalmente. Lo más preocupante en las recogidas de datos es que se permitan cesiones a terceras empresas para las que se desconoce su finalidad.

2. Generación de **contenidos protegibles por propiedad intelectual o por el derecho a la propia imagen** por estas redes, ya que muchas de ellas, en el caso de realizar una inclusión de contenidos, adquieren directamente una licencia sobre los mismos.

3. **El usuario se convierte también en generador de comunicaciones no deseadas**. Se puede generar muy fácilmente spam cuando se bombardea con invitaciones a alguna persona o con aplicaciones, fiestas, eventos, recuerdos, etc. Siendo esta práctica mal vista.

4. **La infancia y los menores de edad**. Muchas de estas plataformas no establecen ningún mecanismo de control de la edad seguro. En un porcentaje muy alto, se pregunta con un simple clic en una casilla que se marca si se es o no menor de edad o se introduce la fecha de nacimiento mediante menús desplegables. Además, en muchas de estas plataformas no concuerda la edad de acceso al servicio con lo que en España se considera un joven adulto.

En este proyecto se quiere demostrar que es posible obtener cierta información, aun cuando ella es privada o el usuario cree que lo es. Además de los problemas relacionados con la privacidad, hay que partir de la base que la mayoría de las personas que utilizan las redes sociales no saben utilizarlas completamente, y además en muchos casos, publican demasiada información sobre ellos.

A continuación, se verán los objetivos que se quieren obtener en el presente proyecto fin de carrera.

1.2. Objetivos

En el apartado anterior, se ha presentado el escenario en el que se va a trabajar y la problemática que se tiene hoy día con las redes sociales, y con la información privada contenida en la Web en general. Es necesario entender este problema y el entorno de trabajo ya que **Skiptracing** hace un uso amplio de la Web 2.0. El **objetivo** de este proyecto es triple:

- Por una lado, se creará una aplicación que será capaz de extraer información de Internet, información que en muchos casos es privada o el usuario cree serlo. El programa, denominado **Skiptracing** ha sido realizado en la empresa **Denodo Technologies**[15], utilizando la plataforma que ellos proporcionan, **Denodo Platform 4.6**[16].

  Aunque **Skiptracing** obtiene información personal de Internet, esto no quiere decir que la información que **Skiptracing** extrae no sea información ”legal”. **Skiptracing** extrae la información que Internet le

proporciona ”públicamente”, es decir, que cualquier otra persona realizando una búsqueda a mano podrá ser capaz de obtener.

- El segundo objetivo, es hacer una demostración del funcionamiento de la plataforma **Denodo** y ver como su uso facilita el desarrollo de la aplicación actual. El actual proyecto fin de carrera, fue realizado en **Denodo Technologies** como prácticas en empresa, y se desarrolló como piloto para un cliente final.

- Demostrar que en el momento en el que se entra en Internet no es difícil poder obtener información sobre cualquier persona, entendiendo que la persona en cuestión realiza un uso de Internet en general, de la Web 2.0 en particular. La extracción de información por parte de redes sociales es el caso más preocupante, en muchos casos dan cierta información que podría ser privada y en caso de que no sea así, queda patente la mala utilización de las redes sociales por parte del usuario.

  Las redes sociales ofrecen opciones e información para proteger nuestros datos, que en muchas, o la mayoría, el usuario ignora dejando sus datos completamente expuestos al mundo.

  Además, existen muchas personas que utilizan un blog personal o realizan consultas personales en foros completamente abiertos, pudiendo extraerse información muy personal por parte de terceras personas.

1.3. Partes del proyecto

El actual proyecto fin de carrera presenta las siguientes partes:

1. **Estado del arte:** En este punto se exponen las tecnologías utilizadas para el desarrollo de la aplicación **Skiptracing**.

2. **Plataforma Denodo:** Se realiza una pequeña explicación de lo que ofrece la plataforma Denodo, y el por qué de su utilización en la aplicación **Skiptracing**. Así como una comparativa con otras plataformas similares existentes.

3. **Skiptracing:** Este será el capítulo principal del proyecto, ya que se explicarán los módulos de los que está compuesto el proyecto y se explicará su desarrollo.

4. **Historia del proyecto:** En este capítulo se explicarán las circunstancias de los autores y el por qué de la realización del actual proyecto fin de carrera.

5. **Conclusiones:** En este capítulo se tienen las conclusiones en cuanto a los objetivos marcados, el cumplimiento de los mismos y las conclusiones sacadas al final de la realización del proyecto.

6. **Trabajos futuros:** Posibles ampliaciones al actual proyecto.

Capítulo 2

Estado del Arte

2.1. Data Mashup

2.1.1. Introducción

El término **mashup**[17][18][19] fue utilizado por primera vez en los medios de comunicación durante la última década, ya que se comenzó a combinar fragmentos de canciones, videos o gráficos de diferentes fuentes para crear nuevos e interesantes contenidos. Más recientemente, el concepto y el término se ha ampliado a aplicaciones de contenido Web, apareciendo, por ejemplo, los conocidos canales RSS[21] que permiten al usuario la combinación de información. También es posible encontrar otra serie de portales que contienen información de distintos sitios Web, obteniendo un producto personalizado de consumo de información. Para el usuario final el mashup es invisible, ellos simplemente disfrutan de los beneficios de un mayor acceso a los datos y mayor capacidad para aumentar el conocimiento.

Los mashups se presentan actualmente en tres formas: **mashups de consumidores**, **mashups de datos** y **mashups empresariales**.

- **Mashup Consumidores:** Un ejemplo de este mashup es la utilización por parte de muchas aplicaciones de Google Maps. Los mashups de este tipo combinan datos de varias fuentes, mostrando la información a través de una interfaz gráfica simple.
- **Mashup de datos:** Este tipo consiste en la mezcla datos proveniente de diferentes fuentes. Por ejemplo, feeds RSS en un solo feed con nuevo un front-end gráfico.
- **Mashup empresarial** (ejemplo, JackBe, http://www.jackbe.com) integra usualmente datos de fuentes externas e internas. Por ejemplo,

podría crear una Web con información sobre la cuota de mercado de un negocio combinando la lista externa de todas las casas vendidas la semana anterior con datos internos de las casas vendidas por una sola agencia.

Mashups dentro de mashups son conocidos como "**mashups monstruos**".

2.1.2. Ejemplos [20]

1. **Chicago Crime**: El departamento de policía de Chicago tiene un mashup (http://gis.chicagopolice.org) que integra la base de datos del departamento de crímenes reportados con Google Maps de modo de ayudar a detener crímenes en ciertas áreas y avisar a los ciudadanos de áreas potencialmente más peligrosas.

2. **WikiCrimes**: WikiCrimes (http://www.wikicrimes.org) es un sitio Web tipo wiki donde los usuarios de Internet pueden reportar crímenes pinchando banderas en un mapa basado en Google Maps.

3. **Minnus** (http://www.minnus.com.ar): minnus es una comunidad virtual donde sus usuarios tienen acceso a información de todo tipo, ya sea cultural, social, histórica, comercial, ambiental, turística, de transito, etc. Dicha información se encuentra geoposicionada en un mapa creado con imágenes de satélite para así formar grupos con mismos intereses.

4. **Flickr**: Flickr es un sitio de almacenamiento de imágenes que permite a los usuarios organizar sus colecciones de imágenes y compartirlas. Utilizando su API el contenido puede ser usado en otros sitios creando mashups. Flickrvision (http://flickrvision.com) es un ejemplo.

5. **Travature**: Travature (http://www.travature.com) es un portal de viajes que ha integrado motores de meta búsquedas con guías de viajes tipo wiki y reseñas de hoteles. También permite compartir experiencias entre viajantes.

6. **Digg y Reddit**: Digg (http://digg.com/) y Reddit (httP://reddit.com) son mashup de varios sitios de noticias controlado casi enteramente por los usuarios del sitio. Al igual que Digg existe la versión española Menéame.

7. **BFreeNews.com**: BFreeNews (http://bfreenews.com/) es un mashup de fuentes de noticias de calidad cruzadas con recomendaciones de noticias de Twitter y búsquedas de Google. Muestra las noticias más comentadas en twitter y más indexadas por Google en las últimas 24 horas.

8. **Histourist**: Histourist (http://www.histourist.com/) es un Mashup Semántico que ofrece una Enciclopedia multimedia geolocalizada de lugares históricos. Los artículos se preparan mediante una combinación de editores y robots de software que explotan los recursos online en fuentes definidas como *confiables* (BBC, National Geographic, DBpedia, The History Channel, etc.) y en particular los servicios de la Web semántica para enriquecer los artículos con videos, fotos, bibliografía, y clasificarlo en las taxonomías del servicio.

2.2. Wicket

2.2.1. Introducción

Wicket[22] es un Framework para el desarrollo de aplicaciones Web en Java[23] y se basa únicamente en Java y HTML. Básicamente está orientado a componentes y el manejo de eventos dentro de una aplicación Web. Es una forma de desarrollo web muy similar a crear una aplicación de escritorio, es decir, como si se usara AWT[24] o SWING[25]. Apache Wicket es un Frameworks de desarrollo Web muy prometedor ya que separa claramente vista y lógica. Todo el comportamiento de la página estará programado en Java.

2.2.2. ¿Por qué Wicket?

Si se busca un lenguaje para desarrollar Web en java, existen muchas posibles elecciones hoy en día, existen muchos Framework Web de java. Cuantos Frameworks de java existen, la respuesta se presenta a continuación:

Echo	Cocoon	Millstone
Struts	SOFIA	Tapestry
RIFE	Spring MVC	Canyamo
JPublish	JATO	Folium
Verge	Niggle	Bishop
Action	Framework Shocks	TeaServlet
Expresso	Bento	jStatemachine
OpenEmcee	Turbine	Scope
JWAA	Jaffa	Jacquard
Smile	MyFaces	Chiba
Jeenius	JWarp	Genie
Dovetail	Cameleon	JFormular
Japple	Helma	Dinamica
Nacho	Cassandra	Baritus
Click	GWT	OXF
WebWork	Maverick	Jucas
Barracuda	wingS	jZonic
Warfare	Macaw	JBanana
Melati	Xoplon	WebOnSwing
Stripes		

Viendo todas estas soluciones, ¿De qué sirve otra solución java? Wicket surge para mejorar los Framework de java que ya existen, y ofrecer mucho más que de lo que pueden ofrecer otros Framework. Los Frameworks más cercanos a Wicket son probablemente **Tapestry** y **Echo**, pero incluso ahí la semejanza es muy poco profunda:

- Como **Tapestry**[30], Wicket utiliza un atributo HTML especial para designar los componentes, lo que permite una fácil edición ordinaria con editores de HTML.
- Al igual que **Echo**[31], Wicket tiene un modelo de componentes de primera clase.

Aun así, las aplicaciones Wicket no son como aplicaciones escritas en Tapestry o Echo, Wicket incorpora lo mejor de ambos mundos. Se obtienen los beneficios de un modelo de componentes de primera clase y un enfoque no-intrusivo en HTML. En muchas situaciones, esta combinación puede resultar ser una importante ventaja en el desarrollo.

2.2.3. Motivación

Para entender por qué Wicket es diferente, se intentarán explicar las motivaciones que llevaron a su desarrollo.

1. **La mayoría de los Framework prevén débilmente o inexistente un soporte para la gestión en el lado del servidor**

   Esto normalmente significa gran cantidad de código ad-hoc en las aplicaciones Web repartiendo la administración del servidor. Mientras Wicket no permitirá dejar de pensar en el estado del servidor.

   En Wicket, todo el estado del servidor es gestionado de forma automática. Nunca se utilizará directamente el objeto HttpSession u objetos similares de estado. Cada componente página tiene una jerarquía de componentes con estado, donde el modelo de cada componente es, en definitiva, un POJO[32] (Plain Old Java Object). Wicket mantiene un mapa de estas páginas en la sesión de cada usuario.

   El programador trata con objetos simples, familiares Java y Wicket lidia con cosas como URL, ID de sesión y peticiones GET / POST, facilitando su uso.

   Wicket también proporciona una solución al problema de ”Botón de vuelta”, tiene una solución genérica y robusta que permite identificar y hacer expirar las páginas que se han vuelto obsoletas debido a los cambios estructurales del modelo de un componente en la página.

   Por último, Wicket ha sido diseñado para trabajar con Frameworks de persistencia POJO como JDO[33] o Hibernate[34]. Esto puede hacer que las aplicaciones Web con bases de datos sean muy fáciles de implementar.

   En términos de eficiencia frente a la productividad, tal vez es Wicket para JSP[27] como Java es para C[39]. Se puede lograr cualquier cosa usando Wicket, incluso hacer que sea más eficiente en términos de memoria o consumo del procesador. Sin embargo, el tiempo de desarrollo es más largo.

2. **Existen más Frameworks que requieren código HTML especial**

   JSP es el que peor oferta ofrece en cuanto a inserción de código especial en HTML, pero hasta cierto punto casi todos los Frameworks anteriormente mencionados (a excepción de Tapestry) introducen algún tipo de sintaxis especial al código HTML. La sintaxis añadida es poco recomendable, ya que cambia la naturaleza del HTML, lo que hace más complicado el mantenimiento.

   Wicket no añade ninguna sintaxis a HTML. En su lugar, se extiende HTML de una manera compatible con los estándares a través de un namespace que incorpora Wicket, totalmente compatible con el estándar XHTML. Esto significa que se puede usar Macromedia Dreamweaver, Microsoft Front Page, Word, Adobe Live Id, o cualquier otro editor HTML existente para trabajar en páginas Web y componentes Wicket.

Para lograr esto, Wicket utiliza un atributo, un id único, **wicket:id**, para marcar las etiquetas HTML que deben recibir un trato especial por parte del kit de herramientas de Wicket.

No existe nada adicional en el código HTML, lo que significa que los diseñadores pueden maquetar páginas que se pueden utilizar directamente en el desarrollo. Añadir componentes Java en el código HTML es tan sencillo como establecer el atributo nombre del componente. Del mismo modo, los programadores pueden trabajar en los componentes de Java que se conectan al HTML sin preocuparse por el diseño que ellos puedan añadir a la Web, se pueden centrar en la funcionalidad.

3. **Existen frameworks complejos**

   La mayoría de las herramientas existentes tienen pobres definiciones o inexistentes del modelo de datos. En algunos casos, el modelo se define utilizando sintaxis XML especial. La sintaxis puede ser tan complicada que se requieren herramientas especiales para manipular toda la información de configuración. Dado que estas herramientas no son simples bibliotecas Java pueden no ser utilizas en las herramientas de desarrollos que normalmente se utilizan.

   Wicket es todo simplicidad. No hay archivos de configuración para aprender en Wicket. Wicket incorpora una biblioteca de clases simples con un enfoque coherente a la estructura de sus componentes. En Wicket, las aplicaciones Web se parecerán más a una aplicación Swing que una aplicación JSP. Si se conoce Java (y especialmente Swing), ya se conoce mucho acerca de Wicket.

   Tapestry y **JSF** tienen modelos de componentes que permiten la reutilización, pero no es particularmente trivial su uso, al menos en comparación con Wicket. Wicket ha sido expresamente diseñado para que sea muy fácil la creación de componentes reutilizables. Es muy simple extender los componentes existentes y fabricar componentes nuevos.

2.3. Spring

2.3.1. Introducción

Spring Framework[36][37] [38] (también conocido como **Spring**) es un Framework de código abierto de desarrollo de aplicaciones para la plataforma Java.

A pesar de que Spring Framework no obliga a usar un modelo de programación en particular, se ha popularizado usarlo junto a Java al considerarse una alternativa y sustituto del modelo de Enterprise JavaBean[49].

2.3.2. ¿Qué es Spring?

La motivación inicial era facilitar el desarrollo de aplicaciones J2EE, promoviendo buenas prácticas de diseño y programación. Se enfoca al manejo de objetos de negocio, dentro de una arquitectura en capas. Además una de sus mayores ventajas es su modularidad, pudiendo utilizar algunos de sus módulos sin comprometerse con el uso del resto.

Uno de los principales objetivos de Spring es no ser intrusivo, es decir, las aplicaciones están configuradas para utilizar *Beans* mediante Spring, no necesitan depender de interfaces o clases de Spring, obteniendo su configuración a través de las propiedades de sus *Beans*. Este concepto puede ser aplicado a cualquier entorno, desde una aplicación J2EE hasta un applet.

Una característica de Spring es que puede conseguir la integración entre diferentes APIs (JDBC[40], JNDI[50], etc.) y Frameworks (por ejemplo entre Structs[28] e iBatis[51]).

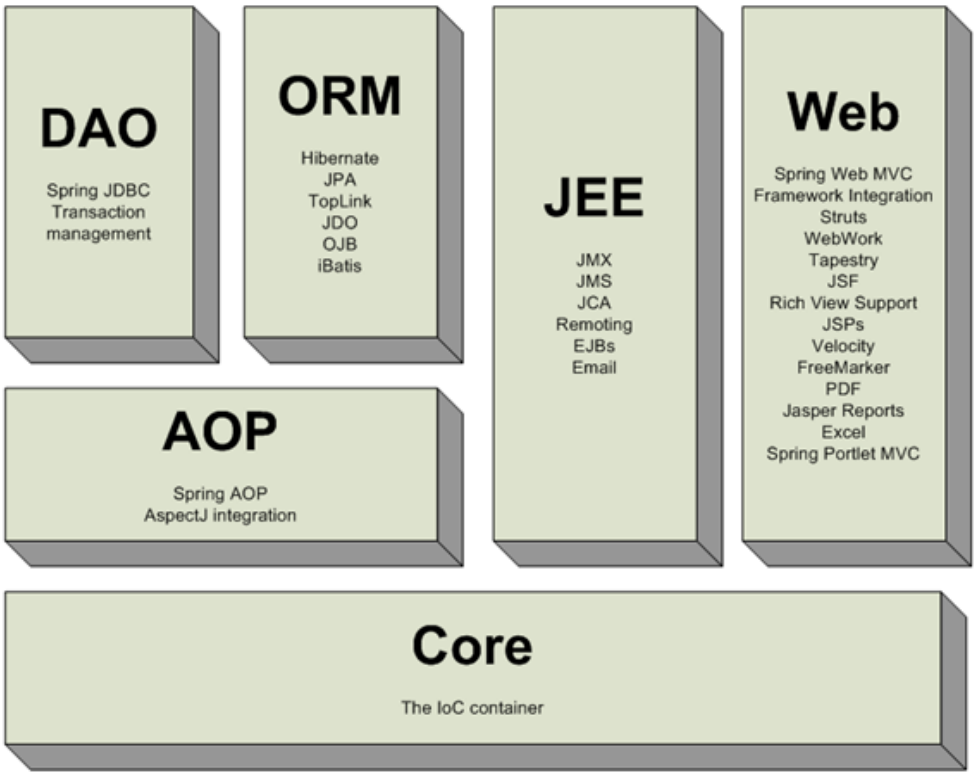


Figura 2.1: Módulos de Spring

Spring está compuesto por un conjunto características, las cuales están agrupadas en seis módulos principales:

- **El módulo Core o "Núcleo"** es la parte fundamental del Framework ya que provee toda la funcionalidad de Inyección de Dependencias permitiéndo administrar la funcionalidad del contenedor de beans. Esto significa que la creación de los objetos lo lleva a cabo un contenedor externo inyectándolos a otros objetos que dependan de los primeros.

- Encima del módulo core se encuentra el **módulo Context (Contexto)**, el cual provee de herramientas para acceder a los beans de una manera elegante, similar a un registro JNDI[50].

- **El paquete DAO** provee una capa de abstracción de JDBC que elimina la necesidad de teclear código JDBC. También provee un mecanismo de administración de transacciones tanto declarativas como programáticas, no solo para clases que implementen interfaces especiales, sino para todos los POJOs.

- **El paquete ORM** provee capas de integración para APIs de mapeo objeto relacional, incluyendo JDO[33], Hibernate[34] e iBatis[51].

- **El paquete AOP** provee una implementación de programación orientada a aspectos compatible con AOP Alliance[54]. Con ello se quiere desacoplar el código de una manera limpia implementando funcionalidad que por lógica y claridad debería estar separada.

- **El paquete Web** provee características básicas de integración orientadas a la Web. El paquete Web *MVC* provee de una implementación *Modelo-Vista-Controlador* para las aplicaciones Web. La implementación de Spring MVC permite una separación entre código y la interfaz Web, permitiendo además el uso de otras características de Spring Framework como lo es la validación.

Capítulo 3

Plataforma Denodo

3.1. Introducción

La **Plataforma Denodo** es una solución de **Denodo Technologies** que realiza la integración de fuentes de distinta naturaleza, es decir, información heterogénea y distribuida, estructurada y no estructurada. Además, la plataforma permite automatizar la navegación Web para extraer información. No solo permite la extracción Web, sino que además es posible realizar la extracción de información de documentos y bases de datos, entre otras muchas fuentes.

Si se observa la expansión de Internet en la actualidad, se llega a la conclusión de que existe una gran cantidad de información útil en la Web y de difícil extracción. Esta información está ofrecida para los usuarios de Internet en interfaces amigables para ellos.

Denodo propone una solución global de integración en tiempo real de fuentes de información heterogénea y dispersa, estructurada y no estructurada.

Para ello combina diversos módulos integrados entre sí:

- El módulo **Virtual DataPort (VDP)**[55][56]: Genera una **Base de datos virtual** de fácil acceso, a través de la unificación de datos procedentes de diferentes fuentes.
- El módulo **ITPilot (ITP)** [57] automatiza la navegación Web extrayendo información de dicho interfaz Web. ITPilot nos proporciona una herramienta para extraer información que es amigable para el usuario pero no para la extracción automática de datos. La extracción de información la realiza con un envoltorio o *Wrapper*. ITPilot proporciona un entorno distribuido y escalable de generación, ejecución y mantenimiento de *Wrappers*.

- El módulo **Aracne**[58] permite el **rastreo o crawling**, indexación, filtrado y consulta de información no estructurada que puede estar almacenada en diferentes sitios y en diferentes formatos. Aracne es capaz de obtener información de la Web, sistemas de ficheros, bases de datos relacionales o servidores de correo electrónico.

- El módulo **Scheduler**[59]. Este módulo es el módulo de planificador temporal. Es capaz de planificar tareas batch que utilizan los módulos anteriormente mencionados.

3.2. Virtual Data Port

3.2.1. Introducción

Denodo Virtual Data Port[55] permite el acceso y la integración fácilmente de diferentes fuentes de información. En la actualidad, las entidades o empresas poseen la información en diferentes formatos (bases de datos relacionales, servicios Web, documentos XML, hojas de cálculo, ficheros planos...) y en diferentes lugares, haciendo que su extracción sea compleja y diferente en cada caso.

Virtual DataPort es una solución global para la integración en tiempo real de fuentes de información heterogénea y dispersas, estructuradas y no estructuradas.

Virtual DataPort integra todo tipo de información sin importar su procedencia ni el formato. Además proporciona una herramienta de fácil uso que permite combinar información de fuentes dispares y poder acceder a ella de manera sencilla a través de la construcción de una base de datos virtual.

3.2.2. Arquitectura VDP

Como ya se ha comentado, **Virtual Data Port** permite a las aplicaciones tratar diferentes fuentes distribuidas y de diversos formatos, incluyendo fuentes externas. El acceso a todas estas fuentes se realiza como si estuviese contenida en una ”Base de datos virtual”. Por ello, **VDP** proporciona un acceso sencillo a la información.

Virtual Data Port proporciona una visión estructurada y unificada de la información. VDP es capaz de tratar fuentes de muy diversa naturaleza, entre otros: Bases de Datos, sitios Web, documentos XML, Servicios Web, ficheros de texto plano, hojas de cálculo, índices sobre información no estructurada, etc.

Cada fuente, dentro de **VDP**, se puede visualizar a través de ”vistas”, pudiéndose acceder a ellas a través de un lenguaje (compatible en gran medida con SQL) llamado **VQL**[69]. Además permite combinar las vistas creando nuevas vistas con la información que más interese en cada caso. Estas vistas, también pueden estar compuestas por fuentes de información no estructurada o semi-estructurada (como pueden ser formularios HTML de los que se extrae la información a través de **ITPilot**[57]), realizando la consulta de la información en tiempo real.

Cuando se ejecuta una consulta VQL, se desencadenan subconsultas que se enviarán en tiempo real a todas las fuentes involucradas. Además VQL posee un módulo de caché que permite decidir el mecanismo de actualización de los datos de las fuentes, es decir, el sistema podrá acceder a los datos de las fuentes en tiempo real o por el contrario, pueden crearse y configurarse caches para las fuentes o vistas que se desee.

Virtual DataPort también permite la actualización de las fuentes de datos, siempre que éstas sean capaces de soportar transacciones.

El sistema de integración de información proporcionado por Virtual Data Port se modela en 3 niveles independientes: la capa de usuario, la capa lógica y la capa física (Wrappers). En la figura 3.1 se muestra una visión general de la arquitectura del sistema. A continuación se describen cada uno de estos niveles.

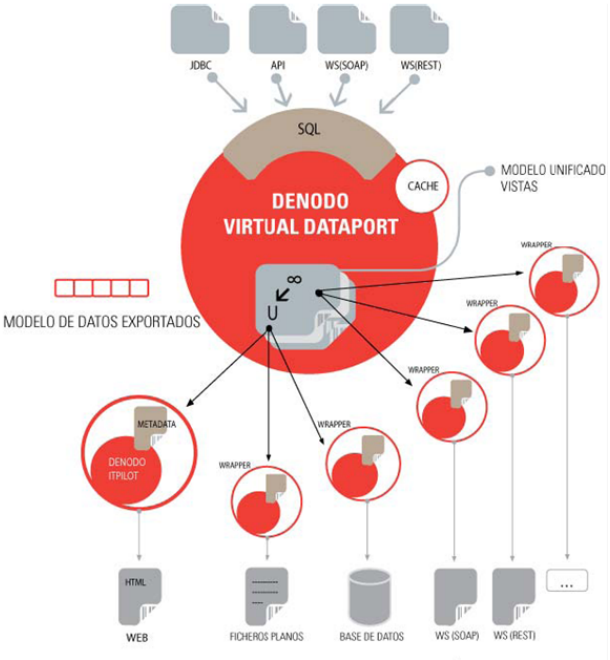


Figura 3.1: Arquitectura VDP

3.2.2.1. NIVEL FÍSICO

El nivel físico tiene la misión de conseguir que las fuentes de datos originales devuelvan los datos de acuerdo con unas estructuras definidas teniendo en cuenta el modelo de datos común que utiliza VDP. Esto se realiza a través de lo que se denomina "Wrapper o envoltorio". Un Wrapper extrae información de las fuentes, interpreta los resultados obtenidos y devuelve la información en el formato que emplea VDP. Si la fuente lo permite es capar de actualizar e insertar información, a través de dicho Wrapper

Como ya se ha comentado anteriormente, VDP es capaz de unir mucha información de fuentes completamente diferentes, los Wrappers serán los encargados de conseguir que la información sea legible para VDP, independientemente del tipo de fuente. Para ello se tienen diferentes Wrapper dependiendo de la fuente originaria. Los Wrappers existentes son:

- **Bases de datos**: Extraen datos desde una Base de Datos Remota vía JDBC u ODBC. También insertan, actualizan y/o borran datos en los casos en los que las fuentes lo permitan.
- **Servicios Web**: Extraen datos realizando llamadas a operaciones definidas por servicios Web.
- **XML**: Extraen datos encapsulados en ficheros XML. Los documentos XML pueden ser accedidos en el disco local o a través de protocolos como HTTP y FTP.
- **JSON**: Permiten extraer datos en formato JSON [46].
- **Ficheros planos**: Extraen datos de ficheros planos en formato CSV (Comma Separated Values) o similar. Los documentos CSV pueden ser accedidos en el disco local o a través de protocolos como HTTP y FTP.
- **ITPilot**: Permiten acceder a información contenida en fuentes Web semi-estructuradas, permitiendo tratarlas de forma similar a como si fuesen bases de datos estructuradas. Los Wrappers de este tipo son generados utilizando Denodo ITPilot [57].
- **Aracne**: Denodo Aracne [58] permite el crawling, indexación y búsqueda de información no estructurada contenida en repositorios tales como la Web, bases de datos relacionales, sistemas de ficheros locales o servidores de correo electrónico. Los índices creados con Denodo Aracne pueden ser importados directamente en VDP para su consulta y combinación conjunta con información estructurada y semi-estructurada.
- **Google Mini**: Google Mini [60] es la solución corporativa de Google para el crawling, indexación y búsqueda de información contenida en la

Web. Los índices creados con Google Mini pueden ser importados directamente en Denodo DataPort para su consulta y combinación conjunta con información estructurada y semi-estructurada.

- **LDAP**: Permiten extraer datos contenidos en un directorio LDAP.
- **CUSTOM**: Extraen la información de una fuente, a través de una implementación Java proporcionada por el administrador de Virtual DataPort. Este tipo de Wrapper permite la construcción ad-hoc de un Wrapper para una fuente específica. Los Wrappers CUSTOM también pueden permitir la inserción, actualización y/o borrado de datos.

El usuario que emplee VDP para unir todas sus fuentes, no necesitará crear un Wrapper diferente para cada caso. VDP es capaz de crear automáticamente el Wrapper para los siguientes tipos de fuente:

- Base de datos
- Ficheros planos
- XML
- Aracne y google mini.

Sin embargo, es necesario crear un Wrapper específico para fuentes Web semi-estructuradas. En el siguiente capítulo sobre ITPilot se explicará cómo se realiza dicho Wrapper.

3.2.2.2. NIVEL LÓGICO

El nivel lógico integra y combina las relaciones exportadas por los diferentes Wrappers, componiendo el esquema global del sistema. Esta combinación se realiza mediante la definición de vistas, expresadas en el lenguaje **Denodo VQL** [69]. Las relaciones creadas inicialmente por los diferentes Wrappers, que representan a las fuentes del sistema, se denominan relaciones base (o, también vistas base).

Una vez creadas las relaciones base representando las fuentes del sistema, el administrador puede crear vistas que deriven de vistas base ya creadas combinándolas como desee, creando de esta manera las vistas del esquema global (o también vistas derivadas). Es importante resaltar que este proceso puede realizarse de manera recursiva en varios pasos: una vista derivada puede ser utilizada como base para la creación de nuevas vistas.

Una vez creadas las vistas del esquema global combinando la información de las fuentes, el nivel lógico es capaz de obtener información sobre consultas VQL tanto sobre las vistas derivadas como sobre las relaciones base.

El lenguaje de consultas VQL se basa en SQL, incorporando diversas extensiones para manejar información heterogénea y distribuida.

Cuando el sistema recibe una consulta, comprueba si se puede resolver en función de las capacidades de consulta de las fuentes, elabora los posibles planes de ejecución, selecciona el más óptimo, y ejecuta la consulta devolviendo a la capa superior los resultados obtenidos.

La capa lógica de **Virtual DataPort** también permite la actualización de las fuentes de datos mediante operaciones INSERT/UPDATE/DELETE, siempre que éstas sean capaces de soportar transacciones. En la capa lógica se pueden diferenciar los siguientes módulos:

- **Generador de Planes de Consulta**: En primer lugar, el generador de planes decide si la consulta recibida puede o no ser contestada, de acuerdo a las capacidades de consulta soportadas por las fuentes. Si puede ser contestado, genera los posibles planes de ejecución para la consulta.

- **Optimizador**: Selecciona el plan de ejecución óptimo de entre todos los posibles (obtenidos por el Generador de Planes de Consulta). La selección del plan de ejecución se basa en la materialización de cada uno de los planes activados y en su complejidad. También se consideran las capacidades de consulta ofrecidas por las fuentes, de manera que se puedan delegar a las mismas las operaciones que puedan realizar localmente, obteniéndose así una ejecución más eficiente y un menor intercambio de datos a través de la red.

- **Motor de Ejecución de Consultas**: Una vez seleccionado el plan óptimo, el motor de ejecución se encarga de ponerlo en práctica, ejecutando las sub-consultas sobre las fuentes e integrando los resultados obtenidos para generar la respuesta global. Se tendrá en cuenta que la información pueda estar precargada en el módulo de caché.

3.2.2.3. MÓDULOS DE DATOS: CACHÉ

Como ya se ha comentado anteriormente, **VDP** dispone de un sistema para almacenar copias locales de los datos de las fuentes que se desee. Esta caché, se almacena en una Base de Datos Relacional accesible vía JDBC. VDP incluye una base de datos Apache Derby[61] embebida que puede funcionar como caché sin necesidad de software adicional. Es posible utilizar otro tipo de base de datos externas para este propósito.

El sistema administrará totalmente la base de datos caché por cada relación base o vista para la que se haya realizado la configuración de la caché.

También hay que destacar, que cada vista mantendrá una descripción de las tuplas actualmente contenidas en la cache, de forma que el gestor pueda saber que consultas sobre la relación son resolubles con datos disponibles localmente.

Al igual que cualquier otra caché, debe tener un tiempo de expiración de manera que las tuplas *caducadas* no se devuelvan al realizar una consulta. Por lo tanto, las tuplas caducadas se eliminarán de forma automática cada cierto tiempo, siendo este tiempo totalmente configurable. El uso o no de la caché es completamente configurable por parte del usuario y se puede realizar la configuración vista a vista.

Este mecanismo permite además realizar precargas periódicas de datos de manera muy sencilla, simplemente escribiendo una consulta describiendo los datos a precargar y planificando temporalmente la repetición de dicha consulta con el intervalo temporal deseado.

Como ya se ha comentado, la configuración de la caché se puede realizar vista a vista, por lo que puede inhabilitarse para una consulta concreta o para las relaciones o fuentes que se desee, de manera que siempre es posible acceder a la información de las fuentes en tiempo real.

3.3. ITPilot

3.3.1. Introducción

La información que se puede encontrar en Internet, se obtiene por medios amigables para el usuario, por lo tanto la automatización de la extracción de esta información no se puede realizar de una manera trivial. Además, clientes y proveedores suelen tener información en interfaces como los presentes en la Web. Por lo tanto, existe la problemática de acceso a estas fuentes de información, ya que los resultados que proporcionan son generalmente en HTML, que se trata de un lenguaje de etiquetas definido para la visualización. HTML no define una estructura y/o la semántica de los resultados generados, y no diferencia estructuralmente entre elementos de navegación, paneles gráficos e información útil para el usuario.

Denodo ITPilot[57] es la solución de **Denodo Technologies** que permite extraer y estructurar de manera sencilla el conjunto de datos que existe en la Web. Para ello, empleará y combinará una serie de componentes, cada uno con una tarea específica. Está basado en el desarrollo de *Wrappers o envoltorios* capaces de automatizar el proceso de navegación y la extracción de información de fuentes Web, en la utilización y configuración de componentes (cada uno con una tarea específica) y de la relación entre ellos. Además también es capaz de extraer información de ficheros Microsoft Word[64] y Adobe

PDF[65].

Denodo ITPilot proporciona una herramienta gráfica que ofrece al usuario una gran variedad de componentes para realizar la automatización de la navegación y la extracción de datos. Cada componente puede relacionarse con otros componentes, a través de relaciones de entrada y salida de información. A partir de esta descripción gráfica de componentes, que proporciona la secuencia de navegación a ITP, se genera un programa envoltorio en JavaScript[72]. Este programa consiste en la declaración de cada uno de los componentes y la relación que existe entre ellos. Entre estos componentes existen dos componentes importantes, uno de extracción de información y otro de navegación. Para cada uno de ellos se emplea un lenguaje específico de navegación y extracción de información denominado NSEQL[67] y DEXTL[68] respectivamente.

Además, ITP, permite probar y depurar el envoltorio generado, antes de desplegarse en el servidor de ejecución.

3.3.2. Distribución de los entornos

3.3.2.1. ENTORNO DE GENERACIÓN

Como se ha comentado en el apartado anterior, el **Entorno de Generación** permite crear Wrappers de una manera visual y sencilla. Este entorno requiere la instalación de dos componentes: la herramienta de *generación de especificaciones* y la herramienta de *generación de secuencias de navegación*. También podrá tener accesible el servidor de Wrappers del entorno de ejecución. La Figura 3.2 muestra la relación entre cada uno de los elementos.

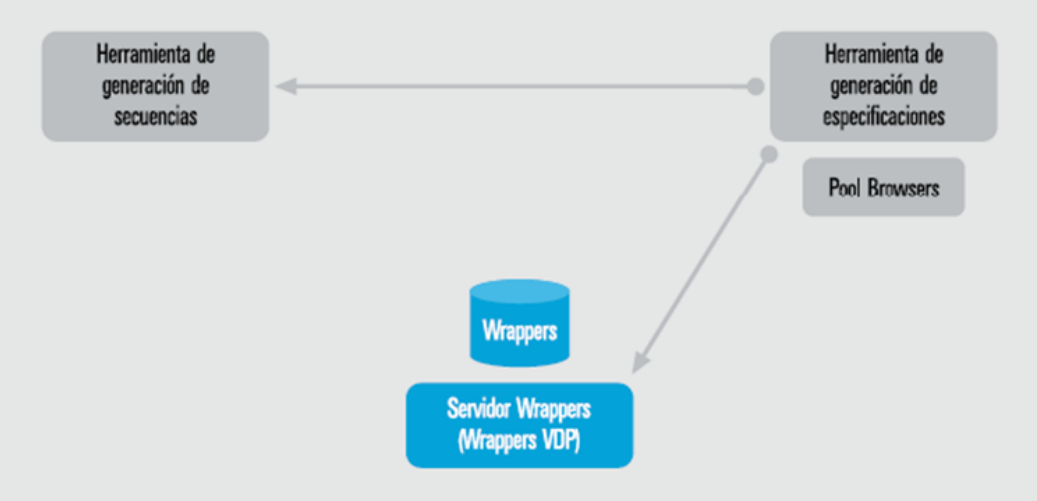


Figura 3.2: Distribución del Entorno de Generación

3.3.2.2. ENTORNO DE EJECUCIÓN

La operación de **Denodo ITPilot** se realiza en el entorno de ejecución, donde se realizan las acciones sobre los Wrappers que encapsulan las fuentes Web cuyos datos se quieren extraer. Se requieren tres componentes en este caso: la herramienta Web de administración, el servidor de Wrappers y el pool de navegadores. La Figura 3.3 describe la relación entre estos elementos.

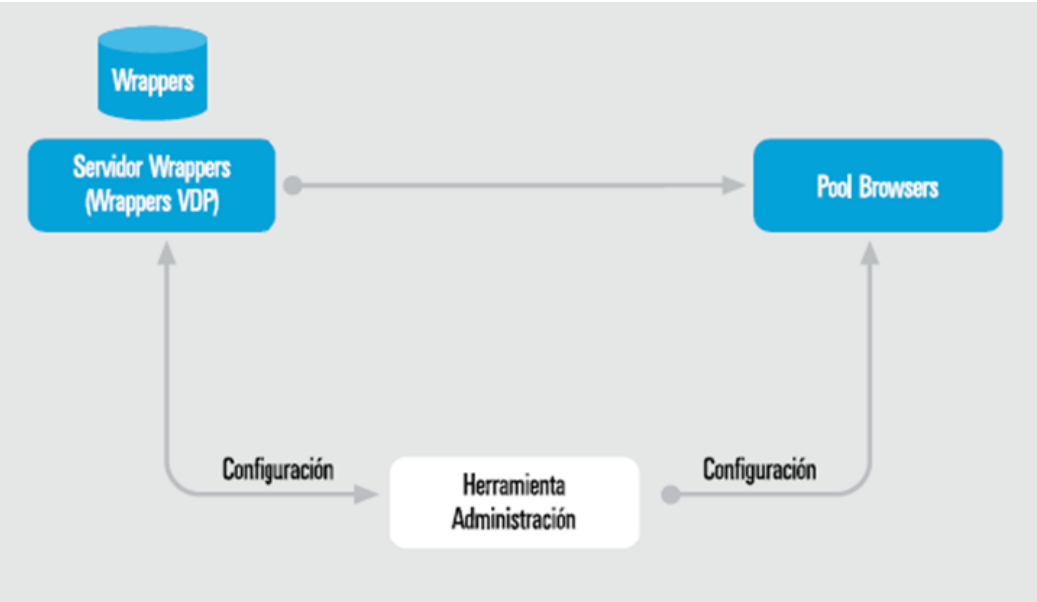


Figura 3.3: Distribución del Entorno de Ejecución

Debido a que el servidor de Wrappers puede ser utilizado en diferentes entornos, se recomienda su instalación en una máquina independiente al resto del sistema. El pool de navegadores puede encontrarse o en la misma máquina que el servidor de Wrappers, o en una máquina independiente; en general, dependerá del número máximo de navegadores que puedan llegar a estar abiertos en la ejecución del sistema.

3.3.2.3. ENTORNO DE MANTENIMIENTO

Este entorno debe ejecutarse junto con el de ejecución, y permite que **ITPilot** pueda monitorizar cambios en las fuentes de las cuáles se están extrayendo datos, y regenere automáticamente aquellos Wrappers que lo requieran. El servidor de mantenimiento, que hace uso de un pool de navegadores, puede ejecutarse en la misma máquina que el servidor de Wrappers, aunque se recomienda su instalación en otra máquina. La Figura 3.4 muestra la relación entre este entorno y el de ejecución.

Figura 3.4: Relación entre Entornos de Ejecución y Mantenimiento

El proceso del servidor de mantenimiento al ejecutar una consulta sobre un Wrapper es el siguiente:

- Esta consulta se envía junto con los resultados producidos al módulo de mantenimiento.
- El módulo de mantenimiento recibe la consulta y sus resultados, ésta se almacena en una base de datos relacional. A estos datos almacenados se les asigna una fecha de expiración. Los resultados caducados se borran de forma periódica.
- Al mismo tiempo que el paso anterior, se ejecutan los test necesarios pasándoles como parámetro la consulta y los resultados.
- Los test devuelven un número entre 0 y 100. 0 significa que la condición no se cumple en absoluto, y 100 que se cumple absolutamente. Este valor es almacenado dentro de un gestor de resultados.

- Se lanza un proceso evaluador que determina si el Wrapper ha cambiado en función del resultado del paso anterior. Si el Wrapper cambia, el sistema de mantenimiento selecciona un subconjunto de todas las consultas almacenadas para regenerar el Wrapper. A partir de estos ejemplos y de la estructura creada durante la fase de generación de especificaciones, el sistema intenta regenerar automáticamente el nuevo Wrapper.

3.3.3. Herramienta gráfica

Esta herramienta, como ya se ha comentado anteriormente, permite extraer visualmente información de fuentes Web. También pueden utilizarse para extraer información de documentos en formato Word y/o PDF. En concreto existen dos herramientas complementarias:

- **La herramienta de Generación de Especificaciones**, que permite crear *Wrappers o envoltorios* Web de una manera fácil e intuitiva. Esta herramienta genera automáticamente programas envoltorio en JavaScript[66].
- **La herramienta de Generación de Secuencias de Navegación**, utilizada para la definición de secuencias de navegación complejas sobre fuentes Web. Esta herramienta genera de manera automática programas NSEQL[67], que podrán ser utilizados en los envoltorios creados con el Generador de Especificaciones.

3.4. Aracne

La plataforma Denodo proporciona funcionalidades para la integración de información de fuentes dispersas, heterogéneas y que pueden presentan un bajo nivel de estructuración. **Denodo Aracne**[58] permite el *rastreo o crawling*, indexación y consulta de información no estructurada en una amplia variedad de formatos.

Entre las principales características de Denodo Aracne se encuentran:

- **Crawling Web**, con capacidad para acceder a páginas que incluyan características como JavaScript, HTML dinámico, autenticación, redirecciones complejas, menús emergentes, etc.
- **Crawling de servidores FTP y de sistemas de ficheros**.
- Posibilidad de **recuperar el contenido de mensajes de correo electrónico** accesibles vía POP3 o IMAP.

- **Crawling de cuentas de correo de Microsoft Exchange Server**.

- **Soporte para los formatos más populares**: HTML, texto, XML, MS Word, RSS (versiones 0.91, 0.92, 1.0 y 2.0), PDF, MS Excel, MS PowerPoint, EML, etc.

- **Búsquedas complejas**: soporte para operadores AND, OR, NOT, +, -, uso de paréntesis, uso de comodines, búsquedas por frase exacta, búsquedas multicampo (título, URL, etc.).

- **Mantenimiento de índices** mediante la eliminación de documentos antiguos, obsoletos, no accesibles, etc.

Denodo Aracne se ejecuta a través de **Denodo Scheduler**[59], planificando y configurando las tareas de crawling.

Denodo Aracne se divide en dos módulos independientes:

- **Aracne Server:** Es el módulo de crawling, es decir, es la herramienta de recuperación automática de información no estructurada disponible en la Web, sistemas de ficheros, servidores de correo electrónico, etc. Denodo Aracne dispone de una serie de crawlers para diferentes fuentes de información no estructurada.

- **Aracne Search/Index Engine:** Es el módulo de indexación y búsqueda sobre índices. Permite almacenar documentos para poder realizar posteriormente búsquedas sobre ellos.

Denodo Aracne también incluye una herramienta de administración de configuración, gestión y búsqueda sobre índices.

Para poder utilizar Aracne, se tiene que utilizar el planificador de tareas Denodo Scheduler. Para ello hay que definir tareas tipo Aracne (ARN) para los motores de crawling implementados por Denodo Aracne.

Por otra parte, el servidor de índices puede también utilizarse en la definición de cualquier tipo de tarea de extracción de Denodo Scheduler, para exportar las tuplas obtenidas como documentos en un índice, de tal forma que se puedan realizar posteriormente búsquedas booleanas sobre ellos.

En la Figura 3.5 se muestra la arquitectura de Denodo Aracne, con sus dos servidores, de crawling e indexación/búsqueda, y su relación con Denodo Scheduler. Adicionalmente, Denodo Aracne posee su propia API de indexación/consulta.

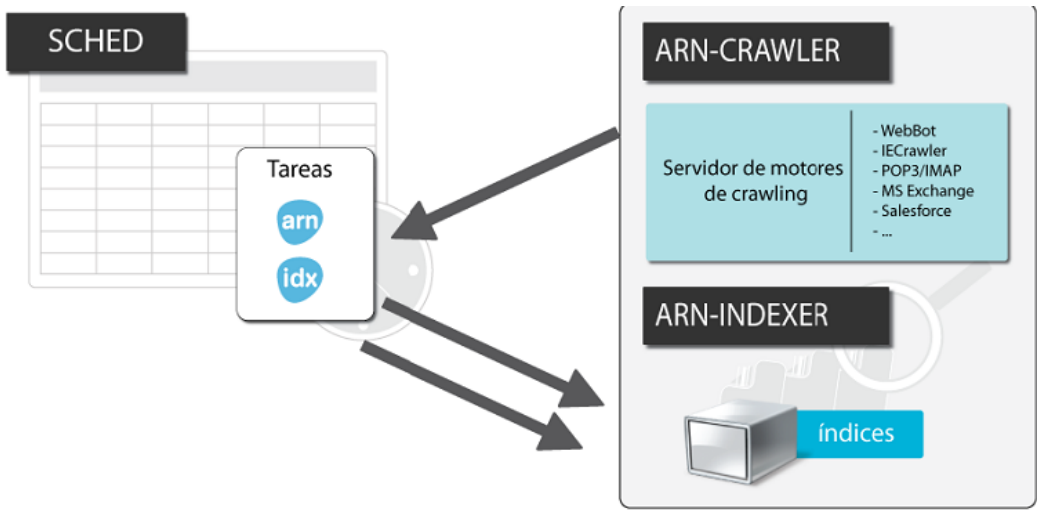


Figura 3.5: Arquitectura Denodo Aracne

El núcleo de ARN-Crawler lo constituyen los robots de crawling, que son los siguientes:

- **IECrawler y Webot**: Parte de una Web inicial y va recorriendo de forma recursiva, teniendo en cuenta el nivel de profundidad, todas las páginas accesibles desde la URL inicial. Además permite conectarse a un servidor FTP y obtener la información contenida en todos los ficheros y subdirectorios de un directorio especificado como URL inicial. **WebBot** es capaz, además, de explorar un sistema de ficheros considerando como URL inicial un directorio y extrayendo la información contenida en todos sus ficheros y subdirectorios.
- **Crawler POP3/IMAP**. Permite recuperar información de correos electrónicos contenidos en servidores accesibles a través de los protocolos POP3 o IMAP. Incluye soporte para ficheros adjuntos.
- **Crawler MS Exchange**. Permite recuperar información de correos electrónicos contenidos en servidores MS Exchange [70]. Incluye soporte para ficheros adjuntos.
- **Crawler Salesforce.com**. Permite recuperar información contenida en entidades de datos accesibles a través de una cuenta en el servicio online Salesforce.com [71].
- **CustomCrawler** permite extraer la información de una fuente de datos, a través de una implementación Java proporcionada por el administrador de Denodo Aracne. Este tipo de robot permite la construcción ad-hoc de un crawler para una fuente específica.

El motor de consulta recibe consultas de los usuarios a través de la interfaz Web o de la API Aracne de búsqueda, recupera los resultados relevantes a esa consulta, utilizando la información contenida en el índice y muestra la respuesta obtenida al usuario en forma de listado de documentos.

El módulo de indexación y búsqueda permite:

- A través de Denodo Scheduler, indexar documentos en diversos formatos.
- Realizar indexaciones y búsquedas de documentos con mayor fiabilidad.
- Representar y realizar consultas sobre las diversas partes de un documento: título, resumen, cuerpo, etc.
- Tener varios índices, lo que posibilita la creación de distintos buscadores temáticos.
- Ordenación de resultados por relevancia basada en el algoritmo TFIDF.
- Búsquedas avanzadas con operadores +, - , *, AND, OR, búsqueda por similitud de palabras (fuzzy), búsquedas por proximidad configurable de los términos, etc.

3.5. Scheduler

3.5.1. Introducción

Denodo Scheduler[59]permite definir, planificar y ejecutar tareas de extracción e integración de datos, definidas sobre los diferentes módulos de la Plataforma Denodo. En combinación con Denodo Scheduler, los módulos de la Plataforma Denodo proporcionan funcionalidades como las siguientes:

- **Virtual DataPort**: Es posible planificar tareas para extraer información de fuentes previamente configuradas en VDP.
- **ITPilot**: al igual que con VDP, es posible planificar la extracción de información a través de un Wrapper de ITP.
- **Aracne**: Como ya se ha comentado, para la utilización de Aracne es necesaria la planificación de tareas de Aracne en Denodo Scheduler.

Entre las principales características de Denodo Scheduler se encuentran:

- **Planificación flexible de tareas batch** sobre los diferentes componentes de la Plataforma Denodo: DataPort, ITPilot y/o Aracne.

- **Generación de informes detallados del resultado de la ejecución de tareas**, incluyendo información detallada de errores.
- Los resultados obtenidos por una tarea pueden **exportarse a un fichero CSV, a una base de datos o a un índice**. Permite también la inclusión de nuevos exportadores desarrollados para un propósito específico.
- Soporte para **extracción de datos de fuentes** con capacidades de consulta limitadas.
- **Tareas persistentes**. Si se reinicia el sistema mientras una tarea se encuentra en ejecución, la tarea puede continuar su ejecución a partir de la última consulta que había sido exitosamente ejecutada.
- **Reintentos** transparentes en caso de fallo.
- Posibilidad de configurar la **ejecución paralela** de las diferentes consultas involucradas en una misma tarea.

3.5.2. Arquitectura

Denodo Scheduler permite definir tareas de extracción para los diferentes componentes de la Plataforma Denodo (VDP,ITP, Aracne). Adicionalmente es posible definir una tarea de tipo JDBC, que explora las tablas especificadas en una base de datos y recupera la información contenida en las mismas. También, permite aplicar distintos algoritmos de filtrado y exporta la información en distintos formatos y repositorios.

Para todas las tareas es posible configurar su planificación temporal (cuándo y con qué periodicidad debe ejecutarse), diversos tipos de filtros para postprocesar la información recuperada por el sistema y la forma en que serán exportados los resultados obtenidos por la tarea. Los exportadores disponibles son:

- Volcado a una base de datos,
- Indexación en el servidor de indexación de Aracne.
- Volcado a un fichero de tipo CSV.

Además, se facilita una API para que el programador pueda realizar un exportador a medida. En la Figura 3.6 se muestra la arquitectura básica del servidor.

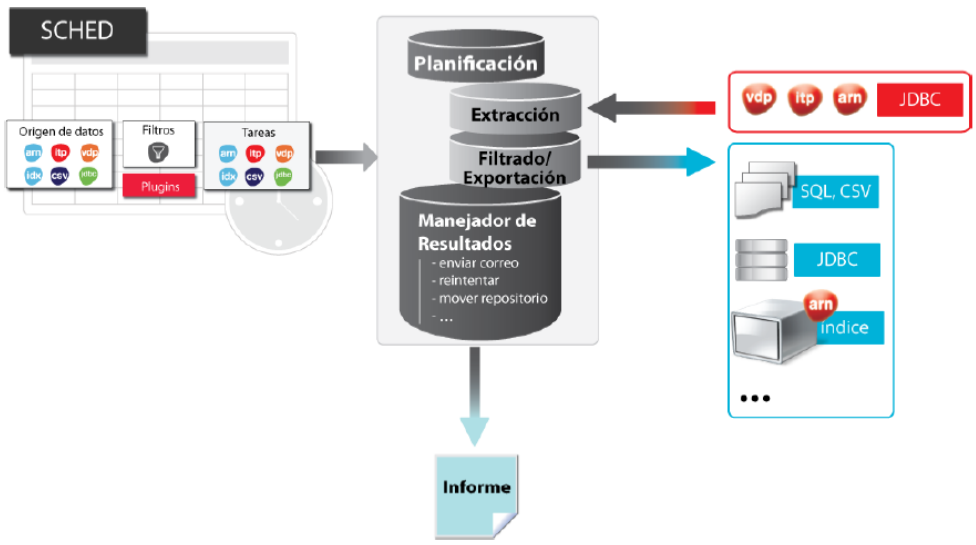


Figura 3.6: Arquitectura de Denodo Scheduler

3.6. Comparativa de Denodo Platform con otras plataformas

En el mercado actual de la virtualización de datos nos encontramos con varias plataformas desarrolladas por grandes empresas. A continuación se hará un repaso de las grandes competidoras de **Denodo Technologies**, sobre todo en virtualización de datos y posteriormente utilizaremos el informe **The Forrester Wave: Information-As-A-Service, Q1 2010**[80], para conocer de forma objetiva como están todas estas plataformas, incluida la plataforma de Denodo Technologies, en el mercado.

Forrester Research[81] es una compañía de investigación de tecnología y del mercado independiente, la cual proporciona informes sobre el impacto tecnológico en los negocios y en los consumidores.

A continuación enumeramos las empresas más importantes y los productos que compiten con la **plataforma Denodo**:

1. **Informatica**

   Informatica Corporation[82] es un proveedor de software de integración de datos. **Informatica** ofrece a sus clientes la posibilidad de consultar, integrar y confiar la información guardada en sistemas tradicionales de la empresa, en sistemas externos y en el cloud (la nube).

   Productos y servicios de Informatica

La plataforma de **Informatica** es una plataforma de software abierta, unificada y completa que está diseñada específicamente para la integración de datos. La plataforma es capaz de acceder a prácticamente cualquier tipo de datos y los hace accesibles, significativos y utilizables para los usuarios y los procesos que los necesitan.

Informatica respalda sus soluciones con un conjunto personalizable y completo de servicios profesionales, de formación y de asistencia al cliente que ayudarán a sacar el máximo provecho al software de Informatica en su entorno.

A continuación se enumeran los módulos de la plataforma de Infomatica más importantes:

- **PowerCenter**[83]: Integra los datos de prácticamente cualquier sistema empresarial, en cualquier formato.
- **B2B Data Transformation**[84]: Extrae información de cualquier archivo, documento o mensaje, independientemente del formato, complejidad o tamaño.
- **PowerExchange**[85]: Permite acceder a todas las fuentes y formatos de datos, entregándolos donde y cuando sea necesario.
- **Data Archive**[86]: Gestiona el incremento del volumen de los datos archivándolos de manera segura. Además, proporciona un acceso continuo a los datos archivados.
- **Data Explorer**[87]: Permite a los propietarios de la información detectar, documentar y resolver los problemas de la calidad de datos.
- **Data Subnet**[88]: Ayuda a reducir el tiempo, el esfuerzo y el espacio de discos necesarios para el soporte de entornos distintos al de producción, mediante la creación, actualización y protección de copias específicas, más reducidas y referencialmente intactas, de datos de producción.
- **Biblioteca de Software AddressDoctor**[89]: La biblioteca de software AddressDoctor es una interfaz de programación de aplicaciones (API) que permite validar direcciones
- **Data Privacy**[90]: ayuda a los departamentos de IT a proteger datos confidenciales o sensibles mediante su enmascaramiento, lo que permite replicarlos de forma segura en sistemas distintos al de producción para realizar tareas de desarrollo, pruebas o formación.
- **Data Quality**[91]: permite a los propietarios de información de negocio y a los departamentos de IT colaborar para limpiar datos y gestionar la calidad de datos en toda la empresa.

- **Cloud**[92]: es una familia de soluciones de integración de datos que permite a las empresas acceder, descubrir, limpiar e integrar los datos que residen en el cloud con los datos de sus sistemas locales de IT
- **Identity Resolution**[93]: permite a los departamentos de IT buscar registros de datos de identidad (incluidos los errores de introducción de datos e intencionales, o bien las variantes de nombres y direcciones internacionales) tanto en modo batch como en tiempo real.
- **Data Services**[94]: Services ofrece datos fiables como servicio a cualquier aplicación, en cualquier latencia, con cualquier protocolo y desde una sola plataforma unificada.
- **Gestión de datos maestros**[95]: permite a las empresas mejorar sus operaciones con acceso en el escritorio a datos consolidados y fiables que son críticos para el negocio.
- **B2B Data Exchange**[96]: Permite a las empresas intercambiar datos de forma sencilla y rentable con su red ampliada de socios comerciales, que incluye a proveedores, canales de distribución y clientes.
- **RulePoint**[97]: Es un software de procesamiento de eventos complejos que proporciona alertas en tiempo real y un conocimiento profundo de la información pertinente que las empresas y los organismos públicos necesitan para actuar de forma más inteligente, rápida, eficaz y competitiva.

2. **Composite**

Cuando **Composite Software**[98] fue fundada en 2002, en esa época las empresas tenían un promedio de datos de sólo un terabyte y un conjunto limitado de usuarios. Sin embargo esta información, junto con la amplitud de los usuarios que necesitan de su acceso, estaba explotando. También lo fueron los plazos de entrega y los costes necesarios para integrar los datos. Para cumplir con los objetivos de negocio, surgió la idea de crear una nueva clase de middleware de integración de datos lo suficientemente ágil para seguir este ritmo de negocio siempre cambiante y las nuevas necesidades de la tecnología.

The Composite Data Virtualization Platform

La plataforma de virtualización de datos de **Composite** integra datos de múltiples fuentes, ubicadas en cualquier lugar de la empresa, de manera unificada para el consumo bajo demanda con una amplia gama de soluciones de negocios de front-end.

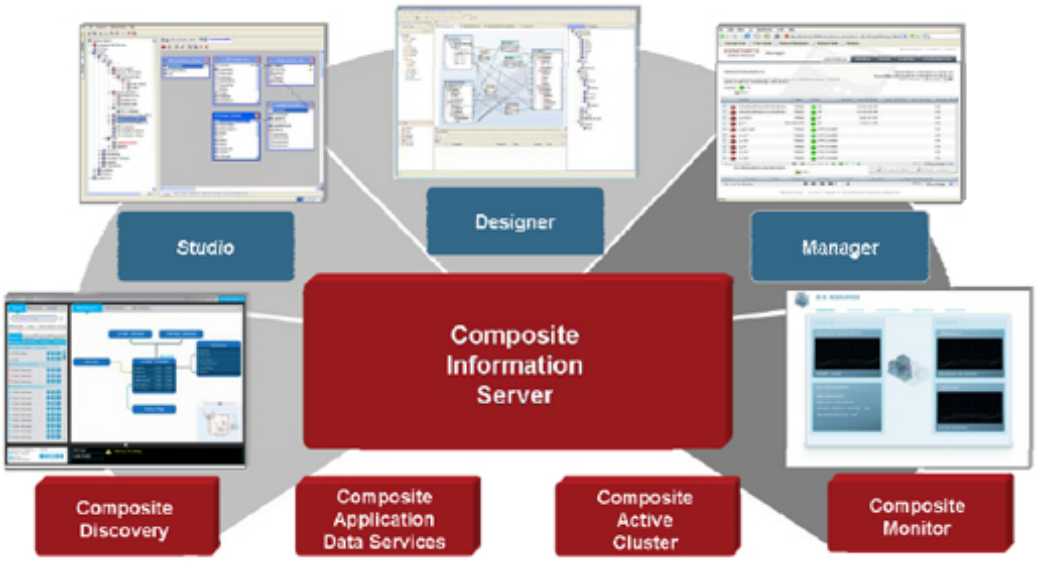


Figura 3.7: The Composite Data Virtualization Platform

La virtualización de datos de la plataforma de **Composite** incluye un software de virtualización que completa del ciclo de vida del desarrollo y operaciones de alta fiabilidad. Durante el desarrollo de requisitos y la fase de diseño a alto nivel, **Composite Discovery**[99] ayuda a descubrir y modelar las entidades clave y las relaciones entre ellas.

En el momento de la construcción, los desarrolladores pueden utilizar el **Composite Information Server**[100], el cual es un entorno de fácil uso para el Desarrollo (relacionales y XML) con generadores de código automatizado, para crear código de alta calidad y compatible con los estándares y servicios de datos. Herramientas que proporcionan uniones complejas y transformación de funciones. Adaptadores estándares que simplifican el acceso y la publicación del desarrollo de actividades. Y la administración del control de seguridad, metadatos, código de fuentes y más...

Composite Applications Data Services[101] automatiza aún más y acelerar el punto de vista crítico y el desarrollo de las actividades de servicios de datos.

En tiempo de ejecución, **Information Server** es un motor de consulta para la realización querys, uniones, resúmenes y entrega datos a soluciones de negocios que las consumen bajo demanda. Además poseen una caché que proporciona más velocidad y flexibilidad.

Composite Monitor[102] y **Composite Active Cluster**[103] proporciona la monitorización balance de carga, alta disponibilidad y recuperación de errores.

3. **IBM**

   IBM[104] posee **IBM InfoSphere DataStage**[105] es una herramienta ETL(Extract, transform, and load) y parte de la suite de **IBM Platforms Solutions** e **IBM InfoSphere**. Incluye una herramienta gráfica para construir soluciones de integración de datos y está disponible en varias versiones, como la Server Edition y Enterprise Edition.

   IBM InfoSphere DataStage permite a las empresas extraer, transformar y cargar datos desde diversos orígenes hasta un almacén de datos. El soporte al hardware de múltiples procesadores que viene incluido permite que DataStage pueda brindar altos niveles de escalabilidad y trabajar eficientemente con grandes volúmenes de datos. Numerosos *conectores* soportan una amplia variedad de formatos de datos de origen y destino, que incluye a los famosos sistemas de gestión de bases de datos de IBM y OEM, las fuentes de datos ODBC, aplicaciones de terceros, mensajes en tiempo real generados por programas de colocación de mensajes en colas y servicios Web, y los formatos de archivos más utilizados.

   DataStage posee **DataStage Designer**[106] para generar trabajos de ETL. Cada trabajo está formado por múltiples *etapas*, y cada una de ellas realiza una tarea determinada. Entre los ejemplos de estas tareas se incluyen la lectura de información desde una fuente de datos, la transformación de los datos de entrada mediante funciones incorporadas, la conversión de un tipo de datos en otro, etc. **DB2 Connector** se puede involucrar con operaciones con XML, y a numerosas operaciones de procesamiento. Existen dos tecnologías de DataStage que resultan fundamentales para estos escenarios: **DB2 Connector**[107] y **XML Pack 2.0**. Y por último se tiene **DataStage Director**[108] para ejecutar el trabajo realizado.

4. **Microsoft**

   Microsoft[109] posee dos plataformas fundamentales en cuanto a virtualización de datos se refiere:

   - **SQL Server 2008**[110], el último lanzamiento de **Microsoft SQL Server**, ofrece una plataforma de datos completa, más segura, confiable, administrable y escalable para aplicaciones críticas. Permite que los desarrolladores creen aplicaciones nuevas, capaces de almacenar y consumir cualquier tipo de datos en cualquier dispositivo, y que todos los usuarios tomen decisiones informadas en base a conocimientos relevantes.

     Microsoft SQL Server es un sistema para la gestión de bases de datos basado en el modelo relacional. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores

de bases de datos como son Oracle o MySQL. **SQL Server 2008** posee las siguientes características:

- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.
- Este sistema incluye una versión reducida, llamada MSDE(Microsoft Data Engine[111]) con el mismo motor de base de datos pero orientado a proyectos más pequeños.
- Es común desarrollar completos proyectos complementando **Microsoft SQL Server** y **Microsoft Access** a través de los llamados ADP (Access Data Project). De esta forma se completa la base de datos (Microsoft SQL Server), con el entorno de desarrollo (VBA Access), a través de la implementación de aplicaciones de dos capas mediante el uso de formularios Windows.
- Para el desarrollo de aplicaciones más complejas (tres o más capas), **Microsoft SQL Server** incluye interfaces de acceso para varias plataformas de desarrollo, entre ellas .NET, pero el servidor sólo está disponible para Sistemas Operativos Windows.

▪ **Microsoft BizTalk Server**[112], a menudo denominado simplemente *BizTalk*, es un servidor business process management (BPM). Por medio del uso de adaptadores diseñados para comunicarse con diferentes tipos de software usados en una empresa de gran tamaño, permite a las compañías automatizar e integrar los procesos de negocio.

5. **Red Hat**

Red Hat[113] es la compañía responsable de la creación y mantenimiento de una distribución del sistema operativo **GNU/Linux** que lleva el mismo nombre: **Red Hat Enterprise Linux**, y de otra más, **Fedora**[116]. Así mismo, en el mundo del middleware patrocina jboss.org, y distribuye la versión profesional bajo la marca **JBoss Enterprise**[117]. **Red Hat** es famoso en todo el mundo por los diferentes esfuerzos orientados a apoyar el movimiento del software libre. No sólo trabajan en el desarrollo de una de las distribuciones más populares de Linux, sino

también en la comercialización de diferentes productos y servicios basados en software de código abierto.

En cuanto a virtualización de datos Red Hat posee dos herramientas fundamentales:

- **MetaMatrix Enterprise Data Services Platform**[114]: Proporciona una manera simple y mejorada de trabajar con los datos a través de diversos sistemas. Ofrece:
  - Un potente sistema de gestión que proporciona herramientas para crear servicios de datos accesibles a través de JDBC, ODBC y protocolos de servicios Web.
  - Un repositorio para la definición de datos almacenados con la metainformación relevante.
  - Una robusta ejecución del entorno que proporciona clases de negocio, integración de datos y seguridad.

  Incluye los siguientes componentes:
  - **MetaMatrix Enterprise Designer**: Está basada en la herramienta de desarrollo Eclipse para la creación y testing de servicios de datos.
  - **MetaMatrix Enterprise Server**: Servidor que ejecuta servicios de datos y proporciona optimización, cacheo y seguridad.
  - **MetaMatrix Repository**: Sistema de gestión de metadatos multiusuario, que incorpora un rico repositorio de metadatos y soporta la compartición de metadatos entre un equipo de un proyecto o toda la empresa.
  - **MetaMatrix Console**: aplicación cliente que proporciona la administración de varios componentes, incluyendo la configuración de equipos, procesos y servicios. Monitorización y acceso al sistema, configuración de seguridad y la administración de otras tareas.
- **JBoss Enterprise SOA Platform**[115]: Es la nueva generación de ESB(Enterprise Service Bus) y la infraestructura para la automatización de procesos de negocio que permite la ejecución de negocios superiores, capacidad de respuesta y la flexibilidad en una plataforma de código abierto. Permite aprovecharse de las IT con métodos de integración que mejoran drásticamente la ejecución de procesos, de negocio, velocidad y calidad.

Se ha podido ver que los competidores de Denodo tienen, en muchas ocasiones, una herramienta muy parecida a la Denodo Platform, y

en otras ocasiones solo cubre una pequeña parte de lo que ofrece la plataforma Denodo, pero de forma más especializada.

El informe **The Forrester Wave: Information-As-A-Service, Q1 2010**[80] presenta a IBM, Informatica, Composite Software y Denodo como "*Líderes en soluciones IaaS* ", es decir en Cloud Computing.

Destaca que IBM "*es uno de los jugadores más agresivos en el espacio IaaS*". Además destaca de su producto, el **IBM InfoSphere Information Server** que "*es el principal producto que ofrece servicios de datos, ofrece la integración con otros productos de **IBM**, incluyendo información Analyzer, QualityStage, DataStage, los CDC, InfoSphere Data Architect, y Federation, para ofrecer una completa plataforma de servicios de datos*".

De **Informatica**, el informe destaca que los productos que ofrece "*son ideales para organizaciones con necesidades complejas de integración de datos*".

En cuanto a **Composite Software** el informe destaca que "*se mueve hacia arriba de un nicho de mercado para ofrecer una solución completa de IaaS*". Además destaca, que con la innovación y el desarrollo de la plataforma han obtenido como resultado "*la compatibilidad con algunas de las más complejas implementaciones de servicios de datos que se ven en los grandes bancos, agencias gubernamentales, productos farmacéuticos, petróleo y gas, y los medios de comunicación, etc...* ".

El informe afirma que " ***Microsoft** y **Red Hat** ofrecen opciones competitivas*", en cuanto a soluciones IaaS. Destaca que "***Microsoft** ha mejorado su **BizTalk Server** y las versiones de **SQL Server** en los últimos dos años para apoyar un volumen aún mayor de datos estructurados, no estructurados, semiestructurados, así como la mejora en la seguridad y disponibilidad de datos, servicios de datos en el desarrollo y la integración, y el soporte para información en tiempo real*".

De **Red Hat** destaca que "*ofrece una alternativa de código abierto* ", y que "*es probable que veamos nuevas características y la innovación en el futuro próximo*".

Por último, de la plataforma de **Denodo Techonologies** afirma que "*ofrece una solución viable para apoyar la mayoría de las implementaciones IaaS*", además destaca que "*el principal objetivo de Denodo Technologies ha sido siempre proporcionar servicios de datos sencillos, de bajo coste y de rápido despliegue, con la opción de escalarlos a un nivel de rendimiento y fiabilidad óptimo para empresas*".

Por lo tanto, se puede afirmar que la Denodo Platform es un middleware, un software de infraestructura que permite:

- Conectarse y obtener datos de muchos tipos diferentes, tanto estructurados (bases de datos, Webservices, listados en aplicaciones Web como Salesforce o Facebook) como no estructurados (hojas de cálculo Excel o Google Docs, Word o PDF's,...).
- Combinar, filtrar y transformar los datos obtenidos para crear nuevos conjuntos de datos y permitir, por ejemplo, tener la información de los sistemas internos de una empresa combinada con datos de proveedores o datos públicos en Internet.
- Consumir los datos, tanto desde otras aplicaciones mediante JDBC, API's o Web Services, como directamente a personas a través de Java Portlets o .Net Web Parts en servidores de portales.

Y con todo lo anterior, además es una plataforma configurable para que sea en real-time o near-real-time, mediante caching.

En definitiva, la Denodo Platform es un software empresarial que permite crear una data layer que abstrae la complejidad de datos en los sistemas origen ofreciendo nuevos conjuntos de datos para ser consumidos por nuevas aplicaciones.

Capítulo 4

Skiptracing

4.1. Introducción

Skiptracing es una aplicación de extracción de información en Internet. Es capaz de obtener información contenida en Internet y en redes sociales. Por lo que, se define el término **Huella digital**.

Se define el término **Huella Digital** como: **Información que se encuentra en Internet de un individuo o empresa, ya pueda ser en una red social, en un blog, foro o cualquier página Web**.

De este término es del que se alimenta la herramienta **Skiptracing**. Es decir, busca en Internet todo tipo de información que se pueda encontrar, tanto en redes sociales, foros, blog como páginas Web generales.

La información que la herramienta es capaz de obtener es únicamente la información pública. Se entiende como información pública, la información que la persona publica en Internet o es de acceso completamente libre, es decir, si el usuario publica su teléfono como privado en su perfil de Facebook, nuestra herramienta no será capaz de obtener dicha información.

Por lo tanto, toda la información obtenida es **totalmente pública y accesible por cualquier persona**, Skiptracing facilita esta búsqueda, además la realiza en distintas fuentes y la muestra de manera simultánea.

4.2. Requisitos

Este capítulo especifica detalladamente los requisitos del proyecto **Skiptracing**. Este documento recolecta, analiza y define las necesidades del Usuario, a alto nivel, y las características del producto. Se debe enfocar en las capacidades que necesita el Usuario, y la razón por la cual éstas existen. De este

modo, se puede desarrollar un modelo del sistema.

La aplicación **Skiptracing** es un sistema que realiza la búsqueda de huella digital de personas o empresas a través de Internet basándose en la tecnología de mediadores y bases de datos virtuales. Se construirá una aplicación que realizará la búsqueda de personas en distintas fuentes, comprobará el grado de validez de dicha búsqueda y se le presentarán los resultados al Usuario.

Además, la herramienta debe permitir no sólo la búsqueda de una persona o empresa, sino también la búsqueda de varias personas simultáneamente, para ello se permite la carga de un fichero de texto con los datos de varias personas a buscar.

4.2.1. Requisitos Funcionales

Los requisitos funcionales de la herramienta se enumeran a continuación:

1. Será capaz de integrar fuentes de información heterogéneas, tanto estructuradas (base de datos), como semiestructuradas (páginas Web), como no estructuradas (documentos).

2. Permitirá combinar la información extraída de las diferentes fuentes y presentar vistas de la información general.

3. Accederá a la información de forma no intrusiva (los datos residen íntegramente en las fuentes y se accede a ellos en tiempo real en cada consulta), únicamente se almacenará información en el caso de la búsqueda masiva.

4. Estará basada en las nuevas tecnologías de bases de datos virtuales.

5. Permitirá acceder no sólo a páginas Web estáticas, sino también dinámicas incluyendo servidores seguros.

6. El Usuario podrá realizar consultas por nombre y apellidos para búsqueda de personas o nombre de empresa para la búsqueda de la misma. Obtendrá resultados en tiempo real de las fuentes de información relevantes.

7. Además, se presentará un formulario de búsqueda avanzada para realizar una búsqueda añadiendo más información. En esta búsqueda no sólo se permitirá añadir más información, sino también se permitirá elegir las fuentes a las que se quiere consultar.

8. Se permitirá cargar un fichero de texto con una línea por persona que se quiere buscar. El resultado de la búsqueda se almacenará en base de datos, pudiendo ser consultado posteriormente.

9. La interfaz de Usuario será amigable tanto en diseño como funcionalidad.

4.2.2. Modelo de requisitos

En este apartado se describe el **Modelo de Requisitos**, basado en Actores y Casos de uso, elaborado a partir del conjunto de características recogido anteriormente.

4.2.2.1. Identificación de Actores

En este caso solo se tiene un tipo de actor, ya que no es necesaria la autenticación para el acceso a la herramienta. El Usuario que emplee la herramienta realizará la búsqueda de personas o empresas. También le será posible la carga de un fichero para poder realizar la búsqueda de varias personas.

4.2.2.2. Identificación de Casos de Uso

A partir de lo recogido en la lista de características, los casos de uso que modelan los requisitos a cubrir son los siguientes:

- **CU_US1**: *Búsqueda de persona*: El Usuario podrá realizar la búsqueda insertando el nombre, primer apellido y segundo apellido. El segundo apellido no es obligatorio.
- **CU_US2**: *Búsqueda de empresas*: El usuario podrá realizar la búsqueda insertando el nombre de la empresa.
- **CU_US3**:*Búsqueda avanzada de personas*: El usuario podrá realizar una búsqueda personalizada de personas. En ella podrá seleccionar las fuentes en las que quiere realizar la búsqueda y si quiere realizar la búsqueda simple o profunda. Además posee más campos para afinar dicha búsqueda.
- **CU_US4**: *Búsqueda avanzada de empresas*: El usuario podrá realizar una búsqueda personalizada de empresas. En ella podrá seleccionar las fuentes en las que quiere realizar la búsqueda y si quiere realizar la búsqueda simple o profunda. Además posee más campos para afinar dicha búsqueda.
- **CU_US5**: *Consulta de detalles de búsqueda de personas*: El Usuario podrá ver los detalles del resultado de búsqueda social. Además podrá acceder a las páginas Web de la que se ha extraído el resto de información.

- **CU_US6**: *Consulta de detalles de búsqueda de empresas*: El Usuario podrá ver los detalles del resultado de búsqueda social. Además podrá acceder a las páginas Web de la que se ha extraído el resto de información.
- **CU_US7**: *Validación de datos*: El Usuario podrá ir añadiendo datos para personalizar la búsqueda lo máximo posible, con el fin de mejorarla. Este caso de uso se da tanto en la búsqueda de personas como en empresas.
- **CU_US8**: *Búsqueda profunda*: El Usuario realizará la búsqueda empleando más combinaciones con los parámetros introducidos, con el fin de mejorarla, con respecto a la búsqueda simple, esta búsqueda se realiza desde el formulario de búsqueda de resultados. Esta opción se da tanto en la búsqueda de personas como empresas.
- **CU_US9**: *Búsqueda simple*: El Usuario podrá realizar la búsqueda que se realiza en el caso CU_US1 o CU_US2 desde el formulario que muestra los resultados de personas o empresas respectivamente.
- **CU_US10**: *Carga de fichero*: El Usuario podrá cargar un fichero de texto con los datos de personas que desea encontrar.
- **CU_US11**: Consulta de ficheros cargados: El Usuario podrá comprobar los ficheros que se cargaron hasta el momento. Y consultar el resultado de cada una de las búsquedas.
- **CU_US12**: *Consulta del resultado del fichero cargado*: El Usuario podrá consultar el resultado de la búsqueda realizada en la carga del fichero.
- **CU_US13**: *Consulta de la búsqueda de cada persona en la carga del fichero*: El Usuario podrá consultar el resultado de cada persona buscada en la carga del fichero.

PRIORIZACIÓN DE CASOS DE USO

El caso de uso de mayor relevancia es el CU_US1, CU_US2, CU_US3, CU_US4 y CU_US10 puesto que son los que permiten al sistema llevar a cabo sus competencias fundamentales.

Le siguen en importancia los casos de uso CU_US8 y CU_US9 que permite realizar la búsqueda profunda y la búsqueda simple de la persona. A continuación se puede destacar el caso de uso CU_US7 que permite la validación de los datos, realizando búsquedas con más información.

A continuación iría el caso de uso CU_US11, CU_US12 y CU_US13 que se encargan de la consulta de los ficheros cargados.

Por último, irían los casos de uso CU_US5 Y CU_US6 relacionados con los detalles de los resultados de la consulta.

4.3. Arquitectura

A continuación se muestra un diagrama con la arquitectura del proyecto Skiptracing. En ella se puede observar donde están cada uno de los módulos que forma parte del proyecto así como la relación entre ellos.

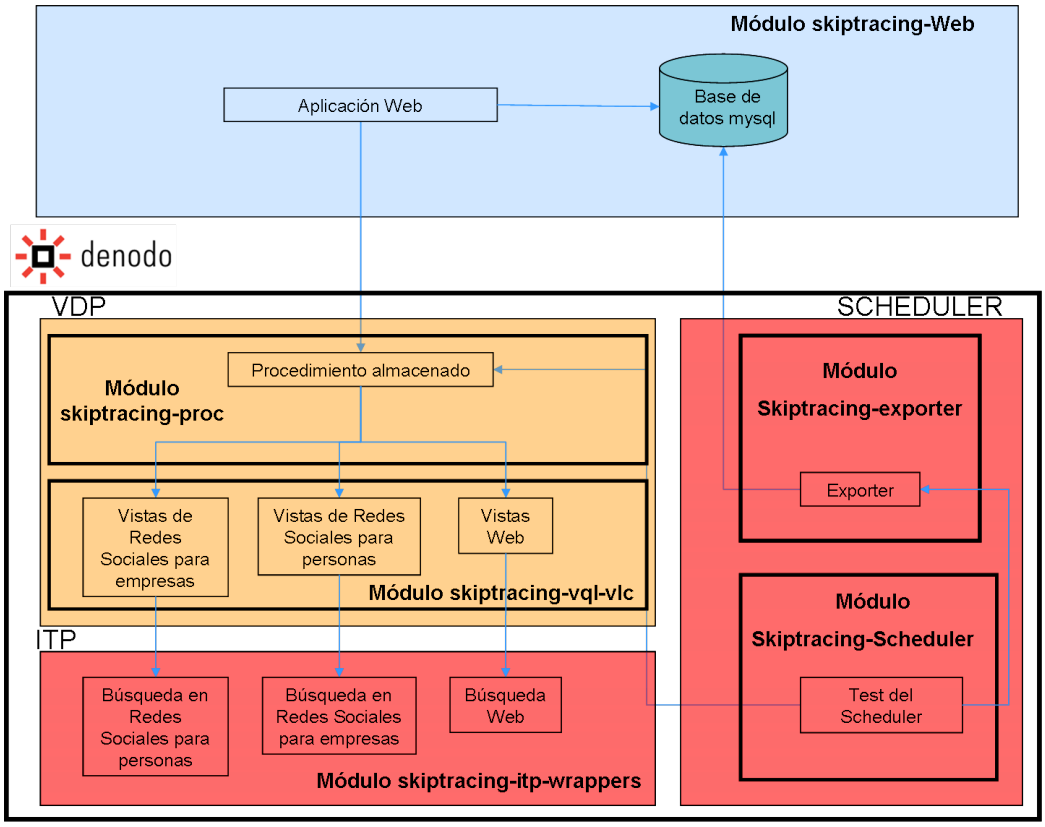


Figura 4.1: Arquitectura Skiptracing

La aplicación Skiptracing está formada por los siguientes módulos:

4.3.1. SKIPTRACING-ITP-WRAPPERS

Contiene los Wrapper de ITP[57] que realizan la navegación a través de las distintas fuentes que se van a emplear en la aplicación. Se explica más en detalle la extracción de información con dichos Wrapper en el apartado de **Análisis de fuentes**.

Los Wrappers que forman parte del proyecto son:

- Itp_google: Wrapper que ejecuta la búsqueda en www.google.com en función de unos parámetros de entrada.
- Itp_google_news: Wrapper que ejecuta la búsqueda en http://news.google.es/ en función de unos parámetros de entrada.
- Itp_google_blogs: Wrapper que ejecuta la búsqueda en http://blogsearch.google.es/ en función de unos parámetros de entrada.
- Itp_facebook: Wrapper que realiza la búsqueda de personas en la red social Facebook.
- Itp_facebook_google: Wrapper que realiza la búsqueda en la Web social Facebook a través del buscador Google.
- Itp_linkedin: Wrapper que realiza la búsqueda de personas en la red social Linkedin.
- Itp_linkedincompany: Wrapper que realiza la búsqueda de empresas en Linkedin. Como Linkedin no tiene búsqueda de empresas, a no ser que estés autenticado en la Web, se realiza la búsqueda a través de Google, utilizando la búsqueda avanzada de dicho buscador.
- Itp_sonico: Wrapper que realiza la búsqueda en la red social Sonico. Como Sonico por sí solo no tiene buscador, sino que utiliza Google, se realiza un filtrado para obtener únicamente los resultados de Sonico.
- Itp_paginasamarillas: Wrapper que realiza la búsqueda en la página Web de páginas amarillas.

4.3.2. SKIPTRACING-SCHEDULER

Contiene la información necesaria relativa a los Datasource y los Jobs[59] para la programación de la búsqueda de personas a través de la subida de un fichero de texto. Este módulo es el encargado de la carga del fichero de búsqueda de personas. Esta tarea se puede automatizar o lanzar una vez se ha cargado el fichero. Se obtendrá cada persona cargada del fichero, se realizará una búsqueda y una posterior clasificación, mostrando finalmente el resultado de cada búsqueda.

Para la clasificación de los resultados se va a tener un umbral que se configurará en el código del exporter. Los resultados con validez mayor que el umbral se clasificarán como **Found** y los resultados con validez inferior como **NotFound**, como se verá a continuación.

Para la realización de estas tareas existen dos Jobs:

1. Test1: Este test del Scheduler realiza la búsqueda de personas en redes sociales. Realiza los siguientes pasos:
   - Obtiene cada una de las personas contenidas en el fichero.
   - Realiza la Query de la búsqueda en redes sociales.
   - Una vez ha ejecutado la Query, asigna un valor que indica la validez a los resultados y almacena estos resultados en la base de datos, en la tabla **person_result**.
   - Una vez se tienen los resultados, se mira la validez. Si la validez alcanzada en la búsqueda es mayor que un cierto umbral se asigna el valor **Found** en el campo **FOUND** de la tabla **person**. Si por el contrario se encuentra datos pero su validez es inferior al umbral, almacena la persona con el valor *Not Found*.
2. Test2: Este test del Scheduler revisa todos los nombres que contiene el fichero de carga, revisando cada uno para saber si está o no en la tabla **person**. Si alguno de ellos no se encuentra en dicha tabla, inserta una tupla en la tabla **person** con el valor del campo FOUND a **NotFound**.

Una vez se sube el fichero de búsqueda de la aplicación Web se ejecutará, según lo programado, el Scheduler lanzando los dos Jobs mencionados anteriormente.

4.3.3. SKIPTRACING-EXPORTER

Contiene el código necesario para guardar los resultados, consolidados y los obtenidos en la búsqueda a través de fichero, en la base de datos. El exporter clasifica la información guardando la información en distintas tablas de la base de datos. El exporter se denomina Skiptracing Exporter para el acceso desde el Scheduler.

4.3.4. SKIPTRACING-VDP-VQL

Este módulo contiene los ficheros vql[69] que generaran las vistas y las relaciones entre ellas. Este módulo consta de dos ficheros vql:

- Skiptracing-model: Este fichero contiene el modelo de vistas base/derivadas de las consultas.
- skiptracing_upload_csv: Contiene las vistas para leer el fichero subido a través de la Web.

Para generar la estructura virtual de datos se cargan estos ficheros en la herramienta Denodo VDP.

4.3.5. SKIPTRACING-PROC

Contiene el código del procedimiento almacenado que se subirá a VDP. El procedimiento almacenado obtiene los campos introducidos por el usuario, construye las querys de búsqueda realizando combinaciones con los parámetros de entrada y finalmente clasifica los resultados obtenidos devolviéndolos a la aplicación Web. En definitiva, encapsula la funcionalidad de buscar, procesar y clasificar la búsqueda de Skiptracing.

Para el funcionamiento del procedimiento almacenado es necesario el uso de ficheros de configuración. Estos ficheros se emplean dentro del procedimiento almacenado para las siguientes tareas:

- Construcción de la consulta que se realizará a la vista correspondiente. Dentro del fichero de configuración existe una consulta genérica, a dicha consulta se le pasarán distintos parámetros, con la combinación de estos parámetros tendremos varias consultas genéricas que se juntarán en una consulta más compleja.
- Cuando se obtienen los resultados, a cada tupla obtenida se le da una puntuación, dicha puntuación se calcula a partir de los parámetros que contienen los ficheros de configuración.
- Dentro del fichero de configuración existen unas reglas con una puntuación. Por cada regla que cumpla el resultado se le sumará el valor asociado de dicha regla. Así cuantas más reglas cumpla la tupla obtenida, tendrá más puntuación.
- Hay un nivel de tuplas aceptadas, es decir, si la tupla no cumple con una puntuación mínima está será descartada.
- Permite la configuración de agrupación de tuplas. Esta agrupación se da cuando coinciden ciertos campos, que contiene el fichero .properties. Esta opción permite agrupar resultados de la misma persona encontradas en distintas fuentes.

Los ficheros de configuración mencionados, se encargan de construir la Query que se enviará a cada una de las vistas. Estos ficheros poseen combinaciones de los parámetros de entrada para conseguir el máximo número de combinaciones, para poder obtener la máxima información en Internet. Para obtener más información sobre los ficheros de configuración, estructura y tipos de ficheros que hay, existe más información en el **Apéndice 3**.

Hay que destacar que existen diferentes ficheros de configuración para la búsqueda simple y para la búsqueda profunda. Empleando en cada caso unas consultas diferentes. La búsqueda profunda es más compleja y por lo tanto el resultado se muestra con más retardo, pero es más efectiva.

4.3.6. SKIPTRACING-WEB

Este módulo contiene el código de la aplicación Web que proporciona un interfaz al usuario para realizar la búsqueda y visualizar los resultados. Este módulo está desarrollado empleando las tecnologías **Wicket**[22], **Hibernate**[34] y **Spring**[36]

La base de datos que emplea la aplicación es:

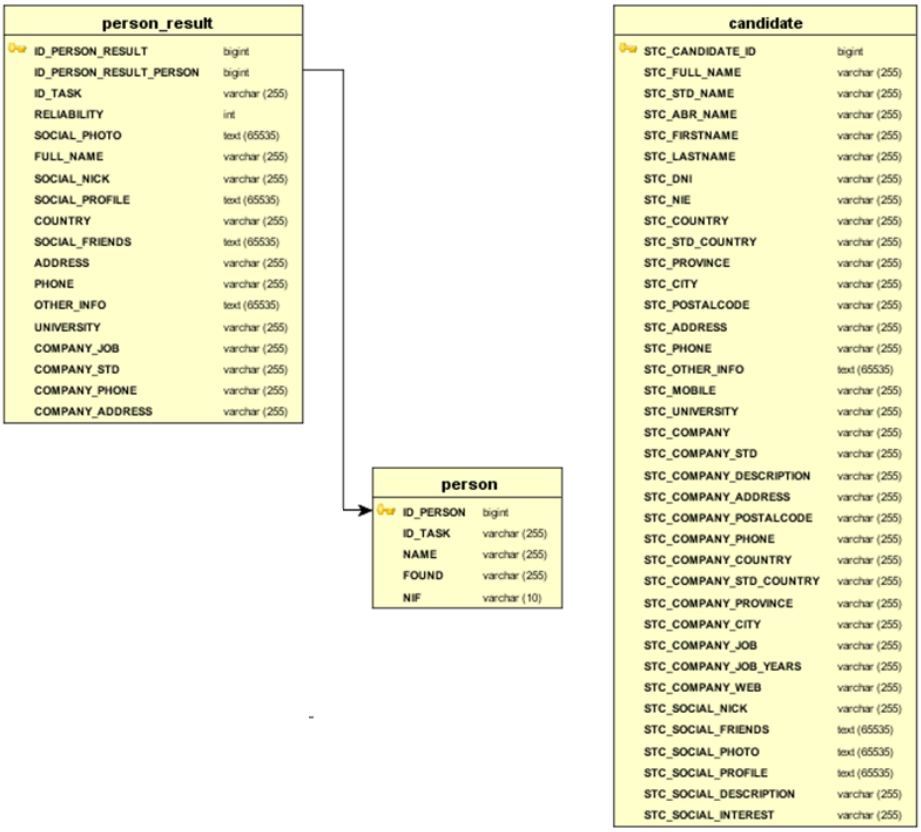


Figura 4.2: Modelo de datos

Las tablas de la base de datos son:

- **Person_result**: Se almacenan los resultados de las búsquedas en redes sociales cuando se realizan a través de la carga de fichero.
- **Person**: Almacena las personas que se quieren buscar dentro del fichero de carga con el resultado de la búsqueda.
- **Candidate**: Almacena los datos validados de las personas. Es decir, cuando en la aplicación Web, en la página que muestra los resultados,

se pulsa *Validar Datos* se almacenan los datos que contiene el formulario de dicha página en base de datos.

4.4. Análisis de fuentes

4.4.1. Fuentes Web

En este capítulo se describen las fuentes a emplear haciendo un análisis detallado de la información que pueden proporcionar, la ubicación y la forma de obtenerla.

4.4.1.1. GOOGLE

1. Información general.

   - Nombre de la fuente: **Google**
   - URL inicial: **http://www.google.es/**, introduciendo **REAL_KEYWORDS** con la palabra que se quiere buscar.

Figura 4.3: Búsqueda en Google

   - Parámetros de entrada:
     - **KEYWORDS**: Parámetro de entrada con el que se realiza la búsqueda en Google. Dicho parámetro se inserta en la URL anteriormente mencionada.
     - **SOURCE**: Parámetro que indica si se debe o no realizar la búsqueda en esta fuente.
   - Descripción de flujo de trabajo:

a) Inicialmente se normaliza el parámetro poniéndolo en minúscula.

b) Se inserta el parámetro normalizado en la URL

- Paginación:
  - En la página donde se encuentran los resultados, en la parte inferior, se encuentran enlazadas varias páginas de resultado, sobre las que el Wrapper realizará un máximo de tres iteraciones, recorriendo cada una de las páginas.
- Datos de Salida: De la pantalla de la Figura 4.4 se obtienen los datos siguientes:

Figura 4.4: Resultado en Google

- **G_SUMMARY**: Resumen de cada uno de los resultados de la búsqueda de Google.
- **G_SOURCE**: Página principal de cada uno de los resultados de la búsqueda de Google.
- **G_URL**: URL completa de cada uno de los resultados de la búsqueda de Google.
- **G_TITLE**: Título de cada uno de los resultados de la búsqueda de Google.

2. Información de Ejecución

   Este Wrapper se ejecuta en tiempo real. Para realizar su ejecución se importa dicho Wrapper a **Denodo Virtual Data Port** creando la vista base **base_itp_google**.

4.4.1.2. GOOGLE BLOGS

1. Información general.

   - Nombre de la fuente: **Google**
   - URL inicial: **http://blogsearch.google.es/**, introduciendo **REAL_KEYWORDS** con la palabra que se quiere buscar.

Figura 4.5: Búsqueda de Blogs en Google

- Parámetros de entrada:
  - **KEYWORDS**: Parámetro de entrada con el que se realiza la búsqueda en Google. Dicho parámetro se inserta en la URL anteriormente mencionada.
  - **SOURCE**: Parámetro que indica si se debe o no buscar en Blogs.
- Descripción de flujo de trabajo:
  a) Inicialmente se normaliza el parámetro poniéndolo en minúscula.
  b) Se inserta el parámetro normalizado en la URL
- Paginación:
  - En la página donde se encuentran los resultados, en la parte inferior, se encuentran enlazadas varias páginas de resultado,

sobre las que el Wrapper realizará un máximo de tres iteraciones, recorriendo las páginas.

- Datos de Salida: De la pantalla de la Figura 4.5 se obtienen los datos siguientes:

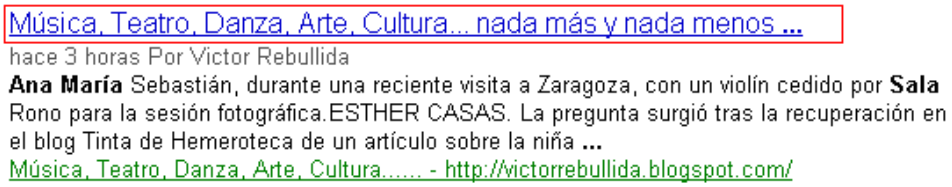


Figura 4.6: Resultado de blogs en Google

- **G_SUMMARY**: Resumen de cada uno de los resultados de la búsqueda en blogs.
- **G_URL**: URL completa de cada uno de los resultados de la búsqueda en blogs.
- **G_TITLE**: Título de cada uno de los resultados de la búsqueda en blogs.

2. Información de Ejecución

   Este Wrapper se ejecuta en tiempo real. Para realizar su ejecución se importa dicho Wrapper a **Denodo Virtual Data Port** creando la vista base **base_itp_google_blogs**.

4.4.1.3. GOOGLE NEWS

1. Información general.

   - Nombre de la fuente: **Google**
   - URL inicial: **http://news.google.es**, introduciendo **REAL_KEYWORDS** como la palabra clave a buscar.

Ana María Matute y Álex de la Iglesia, entre los protagonistas de ...
santanderciudadviva.com - 26/05/2010
La novelista **Ana María** Matute; el director de cine y presidente de la Academia de las Artes y las Ciencias Cinematográficas de España, Álex de la Iglesia; ...
UIMP ofrece en verano 15 ciclos y 120 propuestas culturales, con ... 20 minutos
Santander, escenario UIMP El Diario Montañés
los 17 artículos informativos » Enviar noticia

Acción, amor y fantasía
La Rioja - 21/05/2010
... se arrugaron contra la butaca en algún momento Carmen Roldan, Javier Rodríguez, **Ana María** Marrodán, Carlos Ezquerra, Paulino Martínez y Elena **Fernández**. ...

AVISOS FUNEBRES
LaRepública.com.uy - 23/05/2010
Su esposo: Eduardo Rebellatto; sus hijos: Juan Pablo y **María** Cecilia Rebellatto Ceruti; su padre: Walter Mario Ceruti; sus hermanos: **Ana María** Ceruti, ...

Clausura en un ambiente festivo
Información - 17/05/2010
En la femenina, la campeona fue Lourdes Gallego, mientras que **Ana María Fernández** Martínez alcanzó la segunda plaza. Los campeones en pádel masculino fueron ...

AVISOS FUNEBRES
LaRepública.com.uy - 26/05/2010
Sus hijos: Juan, Francis, **Ana María** y Alejandra; sus nietos; bisnietos; hijos políticos; hermanos; sobrinos y demás familiares participan con profundo dolor ...

Figura 4.7: Búsqueda de Blogs en Google

- Parámetros de entrada:
  - **KEYWORDS**: Parámetro de entrada con el que se realiza la búsqueda en Noticias. Dicho parámetro se inserta en la URL anteriormente mencionada.
  - **SOURCE**: Parámetro que dice si hay o no que buscar en Noticias.
- Descripción de flujo de trabajo:
  a) Inicialmente se normaliza el parámetro poniéndolo en minúscula.
  b) Se inserta el parámetro normalizado en la URL
- Paginación:
  - En la página donde se encuentran los resultados, en la parte inferior, se encuentran enlazadas varias páginas de resultado, sobre las que el Wrapper realizará un máximo de tres iteraciones, recorriendo las páginas.
- Datos de Salida: De la pantalla de la Figura 4.7 se obtienen los datos siguientes:

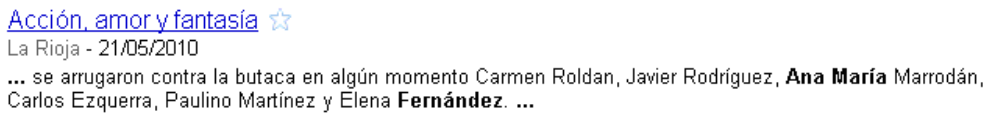

Acción, amor y fantasía
La Rioja - 21/05/2010
... se arrugaron contra la butaca en algún momento Carmen Roldan, Javier Rodríguez, **Ana María** Marrodán, Carlos Ezquerra, Paulino Martínez y Elena **Fernández**. ...

Figura 4.8: Resultado de noticias en Google

- **G_SUMMARY**: Resumen de cada uno de los resultados de la búsqueda en noticias.
- **G_URL**: URL completa de cada uno de los resultados de la búsqueda en noticias.
- **G_TITLE**: Título de cada uno de los resultados de la búsqueda en noticias.

2. Información de Ejecución

   Este Wrapper se ejecuta en tiempo real. Para realizar su ejecución se importa dicho Wrapper a **Denodo Virtual Data Port** creando la vista base **base_itp_google_news**.

4.4.1.4. FACEBOOK

1. Información general.

   - Nombre de la fuente: **Facebook**
   - URL inicial: **http://www.facebook.com/find-friends/?ref=pf**

Figura 4.9: Búsqueda en facebook

- Parámetros de entrada:
  - **FIRSTNAME**: Nombre de la persona que se quiere buscar.
  - **LASTNAME**: Apellido o apellidos de la persona que se quiere buscar.
  - **SOURCE**: Parámetro que dice si hay o no que buscar en Facebook.

  El parámetro FIRSTNAME Y LASTNAME se concatenan y se introducen en el campo marcado en la Figura 4.9, bajo la etiqueta **Búsqueda de Personas**.

- Descripción de flujo de trabajo:

  Inicialmente se realiza la navegación a la URL indicada anteriormente.

  a) Se obtiene el nombre completo de la concatenación de los parámetros FIRSTNAME y LASTNAME.

  b) Se inserta el nombre completo en la casilla mostrada en la Figura 4.9 **Búsqueda de personas**.

c) Se pulsa el botón de buscar (el icono de la lupa) y a continuación Facebook realiza la búsqueda.

- Paginación:
  - En la página donde se encuentran los resultados, en la parte superior izquierda, se encuentran enlazadas varias páginas de resultado, sobre las que el Wrapper realizará un máximo de tres iteraciones, recorriendo las páginas.
  - En el listado de la Figura 4.10 mostrado, se realiza un clic en cada persona encontrada. Llegando a la página que muestra Figura 4.11, de la que se extraen los datos de salida:

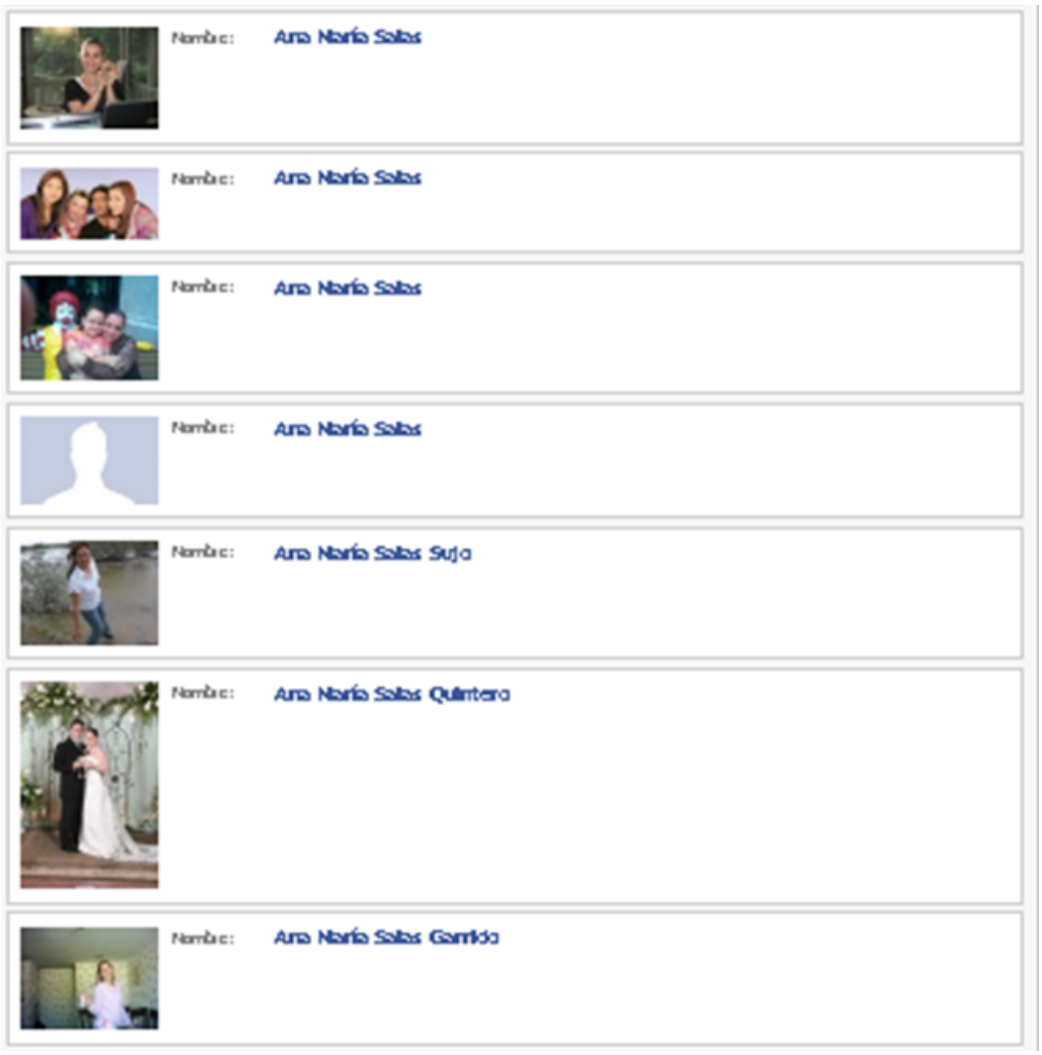


Figura 4.10: Resultado búsqueda Facebook

- Datos de Salida: De la pantalla de la Figura 4.11 se obtienen los

datos siguientes:

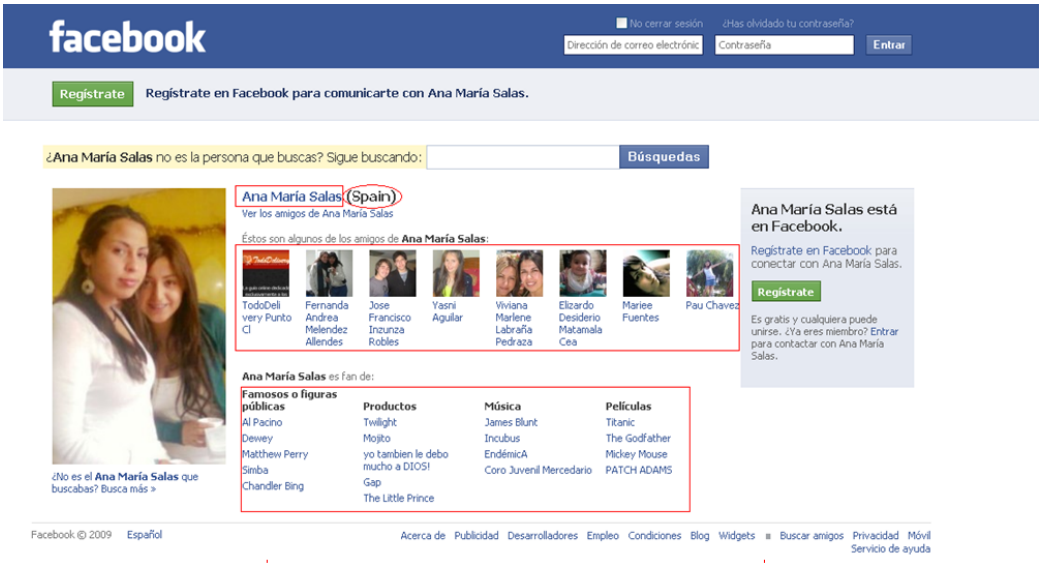


Figura 4.11: Resultado búsqueda Facebook

Cada uno de los resultados encontrados tendrá los siguientes campos de salida:

- **FB_COUNTRY**: País de la persona. En la Figura 4.11, se obtienen el país de la persona encontrada, se encuentra al lado del nombre completo de la persona.
- **FB_DETAIL_URL**: URL donde se ha extraído la información.
- **FB_NAME_DETAILS**: Nombre completo del usuario encontrado en Facebook. Este parámetro se obtiene del primer recuadro rojo que se puede observar en la pantalla de la Figura 4.11.
- **FB_FRIENDS**: Amigos del usuario. Este valor será un array con varios nombres de amigos, dichos amigos se obtienen del segundo recuadro mostrado en la Figura 4.11.
- **FB_REDES**: Redes a las que pertenece se obtiene del último recuadro mostrado en la Figura 4.11.
- **FB_NICK**: Nick que utiliza en facebook. Este Nick se obtiene de la URL donde están los detalles, es decir, de la URL de la página de la Figura 4.11. En este caso la URL es http://es-la.facebook.com/ania.salas por lo que el Nick será ania.salas.
- **FB_NAME**: Nombre del usuario. Nombre insertado en la búsqueda, es decir, el insertado en la Figura 4.9.

- **FB_IMAGE_URL**: URL de la imagen de Facebook.

2. Información de Ejecución

   Este Wrapper se ejecuta en tiempo real. Para realizar su ejecución se importa dicho Wrapper a **Denodo Virtual Data Port** creando la vista base **base_itp_facebook**.

4.4.1.5. LINKEDIN

1. Información general.

   - Nombre de la fuente: Linkedin
   - URL inicial: **http://www.linkedin.com/**, introduciendo **LASTNAME** y **FIRSTNAME** con el nombre y apellidos de la persona que se quiere buscar.
   - Parámetros de entrada:
     - **FIRSTNAME**: Nombre de la persona que se quiere buscar.
     - **LASTNAME**: Apellido o apellidos de la persona que se quiere buscar.
     - **SOURCE**: Parámetro que dice si hay o no que buscar en Linkedin.

     Ambos parámetros se insertan en la URL anteriormente mostrada.
   - Descripción de flujo de trabajo:

     a) Se inserta en el navegador la URL anteriormente mostrada con los valores de los parámetros de entrada.

     b) Tras realizar la búsqueda existen dos posibilidades. Si se obtienen varios resultados, Linkedin mostrará un listado tal y como se observa en la Figura 4.12, si por el contrario, sólo se obtiene un resultado irá directamente a la pantalla de dicho resultado como muestra Figura 4.13.

Ana María Salas Fernández tú
Software Engineer at Denodo Technologies
Madrid y alrededores, España | Telecomunicaciones

Ana María Rosselot Salas Fuera de la red
--
Chile

Ana Maria Guerrero Salas Fuera de la red
ADUANAS en SHCP
Querétaro y alrededores, México | Administración gubernamental

ANA MARIA DEL CARMEN ALVARADO SALAS Fuera de la red
PSICOLOGA en INPET
Perú | Consultoría de estrategia y operaciones

actriz diseñadora de interiores productora de moda
Argentina | Gabinetes estratégicos

Gerente General de COOLT SAC
Perú | Entretenimiento

Ana Salas Ana-Maria SALAS Fuera de la red
INGENIERO en AREVA T&D
Madrid y alrededores, España | Manufactura eléctrica/electrónica

Figura 4.12: Listado búsqueda Linkedin

En este caso se observa un listado con los resultados de la búsqueda.

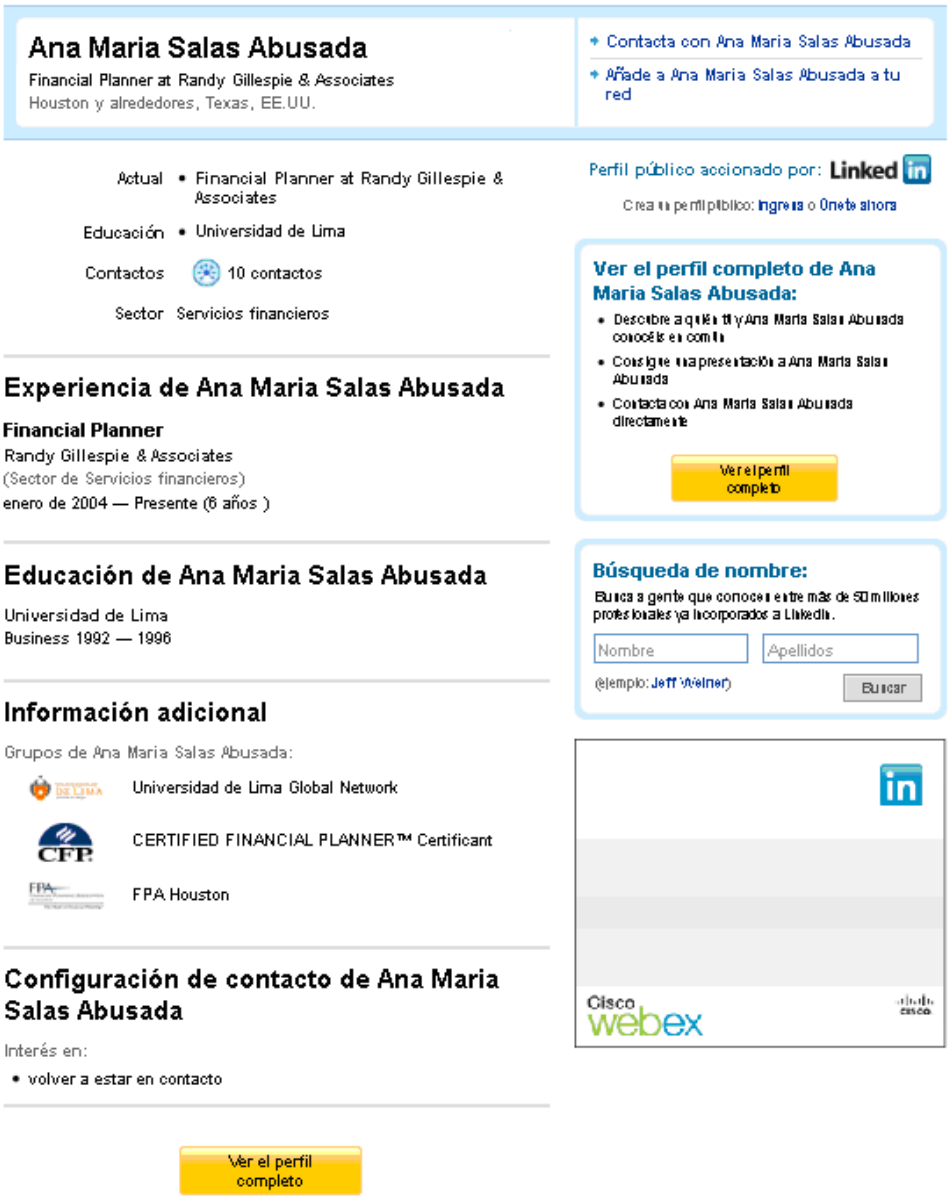


Figura 4.13: Resultado único Linkedin

En este segundo resultado, al sólo existir una persona que encaje con la búsqueda, Linkedin redirige automáticamente a la página de detalles.

- Paginación:
  - En el listado de la Figura 4.12 anteriormente mostrado, se realiza un clic en cada persona encontrada, obtenido para cada resultado una página similar a la mostrada en la Figura 4.13: Resultado único Linkedin.

- Datos de Salida: De la pantalla de la Figura 4.14 se obtienen los siguientes datos:

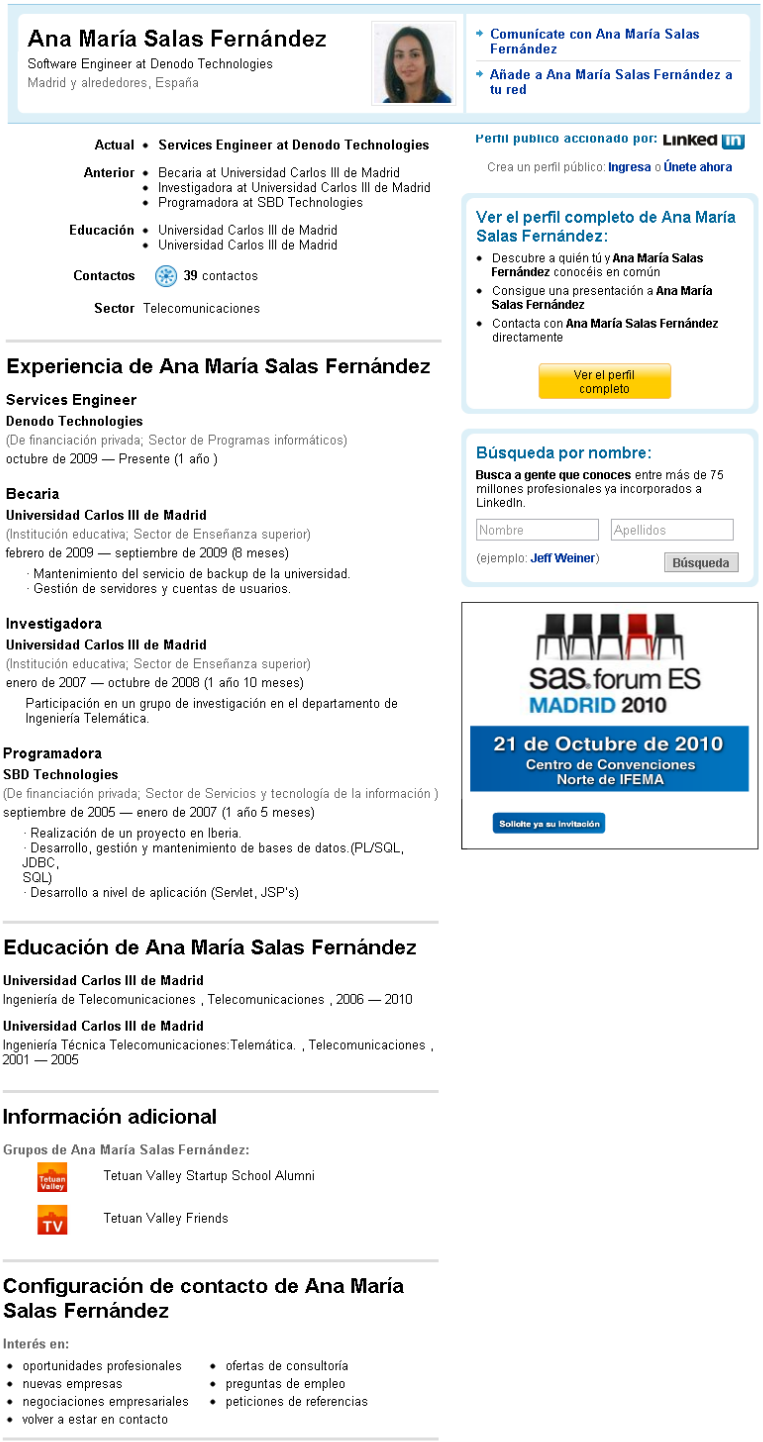


Figura 4.14: Resultado Linkedin I

Cada uno de los resultados encontrados tendrá los siguientes campos de salida:

- **LI_SUMMARY**: Resumen de la persona encontrada. A continuación, en la Figura 4.15, se muestra una parte del resultado, en ella se recuadra el valor del actual campo, como segundo campo recuadrado.
- **LI_FULLNAME**: Nombre completo. Este campo se muestra recuadrado en la Figura 4.15, corresponde al primer campo recuadrado.
- **LI_COUNTRY**: País, se obtiene del segundo recuadro de la Figura 4.15.

Figura 4.15: Resultado Linkedin II

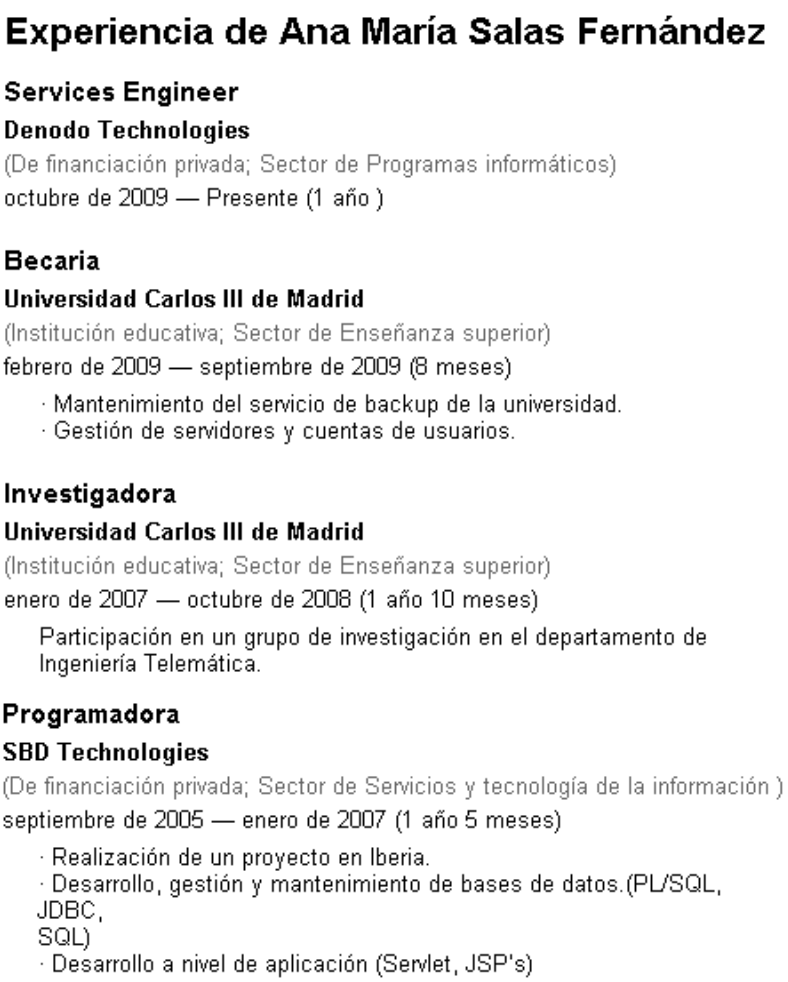
- **LI_CURRENT_COMPANY**: Empresa en la que trabaja actualmente. A continuación, en la Figura 4.16, se muestra parte del resultado de la búsqueda y se recuadra el valor del actual campo, junto a la etiqueta *Actual*.

Figura 4.16: Resultado Linkedin III

- **LI_CURRENT_JOB**: Cargo que desempeña en la empresa actual. Este valor se obtiene, al igual que en caso anterior,

del recuadro mostrado en la Figura 4.16, junto a la etiqueta *Actual*.

- **LI_INDUSTRY**: Industria a la que pertenece. Este valor se obtiene de la página mostrada en la Figura 4.16, junto a la etiqueta *Sector*.
- **LI_UNIVERSITY**: Universidad en la que estudió. Este valor se obtiene de la página mostrada en la Figura 4.16, junto a la etiqueta *Educación*.
- **LI_JOB**: Nombre completo del trabajo que desempeña y la compañía. Ese valor se obtiene en el apartado *Experiencia de usuario*, como se muestra en la Figura 4.17.



Experiencia de Ana María Salas Fernández

Services Engineer

Denodo Technologies

(De financiación privada; Sector de Programas informáticos)

octubre de 2009 — Presente (1 año)

Becaria

Universidad Carlos III de Madrid

(Institución educativa; Sector de Enseñanza superior)

febrero de 2009 — septiembre de 2009 (8 meses)

· Mantenimiento del servicio de backup de la universidad.
· Gestión de servidores y cuentas de usuarios.

Investigadora

Universidad Carlos III de Madrid

(Institución educativa; Sector de Enseñanza superior)

enero de 2007 — octubre de 2008 (1 año 10 meses)

Participación en un grupo de investigación en el departamento de Ingeniería Telemática.

Programadora

SBD Technologies

(De financiación privada; Sector de Servicios y tecnología de la información)

septiembre de 2005 — enero de 2007 (1 año 5 meses)

· Realización de un proyecto en Iberia.
· Desarrollo, gestión y mantenimiento de bases de datos.(PL/SQL, JDBC, SQL)
· Desarrollo a nivel de aplicación (Servlet, JSP's)

Figura 4.17: Resultado Linkedin II

- **LI_PAST_JOB**: Lista de trabajos desempeñados en el pasado. Esta lista se obtiene de la pantalla de la Figura 4.16.

- **LI_EXP_DATA**: Lista de trabajos desempeñados en el pasado con información adicional. Esta lista se obtiene de la pantalla de la Figura 4.16. En estos campos se almacena además de la lista de trabajos la información adicional de ellos.
- **LI_NICK**: Nick que el usuario emplea en Linkedin. Este valor se obtiene de la URL de detalles.
- **LI_IMAGE_URL**: URL de la imagen.
- **LI_DETAIL_URL**: URL de la página de la que se extrae la información.

2. Información de Ejecución

   Este Wrapper se ejecuta en tiempo real. Para realizar su ejecución se importa dicho Wrapper a **Denodo Virtual Data Port** creando la vista **base_itp_linkedin.**

4.4.1.6. LINKEDIN EMPRESAS

Al igual que en el apartado anterior, para la búsqueda de empresas se utiliza Linkedin.

1. Información general.

   - Nombre de la fuente: Linkedin y Google
   - URL inicial: **http://www.google.com/**,introduciendo **NAME** con la palabra que se quiere buscar.
   - Parámetros de entrada:
     - NAME: Nombre de la empresa que se quiere buscar.
   - Descripción de flujo de trabajo:

     a) Se inserta en el navegador la URL anteriormente mostrada con los valores de los parámetros de entrada.

     b) Se filtran los resultados, obteniendo solo las páginas que poseen información de empresas. Para ello se observa que en el título del resultado obtenido si contiene la cadena *Company Profile*.

Figura 4.18: Resultado de búsqueda en Google para Linkedin

c) A continuación, pinchando en cada perfil de empresa, se obtiene la página mostrada en la figura 4.19

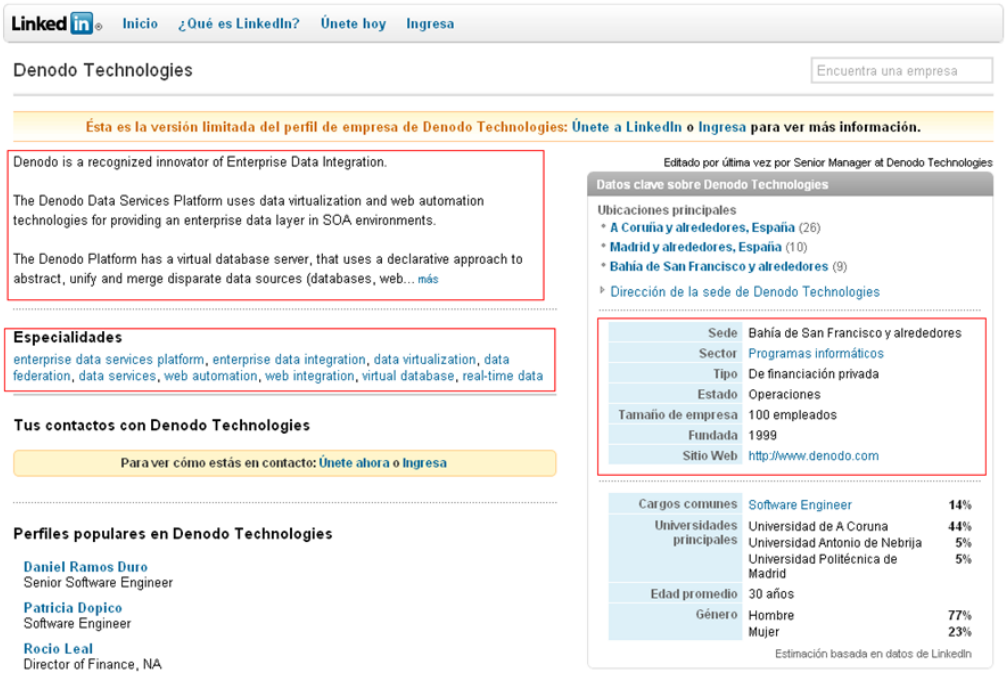


Figura 4.19: Página de detalle de empresas en Linkedin

- Paginación:
  - En el listado obtenido en Google se realiza un clic en cada resultado encontrado que corresponda con el perfil de una empresa.
- Datos de Salida: De la pantalla de la Figura 4.19 se obtienen los siguientes datos, para cada resultado encontrado:
  - **SUMMARY**: Resumen que la empresa ofrece de sí misma en el perfil de Linkedin.
  - **NAME:** Nombre obtenido del perfil de Linkedin
  - **SEDE:** Sede principal
  - **SECTOR:** Sector de la empresa
  - **SIZE:** Número de trabajadores
  - **WEB:** Web publicada en su perfil de Linkedin.
  - **SPECIALTIES:** Especialidades de la empresa.

2. Información de Ejecución

Este Wrapper se ejecuta en tiempo real. Para realizar su ejecución se importa dicho Wrapper a **Denodo Virtual Data Port** creando al vista **base_itp_lcompany**.

4.4.1.7. SONICO

1. Información general.

   - Nombre de la fuente: **Sonico**
   - URL inicial: **http://www.sonico.com**
   - Parámetros de entrada:
     - **FIRSTNAME**: Nombre de la persona que se quiere buscar.
     - **LASTNAME**: Apellido o apellidos de la persona que se quiere buscar.
     - **SOURCE**: Parámetro que dice si hay o no que buscar en Sonico.

     Ambos valores se concatenan y se insertan en el campo *Buscar*, tal como se muestra en la Figura 4.20.
   - Descripción de flujo de trabajo:

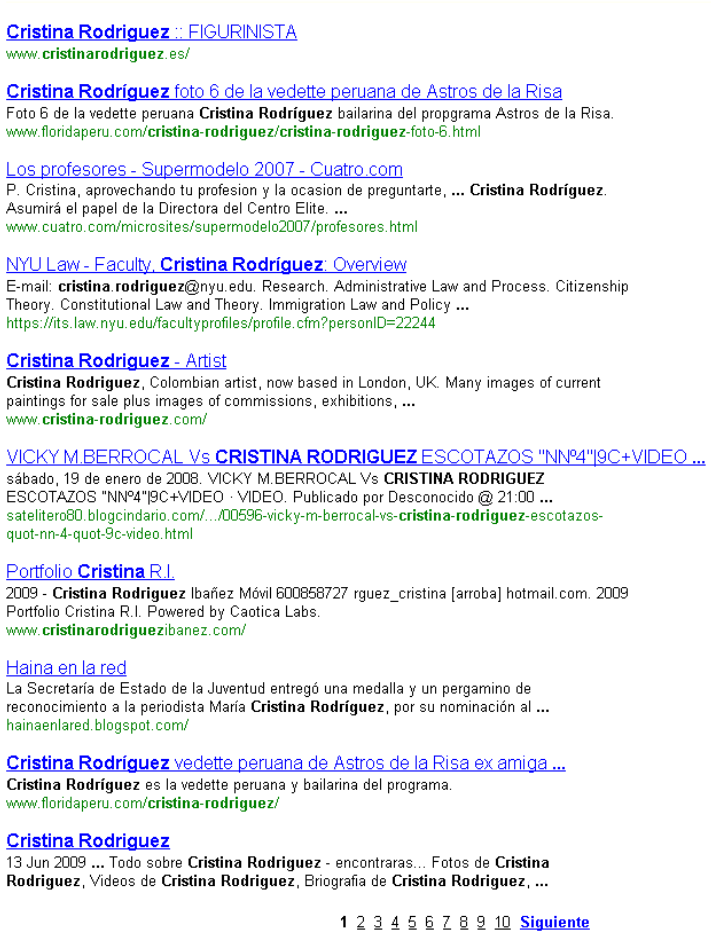
     a) Se ejecuta la URL anteriormente mostrada, y lleva a la página de la Figura 4.20.

     b) Se inserta en el campo mostrado en la Figura 4.20, campo *Buscar*, el nombre completo.

     c) Se pulsa el botón *Buscar*

Figura 4.20: Sonico

Una vez realiza la búsqueda se tiene como resultado la página mostrada en la Figura 4.21:



Cristina Rodriguez :: FIGURINISTA
www.**cristinarodriguez**.es/

Cristina Rodríguez foto 6 de la vedette peruana de Astros de la Risa
Foto 6 de la vedette peruana **Cristina Rodríguez** bailarina del propgrama Astros de la Risa.
www.floridaperu.com/**cristina-rodriguez**/**cristina-rodriguez**-foto-6.html

Los profesores - Supermodelo 2007 - Cuatro.com
P. Cristina, aprovechando tu profesion y la ocasion de preguntarte, **... Cristina Rodríguez**. Asumirá el papel de la Directora del Centro Elite. **...**
www.cuatro.com/microsites/supermodelo2007/profesores.html

NYU Law - Faculty, **Cristina Rodríguez**: Overview
E-mail: **cristina.rodriguez**@nyu.edu. Research. Administrative Law and Process. Citizenship Theory. Constitutional Law and Theory. Immigration Law and Policy **...**
https://its.law.nyu.edu/facultyprofiles/profile.cfm?personID=22244

Cristina Rodriguez - Artist
Cristina Rodriguez, Colombian artist, now based in London, UK. Many images of current paintings for sale plus images of commissions, exhibitions, **...**
www.**cristina-rodriguez**.com/

VICKY M.BERROCAL Vs **CRISTINA RODRIGUEZ** ESCOTAZOS "NNº4"|9C+VIDEO **...**
sábado, 19 de enero de 2008. VICKY M.BERROCAL Vs **CRISTINA RODRIGUEZ** ESCOTAZOS "NNº4"|9C+VIDEO · VIDEO. Publicado por Desconocido @ 21:00 **...**
satelitero80.blogcindario.com/.../00596-vicky-m-berrocal-vs-**cristina-rodriguez**-escotazos-quot-nn-4-quot-9c-video.html

Portfolio **Cristina** R.I.
2009 - **Cristina Rodriguez** Ibañez Móvil 600858727 rguez_cristina [arroba] hotmail.com. 2009 Portfolio Cristina R.I. Powered by Caotica Labs.
www.**cristinarodriguez**ibanez.com/

Haina en la red
La Secretaría de Estado de la Juventud entregó una medalla y un pergamino de reconocimiento a la periodista María **Cristina Rodríguez**, por su nominación al **...**
hainaenlared.blogspot.com/

Cristina Rodríguez vedette peruana de Astros de la Risa ex amiga **...**
Cristina Rodríguez es la vedette peruana y bailarina del programa.
www.floridaperu.com/**cristina-rodriguez**/

Cristina Rodriguez
13 Jun 2009 **...** Todo sobre **Cristina Rodriguez** - encontraras... Fotos de **Cristina Rodriguez**, Videos de **Cristina Rodriguez**, Briografia de **Cristina Rodriguez**, **...**

1 2 3 4 5 6 7 8 9 10 **Siguiente**

Figura 4.21: Búsqueda Sonico

d) Como la búsqueda mostrada Figura 4.21 es realizada en Google, se aplica un filtro en dicha búsqueda, obteniendo únicamente los resultados de Sonico.

- Paginación: En el listado de búsqueda obtenido con los resultados de Sonico, se realiza un clic en cada persona encontrada, como resultado se realiza la redirección a la página de la Figura 4.22:

Figura 4.22: Resultado Sonico I

- Datos de Salida: De la pantalla de la Figura 4.22 se obtienen los datos siguientes, para cada resultado encontrado:
  - SO_IMG_URL: URL de la imagen de Sonico.
  - SO_DETAIL_URL: URL de la página donde se obtienen los datos.
  - SO_FULLNAME: Nombre completo del usuario encontrado. A continuación, en la Figura 4.23, se muestra donde se encuentra dicho valor, en este caso corresponde al primer valor encuadrado:

Figura 4.23: Resultado Sonico II

  - SO_COUNTRY: País del usuario. Dicho valor está encuadrado en segundo lugar en la Figura 4.23.

- SO_FRIEND_LIST: Lista de amigos. En la parte inferior, debajo de la etiqueta *Amigos de Cristina Rodríguez Arroyo* se encuentran los amigos.

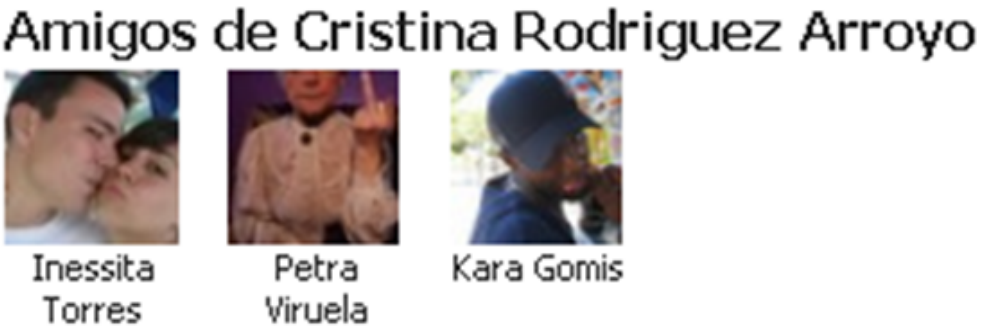


Figura 4.24: Resultado Sonico III

- Información de Ejecución

2. Método de ejecución:

   Este Wrapper se ejecuta en tiempo real. Para realizar su ejecución se importa dicho Wrapper a **Denodo Virtual Data Port** creando al vista base_itp_sonico.

4.4.1.8. PÁGINAS AMARILLAS

1. Información general.

   - Nombre de la fuente: **Páginas amarillas**.
   - URL inicial: **http://www.paginasamarillas.es/**, introduciendo **NAME** el nombre que se quiere buscar, **FINAL_PROVINCE** la provincia de búsqueda, **FINAL_CITY** la ciudad donde se quiere buscar y **FINAL_ACTIVITY** la actividad dentro de la Web donde se quiere buscar.
   - Parámetros de entrada:
     - **NAME**: Nombre de la empresa que se quiere buscar.
     - **PROVINCE**: Provincia de búsqueda.
     - **CITY**: Ciudad de búsqueda.
     - **ACTIVITY**: Actividad de la empresa que se está buscando.

     Todos estos parámetros se insertan en la URL anteriormente mostrada. Los anteriores parámetros son opcionales, exceptuando el nombre que es obligatorio.

- Descripción de flujo de trabajo:

  a) En el inicio del Wrapper se comprueba si los campos opcionales son nulos. Si es así se cambia el valor null por un String vacío para poder ser insertado en la URL.

  b) Se ejecuta la URL anteriormente mostrada con los parámetros de entrada y se muestra un listado como el de la Figura 4.25.

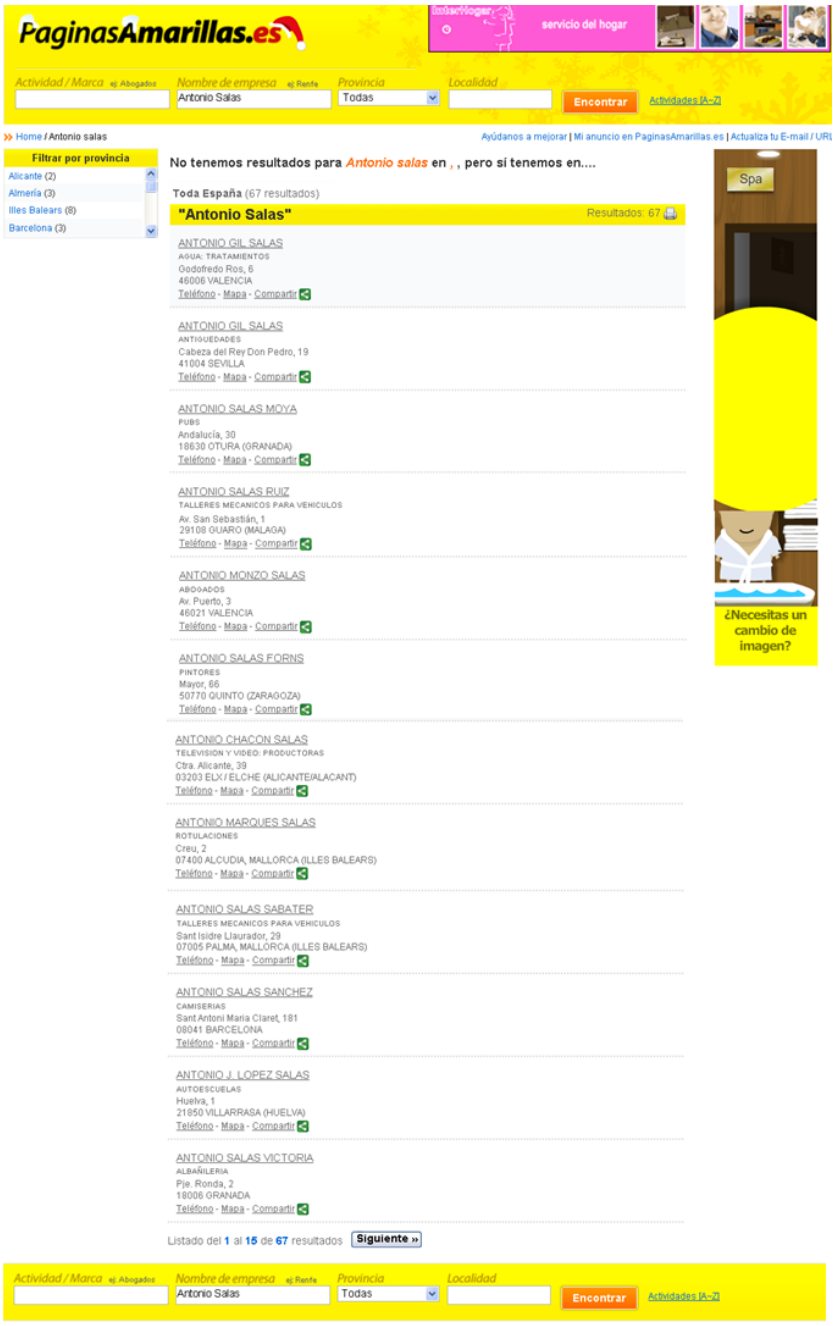


Figura 4.25: Listado Páginas amarillas

- Paginación:
  - En la página donde se encuentran los resultados, en la parte inferior, se encuentran varias páginas de resultados, sobre las que el Wrapper realizará un máximo de tres iteraciones, recorriendo las páginas.
  - En el listado de resultados mostrado en la Figura 4.25 se realiza un clic en cada resultado encontrado, obteniendo una pantalla de detalles como por ejemplo la mostrada en la Figura 4.26:

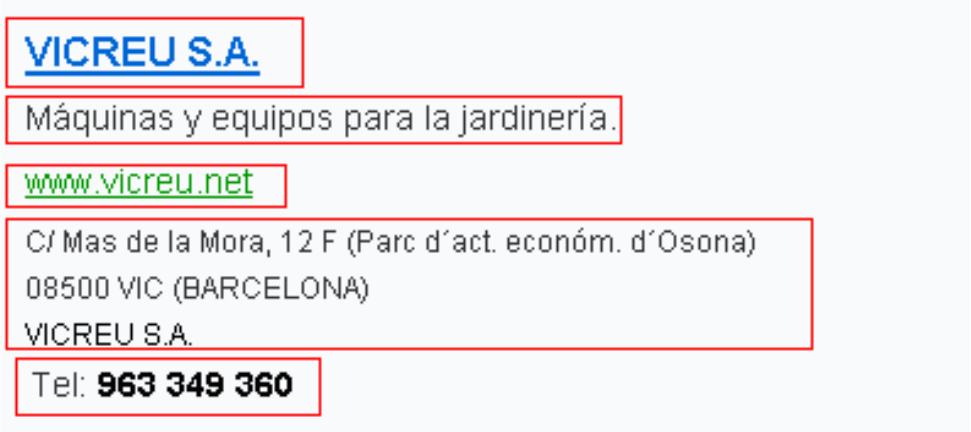


Figura 4.26: Pantalla de detalles páginas amarillas

- Datos de Salida: De la pantalla de la Figura 4.26 se obtienen los datos siguientes, para cada uno de los resultados encontrados:
  - **PA_FULLNAME**: Nombre completo del anunciante. En la Figura 4.26 se observa el valor de este campo encuadrado en primer lugar.
  - **PA_DESCRIPTION**: Descripción de la actividad que realiza. En la Figura 4.26 se observa el valor de este campo encuadrado en segundo lugar.
  - **PA_WEBSITE**: Página Web personal. En la Figura 4.26 se observa el valor de este campo encuadrado en tercer lugar.
  - **PA_ADDRESS**: Dirección de la empresa. En la Figura 4.26 se observa el valor de este campo encuadrado en cuarto lugar.
  - **PA_PHONES_LIST**: Listado telefónico de la empresa. En la Figura 4.26 se observa el valor de este campo encuadrado en quinto lugar.

2. Información de Ejecución

Este Wrapper se ejecuta en tiempo real. Para realizar su ejecución se importa dicho Wrapper a **Denodo Virtual Data Port** creando al vista **base_itp_pamarillas.**

4.4.2. Vistas bases

A continuación se va a hacer un repaso de las vistas base que se tienen en VDP. Estas vistas base están formadas por las fuentes analizadas en el apartado anterior y otras fuentes, como bases de datos y Webservice. Una vez se han importado todas las fuentes en VDP, es posible realizar una consultar fácilmente a cada una de ella como si se estuviera consultando una base de datos normal.

4.4.2.1. Base_itp_google

La vista base base_itp_google está creada a través del Wrapper de ITP **itp_google** explicado en el apartado anterior.

Los campos de salida de la vista:

BASE_ITP_GOOGLE	
G_URL	text
G_SUMMARY	text
G_SOURCE	text
G_TITLE	text
KEYWORDS	text
SOURCES	text

Figura 4.27: Campos vista base_itp_google

4.4.2.2. Base_itp_google_news

La vista base base_itp_google_news está creada a través del Wrapper de ITP **itp_googlenews** explicado en el apartado anterior.

Los campos de salida de la vista:

BASE_ITP_GOOGLE_NEWS	
G_SOURCE	text
G_TITLE	text
G_URL	text
G_SUMMARY	text
KEYWORDS	text
SOURCES	text

Figura 4.28: Campos vista base_itp_google

4.4.2.3. Base_itp_google_blog

La vista base base_itp_google_blog está creada a través del Wrapper de ITP **itp_googleblog** explicado en el apartado anterior.

Los campos de salida de la vista:

BASE_ITP_GOOGLE_BLOG	
G_URL	text
G_SUMMARY	text
G_TITLE	text
G_SOURCE	text
KEYWORDS	text
SOURCES	text

Figura 4.29: Campos vista base_itp_google_blog

4.4.2.4. Base_itp_facebook

La vista base base_itp_facebook está creada a través del Wrapper de ITP **itp_facebook** explicado en el apartado anterior.

Los campos de salida la vista son:

BASE_ITP_FACEBOOK	
FB_COUNTRY	text
FB_DETAIL_URL	text
FB_NAME_DETAILS	text
FB_NICK	text
FB_REDES	text
FB_NAME	text
FB_IMAGE_URL	text
FB_TYPE	text
FB_FRIENDS	itp_facebook_fb_friends
FB_FRIEND_NAME	text
FIRSTNAME	text
SOURCE	text
LASTNAME	text

Figura 4.30: Campos vista base_itp_facebook

4.4.2.5. Base_itp_facebook_google

La vista base base_itp_facebook_google está creada a través del Wrapper de ITP **itp_facebook_google**, explicado en el apartado anterior.

Los campos de salida la vista son:

BASE_ITP_FACEBOOK_GOOGLE	
FB_COUNTRY	text
FB_NAME_DETAILS	text
FB_NICK	text
FB_IMAGE_URL	text
FB_NAME	text
FB_FRIENDS	itp_facebook_fb_friends
FIRSTNAME	text
LASTNAME	text
SOURCE	text

Figura 4.31: Campos vista base_itp_facebook_google

4.4.2.6. Base_itp_linkedin

La vista base base_itp_linkedin está creada a través del Wrapper de ITP **itp_linkedin**, explicado en el apartado anterior.

Los campos de salida de la vista son:

BASE_ITP_LINKEDIN	
LI_SUMMARY	text
LI_CURRENT_JOB	text
LI_CURRENT_COMPANY	text
LI_COUNTRY	text
LI_INDUSTRY	text
LI_JOB	text
LI_TYPE	text
LI_UNIVERSITY	text
LI_FULLNAME	text
LI_DETAIL_URL	text
LI_IMAGE_URL	text
LI_NICK	text
LI_EXP_DATA	itp_linkedin_li_exp_data
LI_JOB_NAME	text
LI_COMPANY_NAME	text
LI_JOB_INIT_DATE	text
LI_JOB_END_DATE	text
LI_JOB_DESCRIPTION	text
LI_PAST_JOB_LIST	itp_linkedin_li_past_job_list
LI_PAST_JOB	text
LI_PAST_COMPANY	text
FIRSTNAME	text
SOURCE	text
LASTNAME	text

Figura 4.32: Campos vista base_itp_linkedin

4.4.2.7. Base_itp_lcompany

La vista base base_itp_lcompany está creada a través del Wrapper de ITP **itp_linkedincompany**, explicado en el apartado anterior.

Los campos de salida de dicha vista son:

BASE_ITP_LCOMPANY	
SUMMARY	text
NAME	text
SEDE	text
SECTOR	text
SIZE	text
WEB	text
SPECIALTIES	text
IMG	text
COMPANY	text

Figura 4.33: Campos vista base_itp_lcompany

4.4.2.8. Base_itp_sonico

La vista base base_itp_sonico está creada a través del Wrapper de ITP **itp_sonico**, explicados en el apartado anterior.

Los campos de salida de dicha vista son:

BASE_ITP_SONICO	
SO_IMG_URL	text
SO_DETAIL_URL	text
SO_FULLNAME	text
SO_COUNTRY	text
SO_TYPE	text
SO_FRIEND_LIST	itp_sonico_so_friend_list
FRIEND_NAME	text
FIRSTNAME	text
SOURCE	text
LASTNAME	text

Figura 4.34: Campos vista base_itp_sonico

4.4.2.9. Base_itp_pamarillas

La vista base base_itp_pamarillas está creada a través del Wrapper de ITP **itp_pamarillas**, explicados en el apartado anterior.

Los campos de salida de la vista son:

BASE_ITP_PAMARILLAS	
PA_FULLNAME	text
PA_DESCRIPTION	text
PA_WEBSITE	text
PA_ADDRESS	text
PA_PHONES_LIST	itp_pamarillas_pa_phon...
NAME	text
ACTIVITY	text
PROVINCE	text
CITY	text

Figura 4.35: Campos de la vista base_itp_pamarillas

4.4.2.10. Base_xml_provinces

La vista base base_xml_provinces está creada a través del WebService de páginas amarillas:

http://callejero.paginasamarillas.es/services/localidades/getLocalidades.asmx/-LocalityServer?param1=(parametro_entrada)

Los campos de salida de dicha vista son:

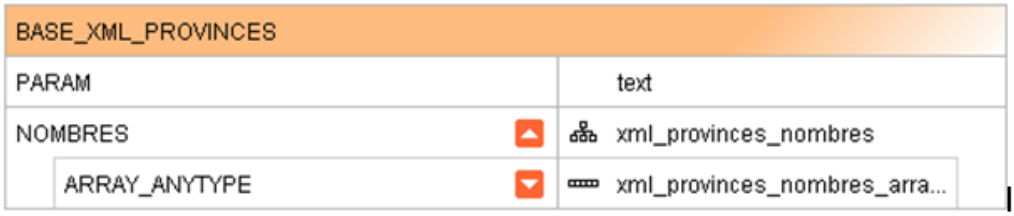


Figura 4.36: Campos vista base_xml_provinces

4.4.2.11. Base_csv_task_input

La vista base **base_csv_task_input** proviene del fichero csv **input_data**. En este fichero se almacena la información obtenida del fichero de carga.

Los campos de salida de dicha vista son:

BASE_CSV_TASK_INPUT	
NIF	text
CUENTA	text
DIREC	text
PLAZA	text
TELEF	text
NOMBRE	text
APEL1	text
APEL2	text
TIPO	text
FILENAME	text

Figura 4.37: Campos vista base_csv_task_input

4.4.2.12. Base_person

La vista base **base_person** proviene de la base de datos local, de la tabla **person**. Esta vista se emplea en la búsqueda realizada a través de la cargar de un fichero a través de la aplicación. En la tabla de la que proviene esta vista se almacena la información proveniente del fichero de carga para realizar la búsqueda y se almacena el estado del resultado de la misma.

Los campos de salida de dicha vista son:

BASE_PERSON	
ID_PERSON	long
ID_TASK	text
NAME	text
FOUND	text
NIF	text

Figura 4.38: Campos vista base_person

4.4.2.13. Base_person_result

La vista base **base_person_result** proviene de la base de datos local, de la tabla **person_result**. Esta vista se emplea en la búsqueda a través de la carga de un fichero a través de la aplicación. En la tabla de la que proviene esta vista se almacena el resultado de la búsqueda de redes sociales a partir del fichero de carga.

Los campos de salida de dicha vista son:

BASE_PERSON_RESULT	
ID_PERSON_RESULT	long
ID_PERSON_RESULT_PERSON	long
ID_TASK	text
RELIABILITY	int
SOCIAL_PHOTO	text
FULL_NAME	text
SOCIAL_NICK	text
SOCIAL_PROFILE	text
COUNTRY	text
SOCIAL_FRIENDS	text
ADDRESS	text
PHONE	text
OTHER_INFO	text
UNIVERSITY	text
COMPANY_JOB	text
COMPANY_STD	text
COMPANY_PHONE	text
COMPANY_ADDRESS	text

Figura 4.39: Campos vista base_person_result

4.4.2.14. Base_csv_nickname

La vista base **base_csv_nickname** proviene de un fichero csv donde se contienen los nicknames asociados a un nombre.

Los campos de salida de dicha vista son:

BASE_CSV_NICKNAME	
name	text
nicknames	text

Figura 4.40: Campos vista base_csv_nickname

La llamada a esta vista se realiza a través del procedimiento almacenado, cuando se componen las querys de consulta de cada fichero de configuración.

4.4.3. Vistas Derivadas finales

En el capítulo anterior, se muestran las vistas base y sus campos. Estas vistas serían las vistas generadas directamente a partir de las fuentes. A continuación se va a realizar un repaso por las vistas que finalmente son las que utiliza la aplicación. Para obtener dichas vistas, ha sido necesario realizar diferentes combinaciones entre tablas, estas combinaciones se podrán ver en los *árboles* que se muestran para cada tabla. En el **apéndice 4** se tiene una explicación detallada de cada una de las tablas intermedias para más información.

En cada explicación de vista derivada se mostrará la pantalla **Tree View** que proporciona la herramienta de gestión de VDP. Este modo de visualización muestra de forma gráfica los sucesivos niveles de vistas que se han ido componiendo para definir la vista. Haciendo clic sobre cualquiera de las vistas del árbol, se accederá a la página que muestra su esquema. Haciendo clic sobre los nodos del árbol que representan operaciones de combinación (joins, uniones, selecciones...) se visualizarán sus principales propiedades. Por ejemplo, en el caso de las operaciones de selección se mostrará la condición utilizada para crearla. A continuación, en la figura 4.41 se muestra un resumen del significado de los símbolos que se pueden ver en las diferentes figuras de Tree View mostradas.

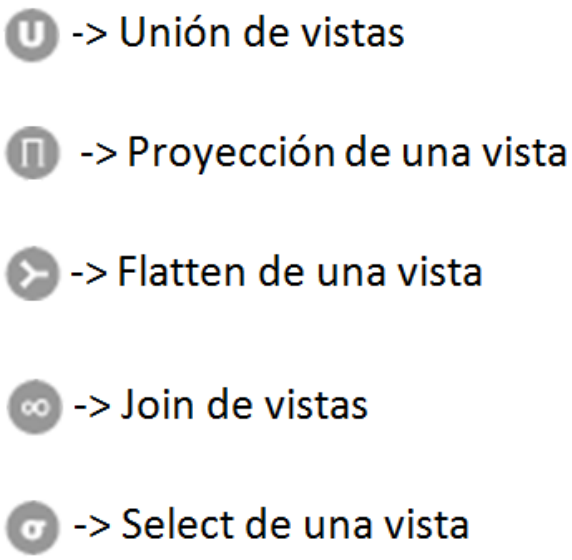


Figura 4.41: Símbolos Tree view

4.4.3.1. Final_google_by_keywords

La vista derivada final_google_by_keywords es una proyección de la vista base **view_google**, como muestra la siguiente Figura 4.42:

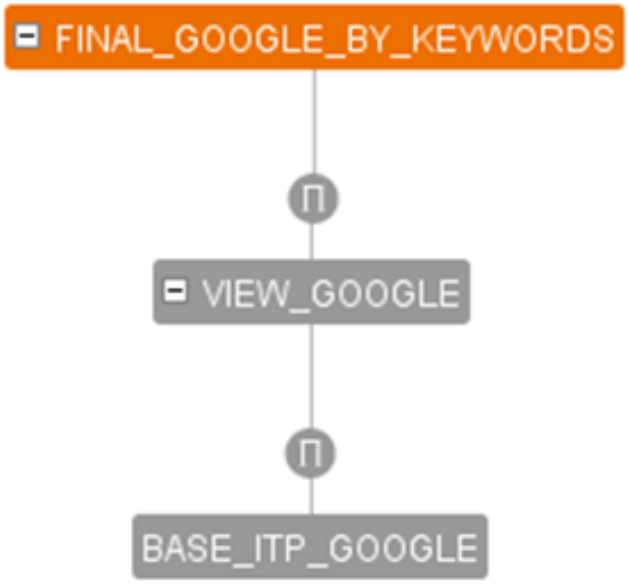


Figura 4.42: Tree view final_google_by_keywords

Esta vista mostrará la búsqueda final de Google. La llamada a esta vista se realizará a través de la aplicación Web directamente.

Los campos de dicha vista son:

FINAL_GOOGLE_BY_KEYWORDS	
ST_ACTIVITY_MESSAGE	text
ST_OTHER_INFO	text
ST_ACTIVITY_URL	text
ST_ACTIVITY_TITLE	text
KEYWORDS	text
SOURCES	text

Figura 4.43: Campos vista final_google_by_keywords

A continuación se explican más detalladamente los campos de la Figura 4.43:

- **ST_ACTIVITY_MESSAGE**: Este campos contiene el resumen de los resultados de la búsqueda en Google.
- **ST_OTHER_INFO**: Este campo contiene otra información adicional obtenida en los resultados de en la búsqueda.
- **ST_ACTIVITY_URL**: Este campo contiene la URL que contiene el resultado de la búsqueda.
- **ST_ACTIVITY_TITLE**: Este campo contiene el título que recibe cada resultado de la búsqueda en Google.
- **KEYWORDS**: Palabra clave con la que se ha realizado la búsqueda.
- **SOURCES**: Campo que indica si se realiza la búsqueda en Web o no.

4.4.3.2. Final_blogs_by_keywords

La vista derivada final_blogs_by_keywords es una proyección de la vista base **base_itp_google_blog**, como muestra la siguiente Figura 4.44:

Figura 4.44: Tree view final_blogs_by_keywords

Esta vista mostrará la búsqueda final de Google en blogs. La llamada a esta vista se realizará a través del procedimiento almacenado cuando el fichero de configuración pasado como parámetro contiene las querys a dicha vista.

Los campos de dicha vista son:

FINAL_BLOGS_BY_KEYWORDS	
ST_ACTIVITY_URL	text
ST_ACTIVITY_MESSAGE	text
ST_ACTIVITY_TITLE	text
ST_OTHER_INFO	text
KEYWORDS	text
SOURCES	text

Figura 4.45: Campos vista final_google_by_keywords

A continuación se explican más detalladamente los campos de la Figura 4.45:

- **ST_ACTIVITY_MESSAGE**: Este campos contiene el resumen de cada resultado de la búsqueda en Google Blogs.
- **ST_OTHER_INFO**: Este campo contiene otra información adicional obtenida en cada resultado.
- **ST_ACTIVITY_URL**: Este campo contiene la URL que contiene cada resultado.
- **ST_ACTIVITY_TITLE**: Este campo contiene el título que recibe cada resultado de la búsqueda en Google Blogs.
- **KEYWORDS**: Palabra clave con la que se ha realizado la búsqueda.
- **SOURCES**: Campo que indica si se realiza la búsqueda en blogs o no.

4.4.3.3. Final_news_by_keywords

La vista derivada final_news_by_keywords es una proyección de la vista base **base_itp_google_news**, como muestra la siguiente Figura 4.46:

Figura 4.46: Tree view final_news_by_keywords

Esta vista mostrará la búsqueda final de Google en noticias. La llamada a esta vista se realizará a través del procedimiento almacenado cuando el fichero de configuración pasado como parámetro contiene las querys a dicha vista.

Los campos de dicha vista son:

FINAL_NEWS_BY_KEYWORDS	
ST_OTHER_INFO	text
ST_ACTIVITY_TITLE	text
ST_ACTIVITY_URL	text
ST_ACTIVITY_MESSAGE	text
KEYWORDS	text
SOURCES	text

Figura 4.47: Campos vista final_news_by_keywords

A continuación se explican más detalladamente los campos de la Figura 4.47:

- **ST_ACTIVITY_MESSAGE**: Este campos contiene el resumen de cada resultado de la búsqueda en Google News.
- **ST_OTHER_INFO**: Este campo contiene otra información adicional obtenida en cada resultado.

- **ST_ACTIVITY_URL**: Este campo contiene la URL que contiene cada resultado.
- **ST_ACTIVITY_TITLE**: Este campo contiene el título que recibe cada resultado de la búsqueda en Google News.
- **KEYWORDS**: Palabra clave con la que se ha realizado la búsqueda.
- **SOURCES**: Campo que indica si se realiza la búsqueda en noticias o no.

4.4.3.4. Final_social_by_keywords

La vista derivada final_social_by_keywords es una unión de las vistas base **base_itp_linkedin**, **view_facebook** y **base_itp_sonico**, como muestra la Figura 4.48:

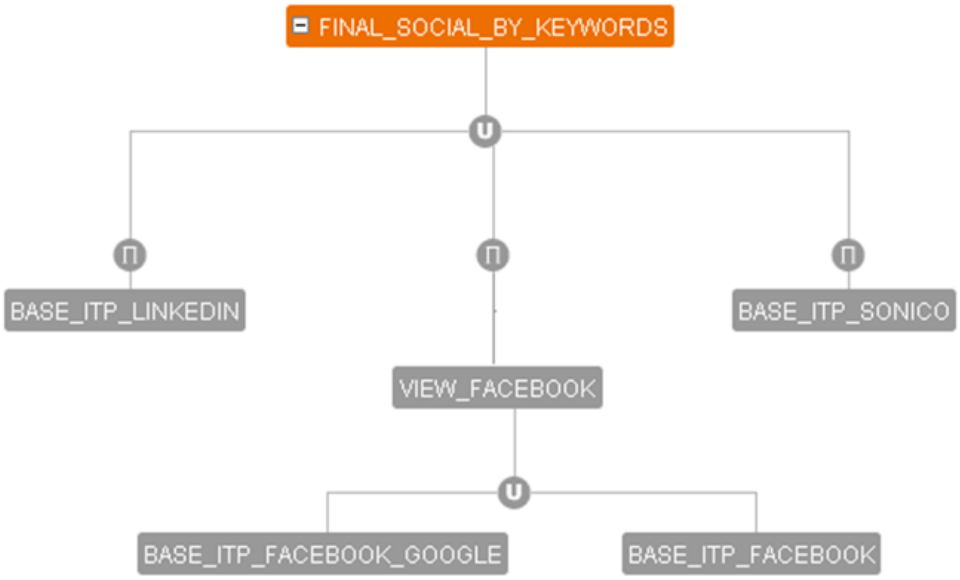


Figura 4.48: Tree view final_social_by_keywords

Se realiza una unión de las tres vistas base y se genera una nueva vista cuyo nombre de los campos coinciden con los que devuelve el procedimiento almacenado.

Esta vista devuelve el resultado final de la búsqueda en redes sociales. LLa llamada a esta vista se realizará a través del procedimiento almacenado cuando el fichero de configuración pasado como parámetro contiene las querys a dicha vista.

FINAL_SOCIAL_BY_KEYWORDS	
ST_STD_NAME	text
ST_FULL_NAME	text
ST_SOCIAL_NICK	text
ST_COUNTRY	text
ST_SOCIAL_PROFILE	text
ST_SOCIAL_DESCRIPTION	text
ST_SOCIAL_PHOTO	text
ST_COMPANY_STD	text
ST_COMPANY_JOB	text
ST_SOCIAL_INTEREST	text
ST_UNIVERSITY	text
FIRSTNAME	text
LASTNAME	text
ST_TYPE	text
ST_SOCIAL_FRIENDS	itp_facebook_fb_friends
FB_FRIEND_NAME	text

Figura 4.49: Campos vista final_social_by_keywords

A continuación se explican más detalladamente los campos de la 4.49:

- **ST_STD_NAME**: Nombre de la persona.
- **ST_FULL_NAME**: Nombre completo de la persona, es decir, nombre extraído de la red social.
- **ST_SOCIAL_NICK**: Nick que emplea en la red social.
- **ST_COUNTRY**: País de la persona.
- **ST_SOCIAL_PROFILE**: URL del perfil de la persona, es la URL de donde se han obtenido los actuales valores.
- **ST_SOCIAL_DESCRIPTION**: Descripción de la persona en la red social.
- **ST_SOCIAL_PHOTO**: URL de la foto que la persona encontrada ha colgado en la red social.

- **ST_COMPANY_STD**: Compañía en la que trabaja.

- **ST_COMPANY_JOB**: Trabajo que desempeña en la compañía indicada en el campo anterior.

- **ST_SOCIAL_INTEREST**: Intereses de la persona.

- **ST_UNIVERSITY**: Universidad de estudios

- **FIRSTNAME**: Nombre de la persona.

- **LASTNAME**: Apellidos o apellido de la persona encontrada.

- **ST_SOCIAL_FRIENDS**: Lista de amigos que posee en la red social.

- **ST_TYPE**: Este parámetro indica donde se realiza la búsqueda, es decir, si en todas las fuentes, en varias o solo en una.

4.4.3.5. Final_company_by_company

La vista derivada final_company_by_company, es una unión entre las vistas **base_itp_lcompany** y **final_company_by_name**.

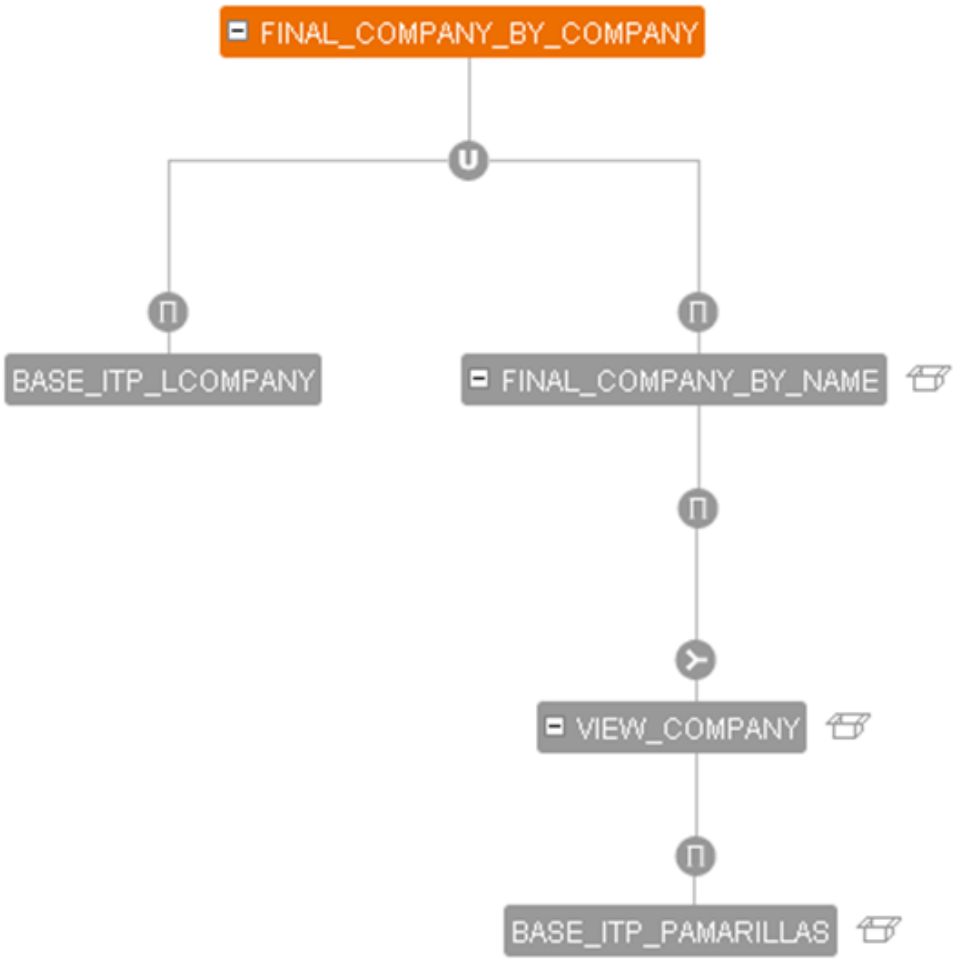


Figura 4.50: Treeview Final_company_by_name

Al igual que en todos los casos se modifica el nombre de los campos para que coincidan con el nombre de los campos que devuelve el procedimiento almacenado. La llamada a esta vista se realizará a través del procedimiento almacenado cuando el fichero de configuración pasado como parámetro contiene las querys a dicha vista.

Los campos de dicha vista son:

FINAL_COMPANY_BY_NAME	
ST_COMPANY	text
ST_COMPANY_DESCRIPTION	text
ST_COMPANY_WEB	text
ST_COMPANY_ADDRESS	text
ST_COMPANY_PHONE	text
NAME	text
ACTIVITY	text
PROVINCE	text
CITY	text

Figura 4.51: Campos de la vista final_company_by_name

A continuación se explican más detalladamente los campos de la 4.51:

- **ST_COMPANY**: Nombre de la empresa.
- **ST_COMPANY_DESCRIPTION**: Descripción de la empresa.
- **ST_COMPANY_WEB**: Página Web de la empresa.
- **ST_COMPANY_ADDRESS**: Dirección de la empresa.
- **ST_COMPANY_PHONE**: Teléfono de la empresa.
- **NAME**: Nombre de la empresa.
- **ACTIVITY**: Actividad a la que se dedica la empresa.
- **PROVINCE**: Provincia de la empresa.
- **CITY**: Ciudad de la empresa.

4.4.3.6. Final_csv_task_input_province

La vista derivada final_csv_task_input_province, es una proyección de la vista **inter_csv_task_input_province**, como muestra la Figura 4.52:

Figura 4.52: Treeview final_csv_task_input_province

Esta vista se emplea en la búsqueda a través de la carga de un fichero. El fichero de carga incluye la dirección de búsqueda de la persona, esta dirección puede no contener la provincia, por lo que se emplea esta vista para obtenerla. A través de la combinación entre la dirección obtenida del fichero y el resultado del Webservice, que emplea la vista base base_xml_province, se obtiene la provincia para poder realizar la búsqueda.

El fichero de carga tiene el siguiente formato:

dni;cuenta;direc;plaza;telef;nombre;apel1;apel2;tipo

Los campos de la Figura 4.53 son similares a los que se tiene en el formato del fichero anteriormente mostrado, obteniendo además la provincia.

Los campos de la nueva vista son:

FINAL_CSV_TASK_INPUT_PROVINCES	
FILENAME	text
PARAM	text
NIF	text
ACCOUNT	text
ADDRESS	text
PLAZA	text
PHONE	text
NAME	text
LASTNAME	text
TYPE	text
PROVINCE	text

Figura 4.53: Campos de la vista Final_csv_task_input_provinces

A continuación se explican más detalladamente los campos de la 4.53:

- **FILENAME**: Nombre del fichero de carga.
- **PARAM**: Parámetros insertados.
- **NIF**: Documento de identidad de la persona.
- **ACCOUNT**: Número de cuenta.
- **ADDRESS**: Dirección completa.
- **PLAZA**: Ciudad de la persona.
- **PHONE**: Teléfono de la persona.
- **NAME**: Nombre de la persona.
- **LASTNAME**: Apellido de la persona.
- **TYPE**: Tipo de búsqueda
- **PROVINCE**: Provincia obtenida a través de la dirección insertada en el fichero utilizando el Webservice de páginas amarillas.

4.5. Skiptracing con Denodo

En el apartado anterior, se ha realizado un análisis de las fuentes de extracción de información, y como por medio de la plataforma Denodo esta información se convertía en tablas de datos virtuales. Para entender mejor el mecanismo de transformación desde la fuente de información hasta la vista derivada final, en el presente capítulo se realizarán unos ejemplos de creación de un **Wrapper ITP** y su posterior importación a la **herramienta VDP**, así como el procesado hasta llegar a una vista base final.

4.5.1. Generación de Wrapper ITP

En este apartado se va a explicar cómo se extraen los datos de Google. Para ello, se debe arrancar la plataforma y ejecutar el **Wrapper Generator Tool**. Los pasos dentro de la herramienta son los siguientes:

1. Se crea un nuevo Wrapper llamado **itp_google**

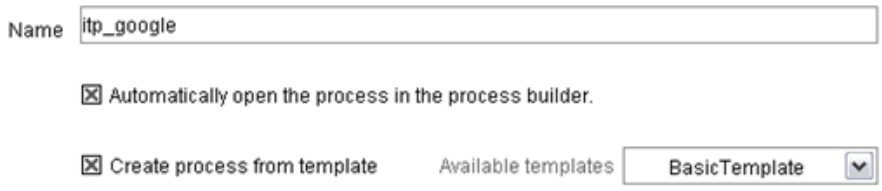


Figura 4.54: Pantalla inicial ITP

Una vez creado el Wrapper se accede a la pantalla 4.55, donde aparece una estructura básica de Wrapper que se irá modificando. La estructura que aparece en la figura 4.55, es una estructura que aparece por defecto siempre que se crea un nuevo Wrapper. Esta estructura contendrá los elementos básicos para que el Wrapper funcione, como son:

- Elementos de inicio y fin
- Elemento Sequence, para grabar la navegación.
- Elemento Extractor, para extraer la información.
- Elemento Iterator, para iterar sobre los resultados encontrados y extraídos.
- Elemento Record Constructor, para configurar los parámetros de salida.

- Elemento Output, para devolver los parámetros de salida.

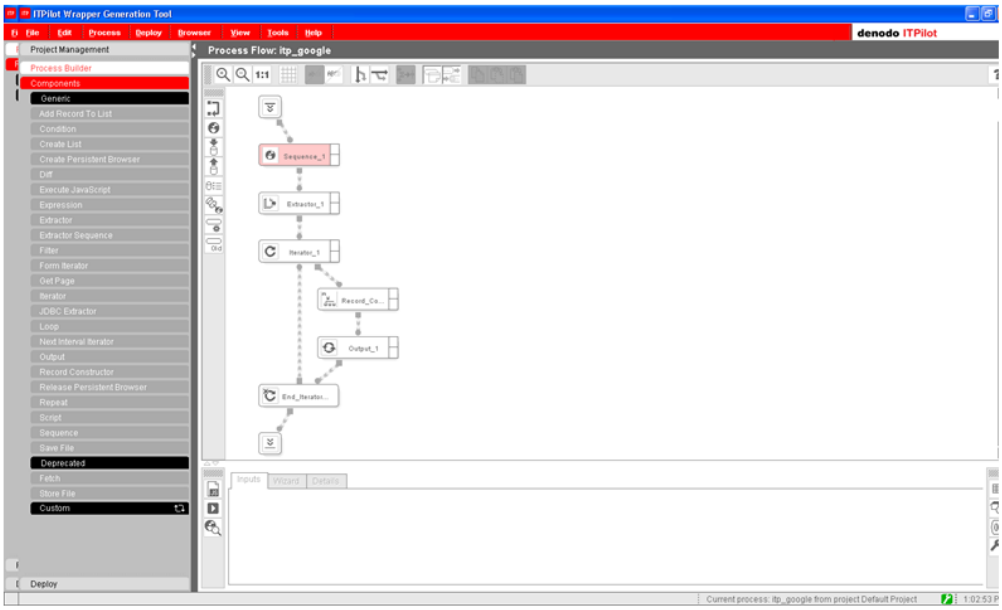


Figura 4.55: Pantalla principal ITP

2. Se añaden los parámetros de entrada que tendrá el Wrapper, pinchando en el elemento inicial:

Figura 4.56: Pantalla de definición de parámetros de entrada

En este caso será **KEYWORD** que será la palabra clave que se quiere buscar.

3. A continuación, se configura el elemento **Sequence**. Este elemento contiene el código necesario para automatizar la navegación. Dicha navegación se graba a través de la **barra de Denodo** (Denodo Toolbar), mostrada en la figura 4.57 y posteriormente se importa en el elemento Sequence. La barra de Denodo, contiene el botón **Rec**, que permite grabar los pasos necesarios para automatizar la navegación.

Figura 4.57: Pantalla de definición de parámetros de entrada

En este caso, la navegación comienza en www.google.com, y se inserta dentro del cuadro de texto del buscador el valor **KEYWORD**. Una vez insertado se pulsa el botón **Buscar**. Una vez se ha terminado de grabar se pulsa el botón **stop** de la barra de Denodo. Finalizando estos pasos se muestra la figura 4.58:

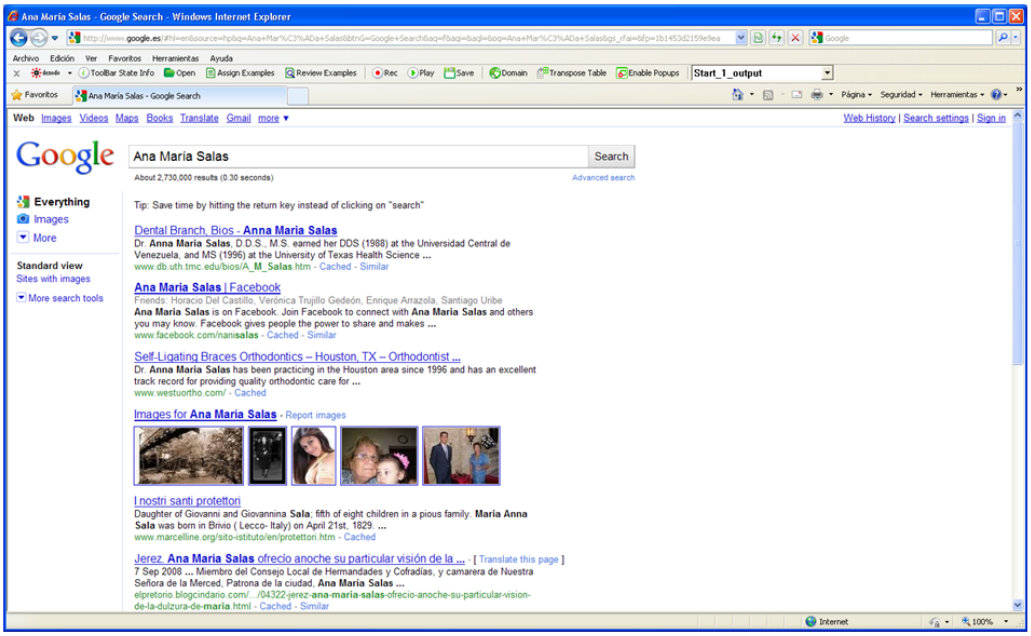


Figura 4.58: Pantalla de la fuente de extracción de información

Una vez se ha automatizado la navegación, se vuelve al elemento **Sequence** y se importan los datos a dicho elemento. Con esta acción se obtiene el código **NSEQL**[67] que automatiza la navegación. En la figura 4.59 y 4.60 se muestra dicho código:

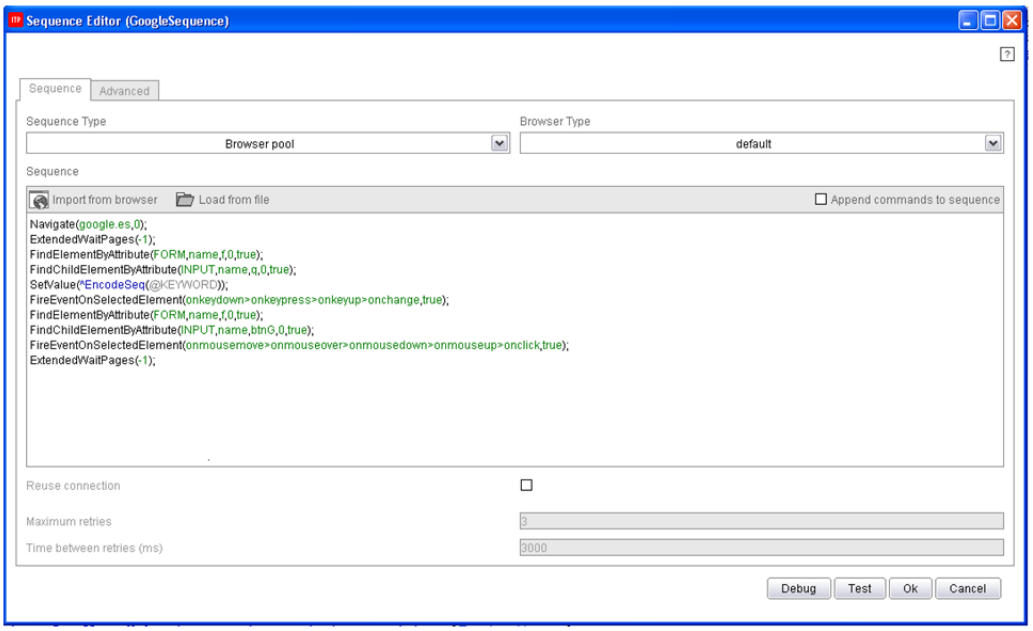


Figura 4.59: Código NSEQL del elemento Sequence

```
Navigate(google.es,0);
ExtendedWaitPages(-1);
FindElementByAttribute(FORM,name,f,0,true);
FindChildElementByAttribute(INPUT,name,q,0,true);
SetValue(^EncodeSeq(@KEYWORD));
FireEventOnSelectedElement(onkeydown>onkeypress>onkeyup>onchange,true);
FindElementByAttribute(FORM,name,f,0,true);
FindChildElementByAttribute(INPUT,name,btnG,0,true);
FireEventOnSelectedElement(onmousemove>onmouseover>onmousedown>onmouseup>onclick,true);
ExtendedWaitPages(-1);
```

Figura 4.60: Código NSEQL

4. Una vez se ha automatizado la navegación, se va a configurar el elemento **Next Interval Iterator**. Este elemento no aparece en la estructura inicial, por lo que será necesario añadirlo y colocarlo correctamente. A continuación, en la figura 4.61 se muestra como queda la estructura con este nuevo elemento:

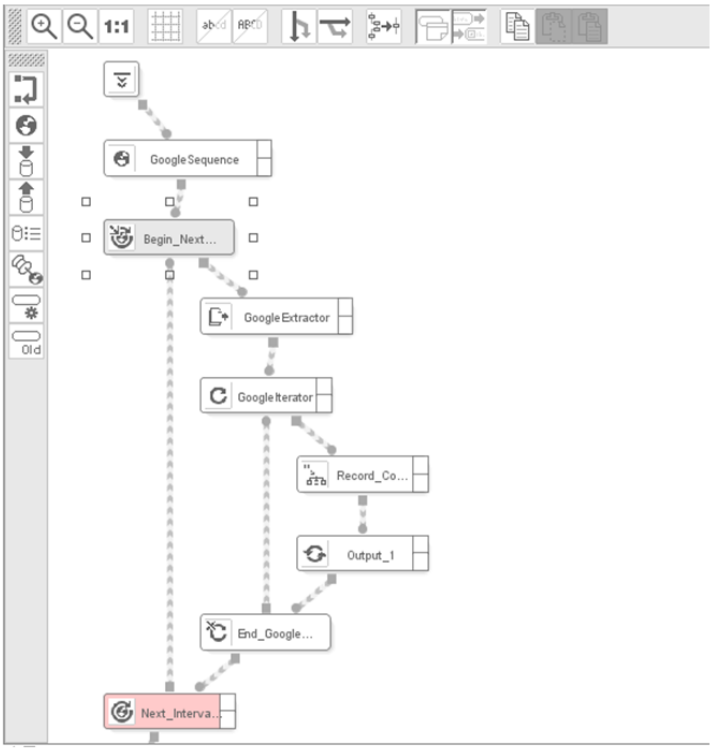


Figura 4.61: Wrapper ITP con nuevo elemento Next Interval Iterator

Para entender mejor cómo funciona el elemento **Next Interval Iterator**, se presenta la figura 4.62;

Figura 4.62: Iterador de páginas Google

Se observa que Google muestra un número de resultados por página y cada búsqueda puede generar varias páginas. El elemento **Next Interval Iterator** es capaz de iterar entre cada página. Para ello se le puede indicar que itere hasta que termine el número de páginas o solo realice un número predeterminado.

En este caso, se configura con 3 iteraciones ya que si se obtuvieran todos los resultados ralentizaría mucho la aplicación y habría mucha información irrelevante.

Para realizar esta automatización, se utiliza la barra de Denodo, al igual que se ha hecho en el elemento Sequence, grabando la secuencia e importándola al elemento, quedando como muestra la figura 4.63 y 4.64:

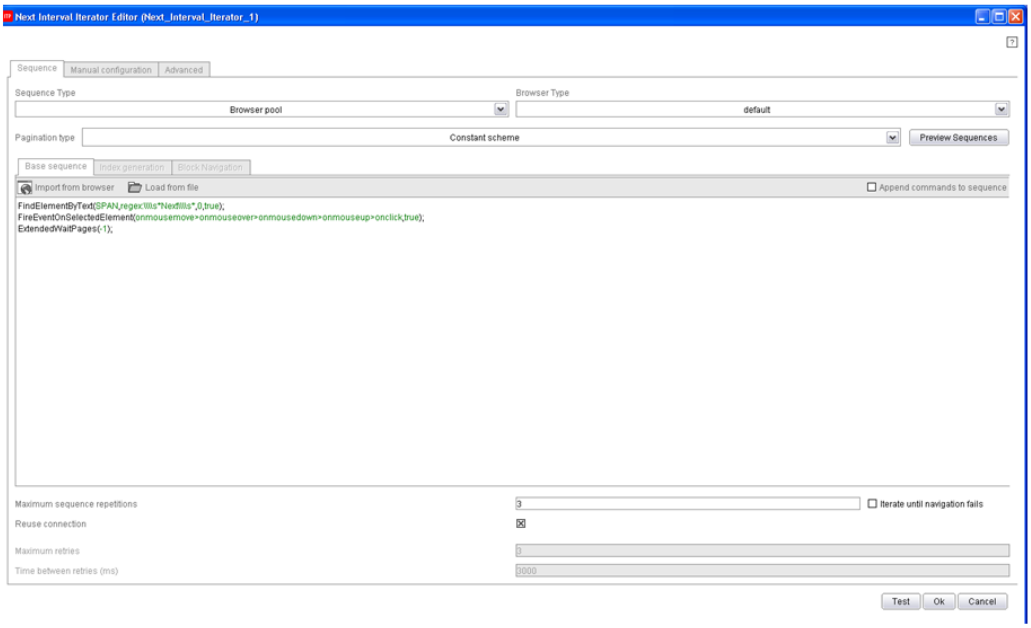


Figura 4.63: Iterador NSEQL del elemento Next Interval Iterator

```
FindElementByText(SPAN,regex:\\ls*Next\\ls*,0,true);
FireEventOnSelectedElement(onmousemove>onmouseover>onmousedown>onmouseup>onclick,true);
ExtendedWaitPages(-1);
```

Figura 4.64: Código NSEQL del elemento Next Interval Iterator

5. A continuación, se configura el elemento **Extractor**, en este elemento se define la estructura que se va a extraer y se obtendrá la información de la página Web obtenida en cada secuencia.

   Lo primero, se crea la estructura como se observa en la figura 4.65 y 4.66:

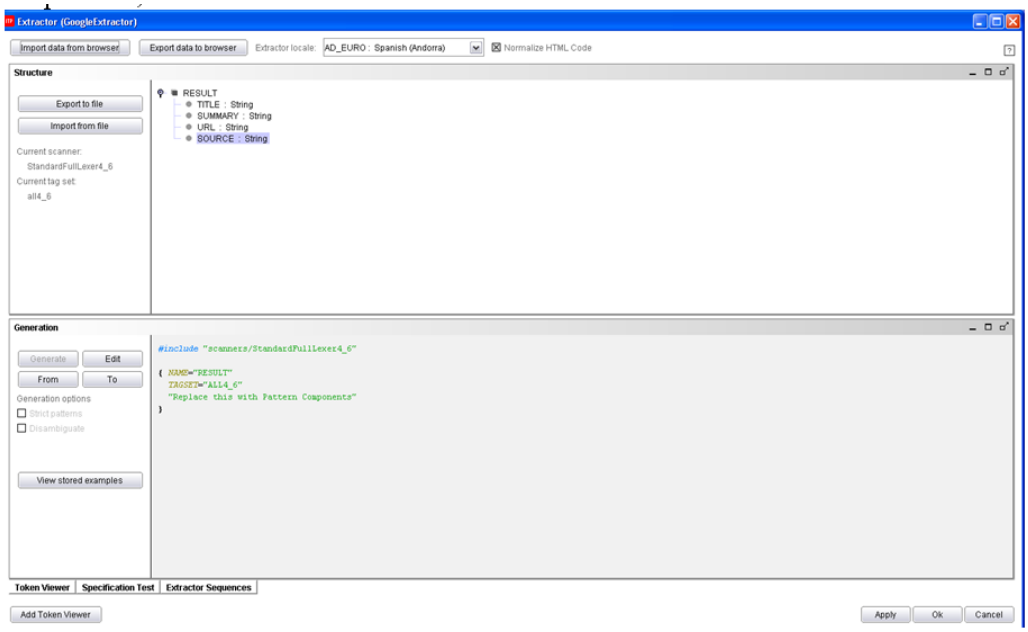


Figura 4.65: Estructura del Extractor

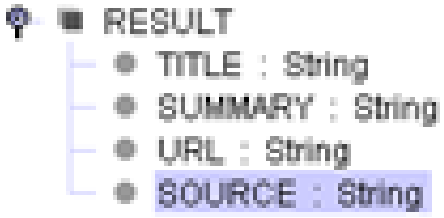


Figura 4.66: Estructura

La estructura generada se exporta al navegador y con la barra de Denodo se realiza la extracción de ejemplos, a continuación se muestra un ejemplo de extracción:

Figura 4.67: Asignación de ejemplos

Una vez se ha creado un número de ejemplos representativo para crear correctamente el código de extracción, se importa la información del browser o navegador. Esta acción desencadena la generación automática del código **DEXTL**, como se puede ver a continuación en la figura 4.68:

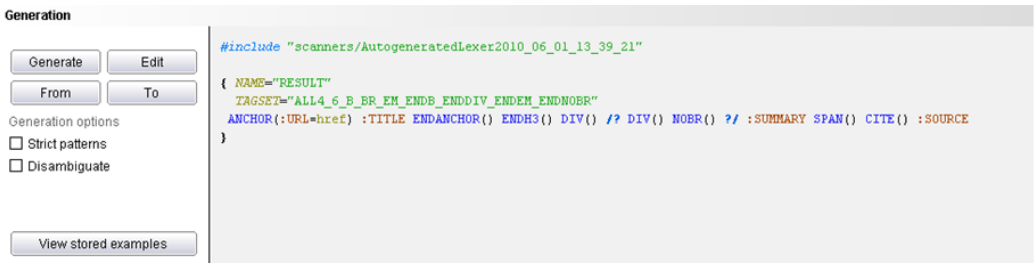


Figura 4.68: Código DEXTL para extracción de información

Este código se puede generar automáticamente, aunque hay veces que es complicada la generación automática y se tiene que escribir el código a mano. Para facilitar este trabajo, existe la pestaña **Token Viewer**. Esta pestaña muestra un editor que permite obtener los **token** de HTML de una parte seleccionada de la página que se seleccione en el navegador, así facilita la generación del código a mano.

6. Una vez se tienen los ejemplos y el extractor configurado, le toca el turno al Iterador. El iterador coge cada uno de los resultados extraídos por el extractor y trabaja con ellos. En este caso, simplemente se van a sacar los resultados obtenidos, es decir, no se realiza ninguna operación con ellos.

7. Dentro del iterador, se añade el elemento **Record Constructor**, en él se configura cómo y qué elementos se quieren sacar como resultado. Después del elemento Record Constructor, se coloca el elemento **Output** que saca los datos configurados previamente.

8. Se ejecuta el Wrapper introduciendo la palabra que se va a buscar y obteniendo los resultados.

   Primero se inserta el parámetro de búsqueda:

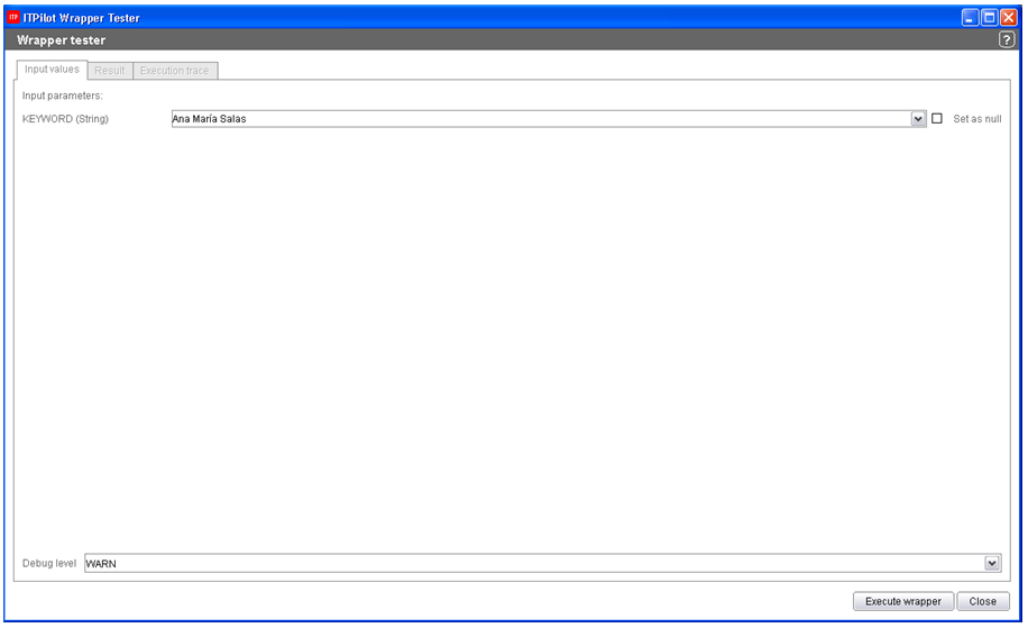


Figura 4.69: Ejecución del Wrapper

A continuación se pulsa el botón **Execute Wrapper**:

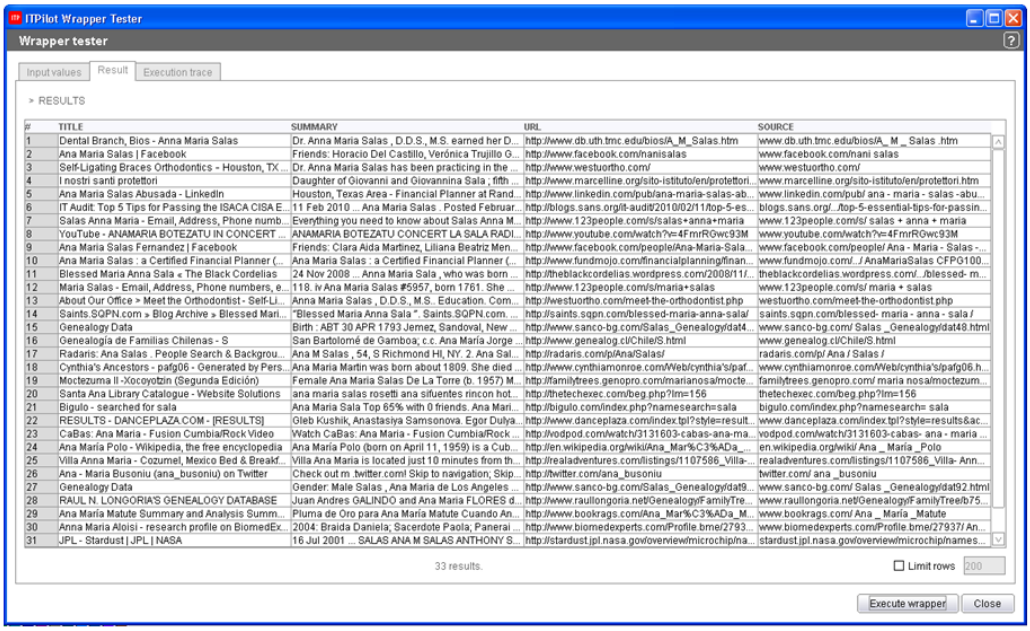


Figura 4.70: Resultado de ejecución del Wrapper

Por último, se muestra el Wrapper final que se acaba de ejecutar:

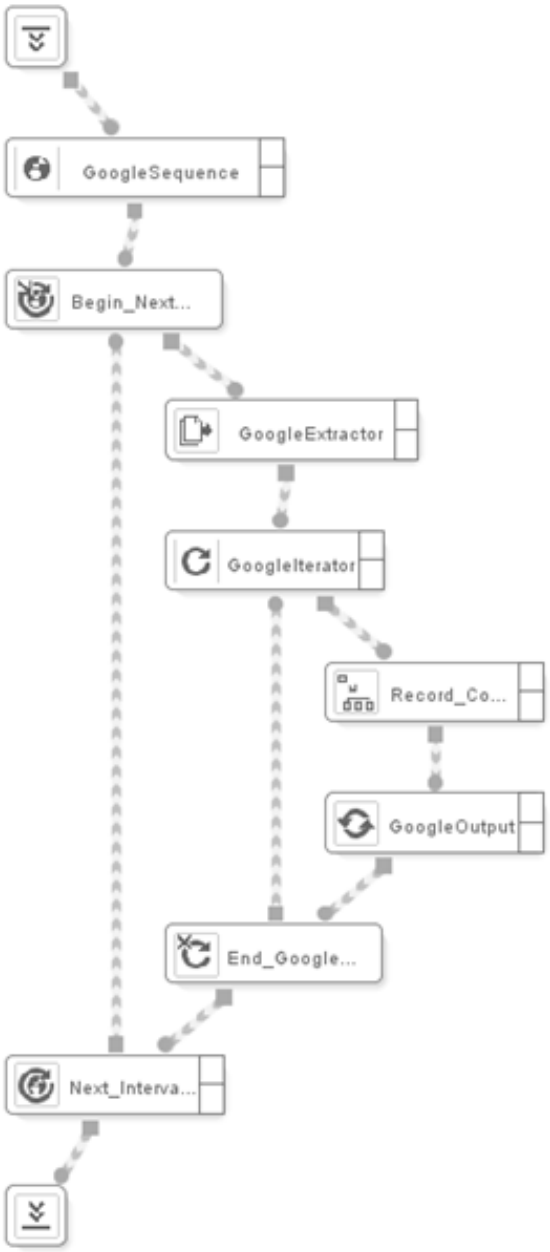


Figura 4.71: Wrapper ITP

4.5.2. Importación de fuentes VDP

Una vez se genera el Wrapper ITP, se procede a realizar la exportación del mismo a VDP. Para ello, dentro de la herramienta **ITP Generator Tool**[57], se accede a la opción del panel **Data Export Tool**, y dentro de ella la opción **Server Deploy**. Pulsando esta opción se presenta una pantalla como en la figura 4.72

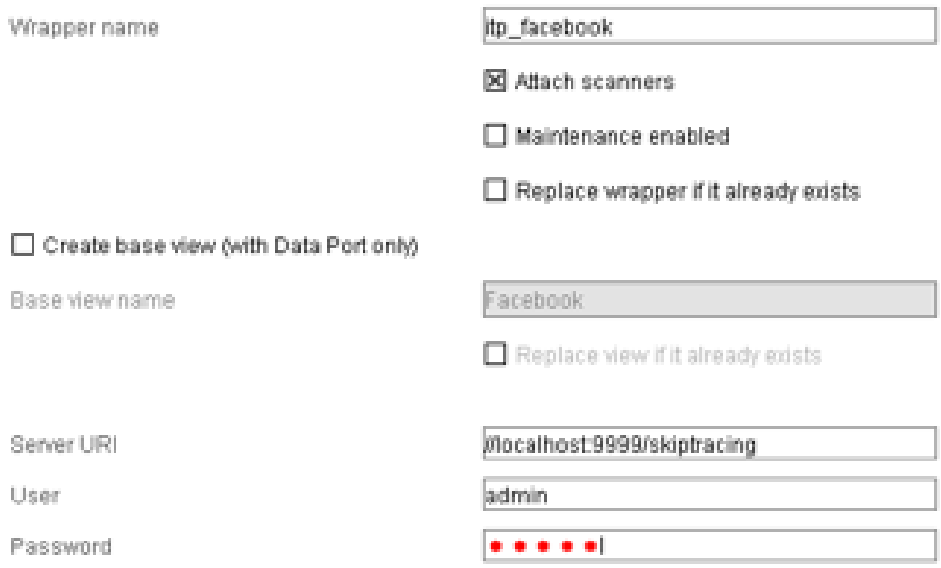


Figura 4.72: Exportación Wrapper ITP a VDP

Dentro de la ventana anterior, se añaden los valores necesarios para desplegar el Wrapper, entre ellos, el nombre que tendrá el Wrapper al ser exportado a VDP[55], URI del servidor VDP (con la base de datos de VDP a la que se quiere exportar el Wrapper), el usuario y contraseña.

Una vez se ha exportado correctamente el Wrapper, se abre la herramienta **Administration Tool de VDP**, para trabajar con él. En la figura 4.73 se observa la correcta importación del Wrapper, y partir de ahí se pueden generar vistas base y derivadas de dicho Wrapper.

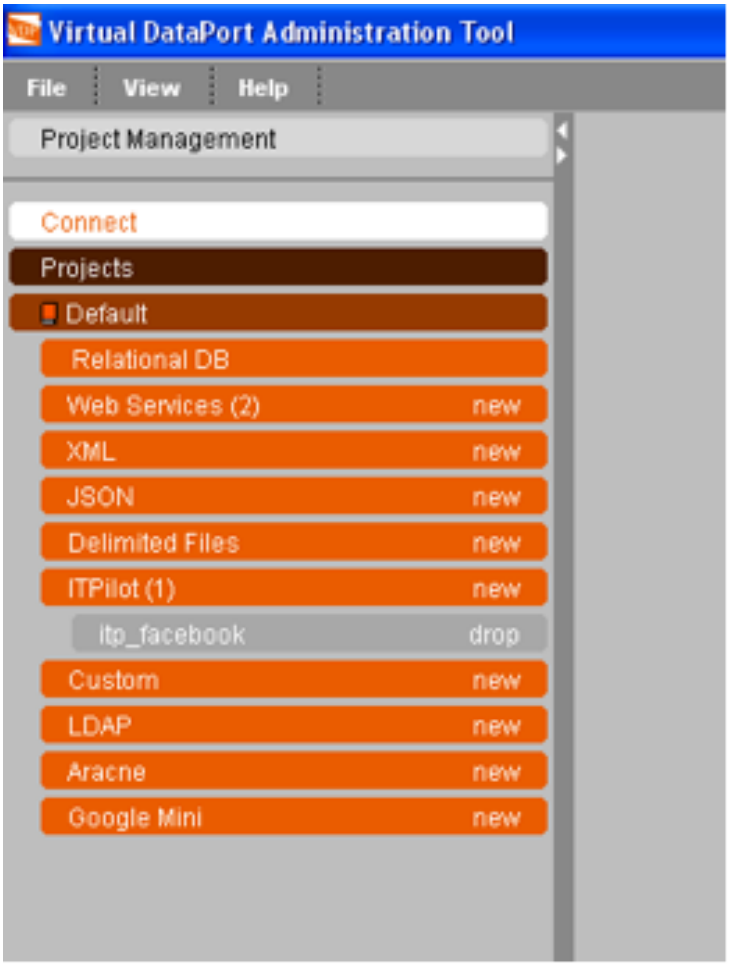


Figura 4.73: Fuentes importadas en VDP

Pinchando sobre el Wrapper **itp_facebook**, aparece otra ventana para crear la vista base. Para crear la vista base, se pueden realizar modificaciones sobre el Wrapper inicial, se puede modificar el nombre de las variables, el tipo, etc. En la figura 4.74 se ve un ejemplo de generación de la vista base a través de un Wrapper.

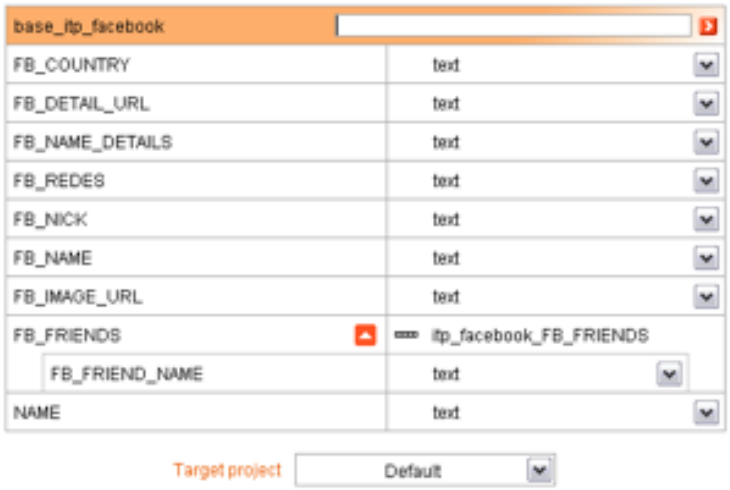


Figura 4.74: Creación de vista base a través de Wrapper ITP

Una vez se aceptan los cambios de la pantalla 4.74, se crea una vista base llamada **base_itp_facebook**.

Además de Wrapper, también se pueden añadir otras fuentes de datos, como ya se ha comentado anteriormente. A continuación se va a explicar cómo importar una base de datos a VDP.

En la figura 4.73 se observa que se pueden añadir varias fuentes, en este caso se elegirá la opción *Relational DB* y se completaran los campos de la ventana que se presenta en la figura 4.75

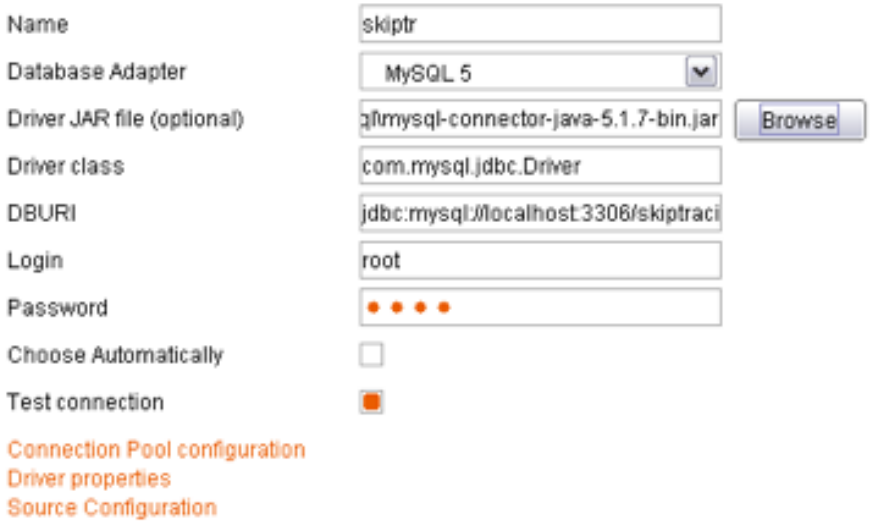


Figura 4.75: Importación de base de datos en VDP

De la base de datos importada a VDP se pueden añadir las tablas deseadas, creando para cada una de ellas una vista base con la que luego se podrá trabajar.

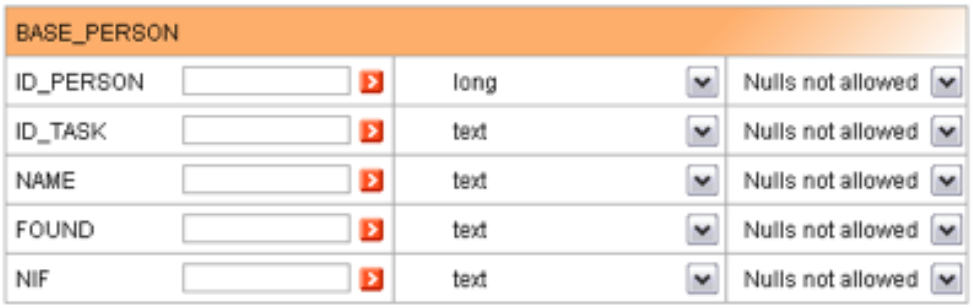


Figura 4.76: Vista base creada a través de una tabla de una base de datos

4.5.3. Procesado de vistas base a vistas derivadas

Una vez se han importado las fuentes y generado las vistas base. Se puede trabajar con las vistas base y realizar una serie de operaciones sobre ellas. Las operaciones que se pueden hacer se muestran en la figura 4.77

Figura 4.77: Operaciones que se pueden realizar

A continuación, se muestra un ejemplo de unión de vista. Para crear una vista derivada que venga de la unión de dos vista, se pincha sobre la opción **Union** y se procede a arrastrar las vistas interesadas, una vez se realiza esta acción se presenta una pantalla como la de la figura 4.78, en la que se puede configurar la unión:

Figura 4.78: Unión de dos vistas

Cuando se produce la unión, se relacionan los campos de las distintas vistas entre sí, esto se ha realizado directamente porque en este caso los campos tienen los mismos nombres (se trata de búsqueda de información en Facebook y en Facebook a través de Google), pero si no fuera así se podrían asociar manualmente. Una vez realizadas las asociaciones necesarias, y se ha modificado lo que se desea, se pulsa el botón **Aceptar** creando una nueva vista derivada con la que poder trabajar.

4.5.4. Configuración del Scheduler

Por último, se va a explicar brevemente que es necesario configurar de la herramienta Scheduler de la plataforma de Denodo, para que funcione con la aplicación Skiptracing.

Lo primero que se va a configurar son los **Datasources** necesarios, es decir, las fuentes de las que saca información Skiptracing. Hay que configurar 2 datasources diferentes:

Figura 4.79: Configuración Skiptracing

- **Skiptracing_database**: Base de datos donde se almacena la información encontrada.

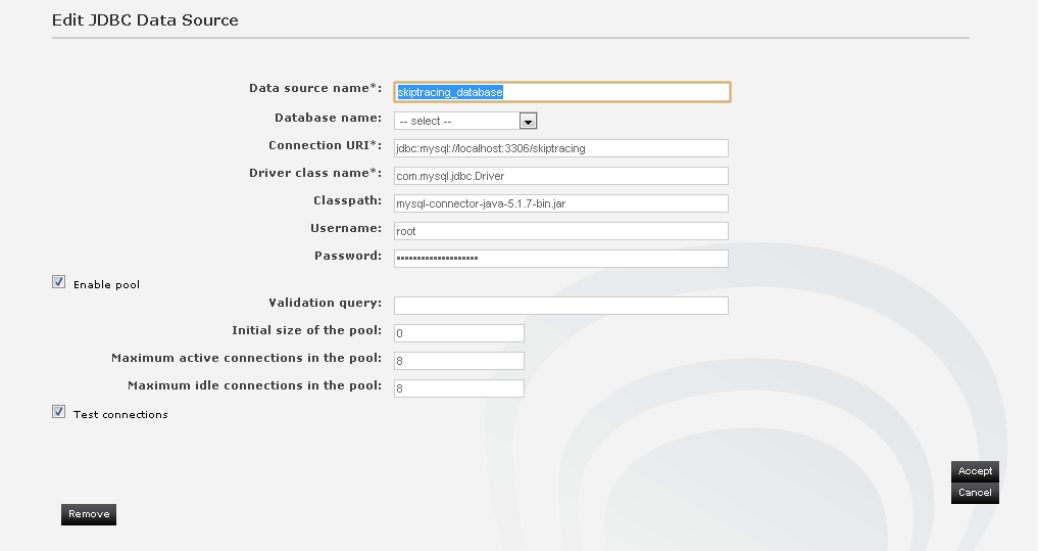


Figura 4.80: Datasource JDBC

- **Skip Tracing**: Este datasources es de VDP, ya que para la realización de la búsqueda utiliza el procedimiento almacenado que contiene VDP.

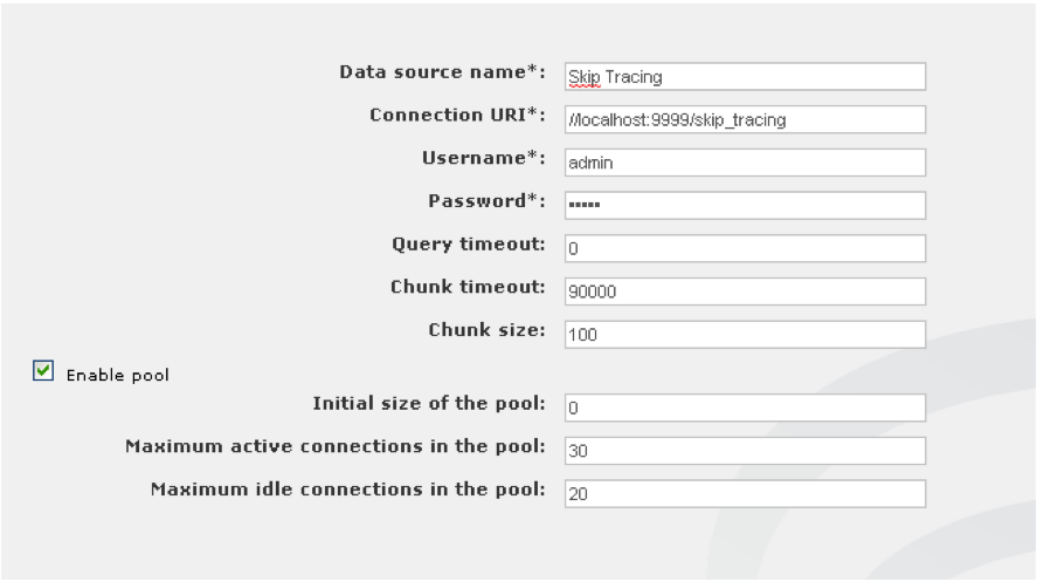


Figura 4.81: Datasource VDP

Para ello habrá que indicar la URI donde corre el servidor y la base de datos que interesa, así como el usuario y contraseña para poder acceder a VDP.

A continuación, hay que añadir el exportador de Skiptracing para exportar la información, para ello se puede adjuntar el exportador como un plugin como muestra la figura 4.82:

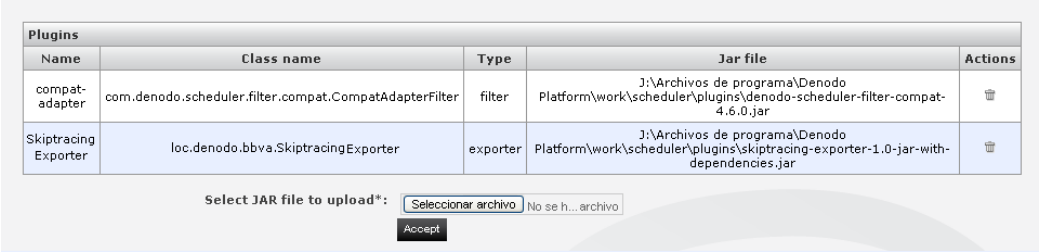


Figura 4.82: Skiptracing Exporter

A continuación se crean los Jobs, que serán las tareas que se ejecutarán en el Scheduler. En Skiptracing existen dos Jobs:

- Test 1: Este Job es el que realiza la extracción de VDP, ya que lo primero que se hace es cargar el fichero subido, obtener sus datos y posteriormente realizar la llamada al procedimiento almacenado que se encuentra dentro de VDP. Dentro del Job hay que configurar la extracción, como se ve en la figura 4.83

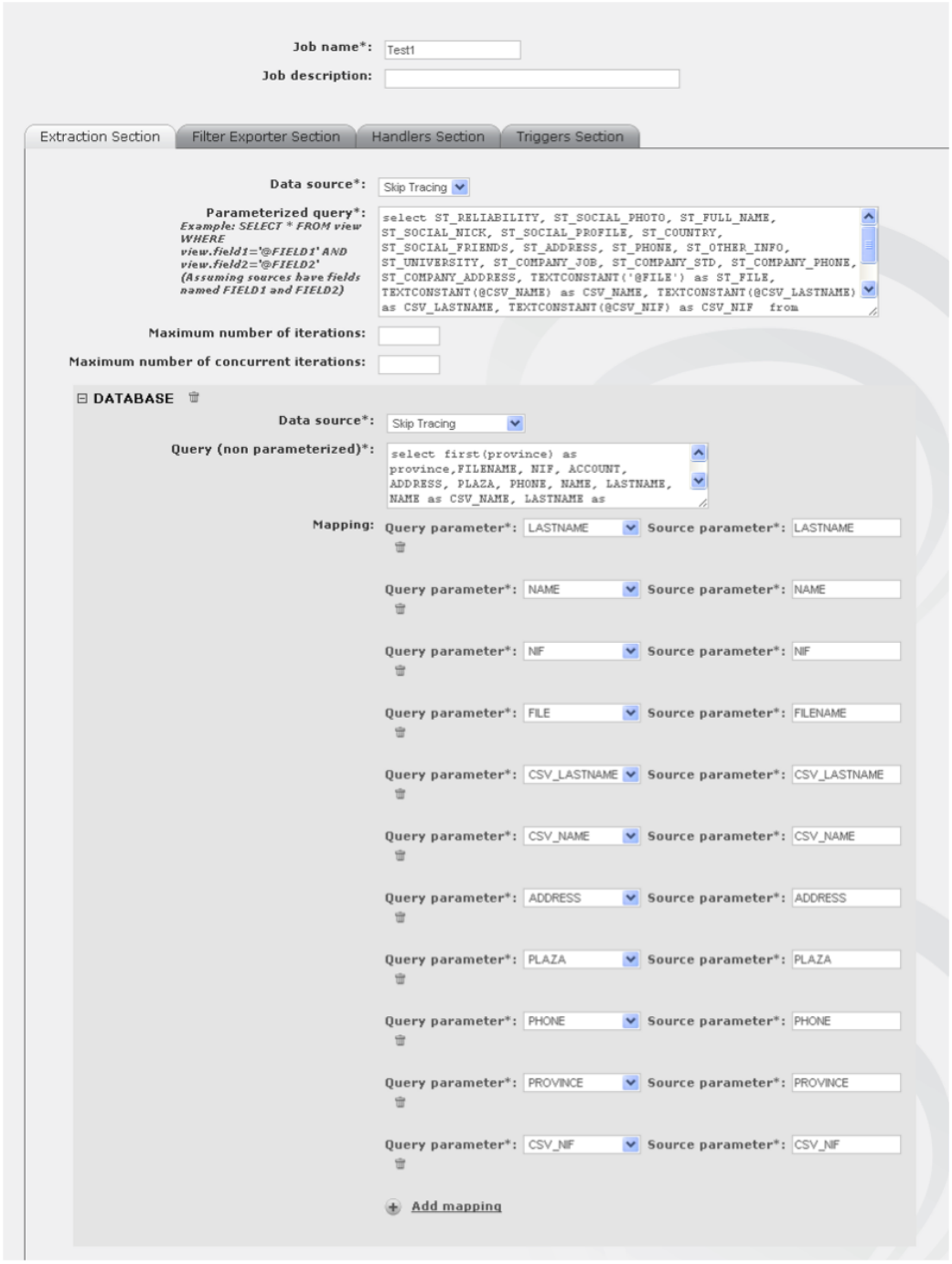


Figura 4.83: Extraction Section Test 1

Y el exportador que se va a emplear en este caso se trata del exportado subido como plugin: **Skiptracing Exporter**.

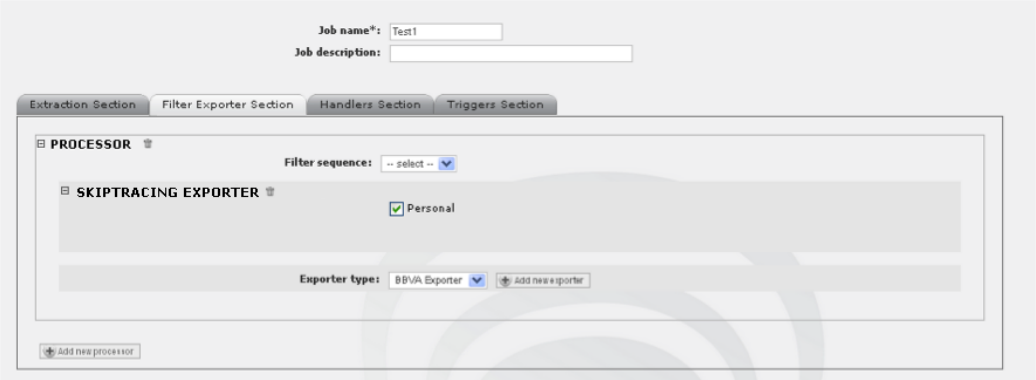


Figura 4.84: Exporter Section Test 1

- Test 2: Este Job es el encargado de revisar los resultados y todas las **personas** pasadas en el fichero y determinar si se ha encontrado información o no. Al igual que en el caso anterior, se definen las fuentes de Exportación:

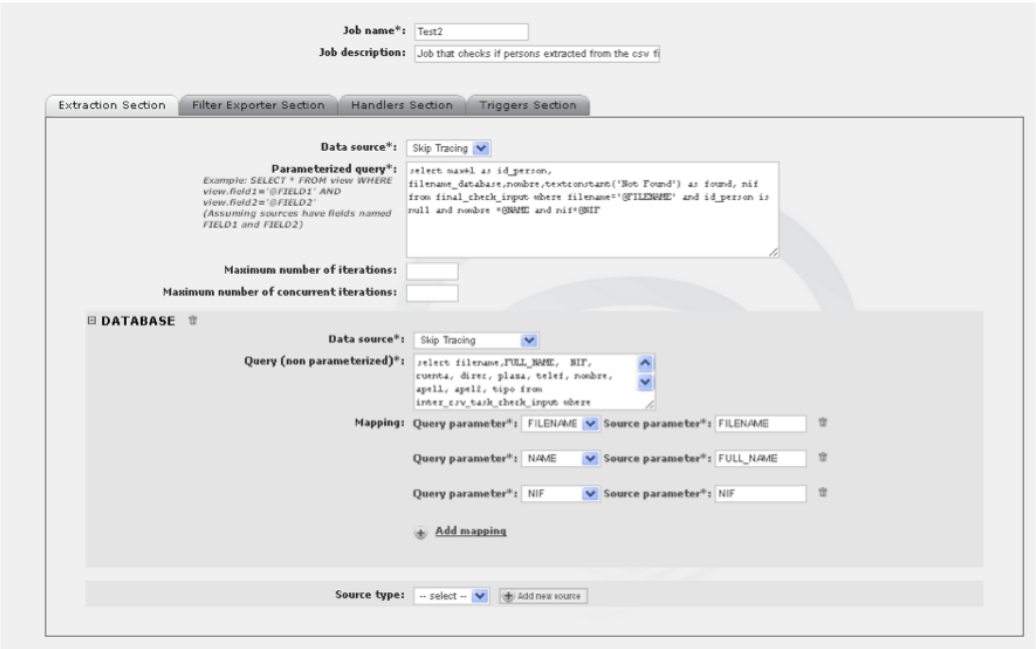


Figura 4.85: Extraction Section Test 2

Y a continuación se define la exportación de los datos:

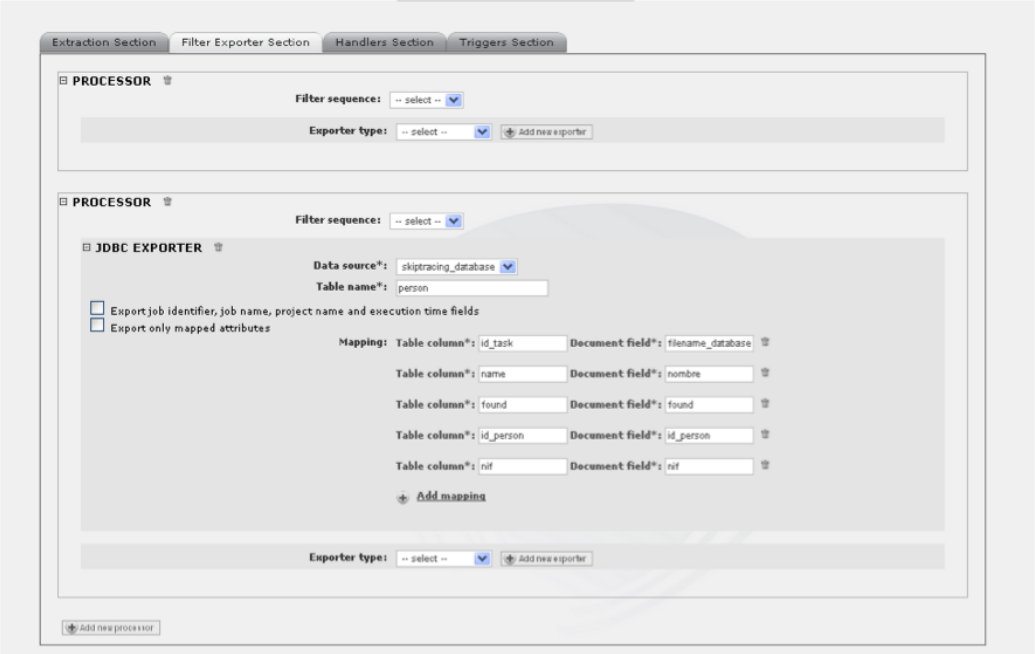


Figura 4.86: Exporter Section Test 2

En este caso se ve claramente que la información se exporta a la base de datos.

Estos Jobs son los que se ejecutan en la carga del fichero. Los Jobs se pueden ejecutar directamente al subir el fichero, llamando al Scheduler directamente desde la aplicación Web, o programar la ejecución si el fichero de configuración es muy grande.

4.6. Funcionamiento general de Skiptracing

Anteriormente se han definido las diferentes partes que existen dentro de la herramienta Skiptracing. Tal como se ha comentado todos los módulos están relacionados entre sí, como muestra la figura 4.1. Además se ha hecho un repaso por las fuentes, así como las vistas que se tienen y además se ha explicado cómo se consigue todo ello a través de la plataforma Denodo. Este capítulo viene a completar todo lo anterior, explicando paso a paso que ocurre cuando un usuario pulsa el botón buscar, o cuando se sube un fichero.

4.6.1. Búsqueda de una persona

El usuario que utiliza Skiptracing, se encuentra con un interfaz Web amigable, que le permite tanto insertar el nombre de la persona que desea buscar como algunos parámetros más para afinar la búsqueda. Para obtener más información sobre el funcionamiento de la herramienta, existe un manual de usuario en el **Apéndice A**

Cuando el usuario inserta los parámetros necesarios para la búsqueda y pulsa el botón **buscar**, se realiza una petición a base de datos de manera normal, pasando como parámetros los valores insertados por el usuario. La diferencia es que la base de datos a la que se quiere acceder no es una base de datos común, sino que se trata de una **base de datos virtual** creada por la herramienta VDP de la plataforma Denodo.

Concretamente, esta petición se realiza al procedimiento almacenado **SKIP-TRACING-PROC**. A él se le pasan todos los parámetros insertados por el usuario, y el fichero de configuración necesario en cada caso. Es decir, si el usuario quiere realizar búsqueda en Google, búsqueda en redes sociales, en blogs y foros, se realizará una petición por cada conjunto de fuentes de búsqueda. Además, si en lugar de una búsqueda simple, se quiere realizar una búsqueda profunda, la aplicación Web detectará que es así y simplemente pasará el fichero de configuración correspondiente a dicha búsqueda profunda.

Una vez ha llegado la petición al procedimiento almacenado, éste genera la consulta necesaria con los parámetros y el fichero de configuración. Se realizará esta tarea por cada llamada realizada, ejecutándose simultáneamente. La consulta que construye, pertenece a las vistas finales que se han creado a través de las fuentes de las que se saca información. Por lo tanto, si por ejemplo se realiza la búsqueda en Facebook, se realizará la llamada a la vista final que deriva del Wrapper de ITP de Facebook. Cómo las vistas finales, dependen de fuentes, en el caso del ejemplo que se ha puesto, la herramienta VDP lanzará el Wrapper para la extracción de información de dicha fuente. Una vez ha extraído toda la información del Wrapper, la información es devuelta al procedimiento almacenado con los valores definidos en la vista base final.

Cuando el procedimiento almacenado tiene todos los datos, es el momento de mirar su validez. Como ya se ha comentado anteriormente, los ficheros de configuración que generan las consultas también poseen una **puntuación** dependiendo de las veces que aparezcan los parámetros de entrada, por ejemplo: Si en el interfaz Web se busca a *Ana María Salas Fernández* y se obtiene tanto información de *Ana María Salas* como de *Ana María Salas Fernández*, los datos obtenidos con el segundo nombre obtendrán más puntuación ya que poseen el nombre completo de la persona a buscar.

Además, no solo se realiza una puntuación de validez, también se buscan tuplas repetidas. Como se ha mencionado anteriormente, es posible realizar una combinación de parámetros para generar las consultas y así afinar la búsqueda. Por ejemplo, si se busca a *Ana María Salas Fernández*, se realiza más de una búsqueda, por ejemplo, con todos los apellidos, quitando el último apellido, buscando por la inicial del nombre y los apellidos... etc. Al realizar varias búsquedas es posible que existan tuplas que estén repetidas, por lo que si es así se eliminan.

Por último, como el procedimiento almacenado realiza la búsqueda en distintas redes sociales, es posible encontrar información sobre la misma persona en dos redes sociales diferentes. Si esto ocurre, Skiptracing intentará fusionar los resultados. Para ello, en el fichero de configuración se definen los campos de los resultados que deben coincidir para que dos resultados se fusionen.

Una vez realizadas todas las tareas, el procedimiento almacenado devuelve toda la información, siendo mostrada por el interfaz Web.

La búsqueda de empresas es exactamente igual, modificando los ficheros de configuración y los parámetros de entrada, pero su ejecución es prácticamente idéntica.

4.6.2. Subida de un fichero

Esta tarea se realiza de forma parecida que en el caso anterior, con la diferencia que entra en juego la herramienta Scheduler de Denodo.

Hay dos maneras de realizar la búsqueda simultánea:

- Ejecutando el Scheduler un momento programado.
- Ejecutando el Scheduler en el momento que se sube el fichero.

Ambos procesos se producen de igual forma, por lo que se explicará desde el momento en que se ejecuta el Scheduler.

Cuando el Test1 se lanza, primero se realiza la extracción de información. Como ya se ha comentado, el fichero de configuración importado a Skiptracing debe tener una estructura en concreto. Dicha estructura se define además en el fichero **skiptracing_upload_csv** contenido en el módulo VDP de Skiptracing. Por tanto, lo primero que se realiza es la extracción de la información contenida en el fichero subido por el usuario, y separado convenientemente para su posterior procesado. En dicho procesado, se realiza la búsqueda de la provincia a través de la dirección introducida en cada persona a buscar. La búsqueda de la provincia se realiza llamando a un Webservice de páginas amarillas, que obtendrá la información de la provincia según una dirección dada.

Una vez se tienen todos los datos, con cada una de las entradas de la base de datos se realiza la llamada al procedimiento almacenado, de la misma manera que ocurre en el caso de la búsqueda simple. Una vez se obtienen los datos, estos se almacenan en la tabla **person_result** con toda la información encontrada sobre la persona. Además el Test1 almacena el valor *Found* en la columna **FOUND** de la tabla **person** si la validez del resultado supera un cierto umbral, y si no es así se almacena el valor *NotFound*.

Una vez termina el Test1, se ejecuta el Test2. Este test del Scheduler revisa todos los nombres que contiene el fichero de carga, revisando cada uno para saber si está o no en la tabla **person**. Si alguno de ellos no se encuentra en dicha tabla, inserta una tupla en la tabla **person** con el valor del campo **FOUND** a **NotFound**.

En esta tarea, se ha dicho varias veces que se exporta a la base de datos, para realizar dicha exportación se utiliza el módulo **SKIPTRACING-EXPORTER**.

Cuando el usuario consulta el fichero cargado, se le presenta una tabla con los nombres incorporados en el fichero y se indica si se ha encontrado información o no. Si no se ha encontrado información, se puede pinchar en ellos lanzando en tiempo real una búsqueda en Google, blogs y foros, ya que ésta no se almacena al realizar el lanzamiento del Scheduler (ya que su búsqueda es más rápida). Si por el contrario, la persona buscada sí que ha sido encontrada, se pueden consultar los resultados almacenados y además, al consultar sus resultados, se lanza en tiempo real la búsqueda en Web, blogs y noticias, de igual forma que en el caso anterior

4.7. Pruebas y Problemas abordados

A continuación se va a enumerar una serie de problemas que fueron encontrados en el desarrollo de la aplicación Skiptracing, además se verá como se solucionan en algunos casos.

1. **Campos de búsqueda**

   Skiptracing inicialmente realizaba la búsqueda por Nombre + Apellidos. El campo opcional no era el segundo apellido sino la provincia. El usuario podía insertar uno o dos apellidos según su criterio, en el campo de Apellidos. A simple vista esto no debería ser un problema pero a la larga dificultaba una buena búsqueda.

   Si solo se tiene un campo para los apellidos siendo posible que el usuario insertara uno o los dos apellidos de la persona, no se tenía ningún control sobre ello. Esto conlleva que:

- Al realizar la búsqueda no era posible realizarla separando los apellidos, siempre se realizaba la búsqueda con ambos apellidos.
- Al realizar la validación de los resultados se podrían perder resultados ya que es posible que en Internet la persona no inserte siempre ambos apellidos.

Solución

La solución a este problema fue añadir un nuevo campo *segundo apellido* y que en el primer campo solo se inserte el primer apellido, dejando el segundo apellido como campo opcional.

Ahora, en los ficheros de configuración, es posible introducir diferentes combinaciones con o sin el segundo apellido para afinar el resultado de la búsqueda. En caso de querer insertar la provincia se proporciona la búsqueda avanzada.

Además se mejora la validación de las tuplas. En este caso se da más puntuación si se encuentra una entrada con el nombre y ambos apellidos que si solo contiene el primer apellido.

2. **Problemas al realizar la búsqueda de empresas.**

   Linkedin solo permite realizar la búsqueda de personas, no es posible realizar una búsqueda de empresas de igual manera que se realiza la búsqueda de personas.

   Solución

   El buscador Google permite realizar una búsqueda en una página concreta. Como Linkedin tiene perfiles de empresa la búsqueda tiene dos pasos:

   - Búsqueda de la empresa dentro del navegador Google, de la siguiente manera **site:www.linkedin.com + nombre empresa**.
   - Filtrado por los perfiles de empresa. Esto es necesario ya que al realizar la búsqueda de esta forma, también devuelve resultados de personas que trabajan o han trabajado en la empresa buscada.

3. **Problemas cuando se realizan muchas búsquedas en Facebook**

   Realizando muchas pruebas, entre ellas peticiones de búsqueda en Facebook, nos dimos cuenta que al realizar un número elevado de peticiones era posible que saliera un Captcha[73].

   Captcha es el acrónimo de Completely Automated Public Turing test to tell Computers and Humans Apart (Prueba de Turing pública y automática para diferenciar máquinas y humanos), es decir, se trata de una prueba desafío-respuesta para determinar cuándo el usuario es o no humano.

En este caso no se obtenían respuesta de Facebook.

Solución

Al igual que en el caso anterior, se puede realizar una búsqueda en Google especificando el sitio. Por lo que para evitar el Captcha, se utiliza el buscador Google con la búsqueda *site:www.facebook.com palabra clave*

Este *truco* hace que si Facebook proporciona este *test* sea posible seguir obteniendo datos de Facebook. Aún así, los resultados son menos satisfactorios que los realizados directamente desde Facebook. Por lo que para utilizar la fuente que mejor información proporciona y evitar el problema anteriormente mencionado, se combinan las vistas de ambas búsquedas, es decir, la vista proporcionada por Facebook directamente y la proporcionada por Facebook a través de Google. Por lo tanto ahora se realiza la búsqueda simultáneamente en ambas fuentes. Es posible obtener resultados repetidos, pero nuestra herramienta evalúa y elimina estos resultados repetidos.

4. **Combinación de resultados**

   En Skiptracing, se intenta mostrar el máximo de información de la persona o empresa que se va a buscar. En concreto en la búsqueda de personas, se pueden combinar los resultados, en búsqueda social, si se detecta que se tiene la misma persona. Esto se realiza comparando varios campos de resultados, si coinciden estos campos se entiende que son la misma persona y se combina su información. Los campos que se comparan son configurados en los ficheros de configuración nombrados y explicados anteriormente.

   Al realizar esto, existe un problema. Los campos a comparar elegidos, en nuestro caso, son *Nombre*, *apellidos* y *Nick*, ya que se supone que el usuario emplea el mismo Nick en todas sus cuentas. El problema es que en Linkedin se pueden tener varias personas con el mismo nombre y Nick de diferentes países por lo que se tiene un problema con los resultados combinados de personas que no tenían nada que ver.

   Solución

   Para solucionar este problema sólo se permite la combinación en caso de ser dos resultados de diferentes fuentes, es decir, si el usuario es de Facebook y de Linkedin.

5. **El usuario puede no insertar su nombre completo**

   En redes sociales, y sobre todo en redes con Facebook o Sonico es posible que el usuario no inserte su nombre completo, es decir, si el usuario se llama *Ignacio Pérez* y todo el mundo le llama *Nacho Pérez*

es prácticamente seguro que se registre con el segundo nombre. Si es este el caso la herramienta no encontrará el usuario en cuestión.

Solución

Se añade una biblioteca de *Nicknames* como por ejemplo:

Ignacio ->Nacho, Iñaqui

José ->Pepe, Josito

Francisco ->Paco, Kiko

En esta tabla no se tendrán todos los *nicknames* conocidos pero ayuda a mejorar mucho la búsqueda.

6. **Muchos resultados**

   Es posible obtener un número muy grande de resultados, ya que al obtener los datos se ordenan por validez y se devuelven todos. Es posible que muchos de ellos no sean útiles, sean *basura*.

   Solución

   Se inserta en el fichero de configuración un campo numérico. Este campo representa el límite de aceptación de resultados. Es decir, si el resultado está por debajo de este número las tuplas obtenidas no se muestran como resultado.

7. **Búsqueda de un amigo de la persona buscada.**

   En el resultado de búsqueda social, dentro de los detalles de la búsqueda social, es posible realizar la búsqueda de un *amigo* de la persona buscada, realizando clic sobre ella.

   El problema en este caso, es que no se conoce la estructura del nombre del amigo en el que se pincha, es decir, no se puede saber si el nombre consta de un nombre y un apellido, un nombre y dos apellidos, nombre compuesto más un apellido, etc.

   Solución

   Se presenta un formulario en el cual el usuario que utiliza la herramienta debe seleccionar la estructura del nombre de la persona seleccionada ya que esta opción no se puede automatizar.

8. **Búsquedas repetidas**

   Al realizar la búsqueda en Google de una persona o empresa, es posible obtener información de las redes sociales en las que también se realiza la búsqueda, por lo que se tendrían resultados redundantes.

   Solución

Para evitar este problema, y gracias a las opciones que posee el buscador Google, al realizar la búsqueda de una persona se añade *-sonico -facebook -linkedin*. Con el símbolo - Google entiende que tiene que ignorar los resultados con estas palabras en sus resultados.

4.8. Comparación de Skiptracing con otras herramientas

Skiptracing realiza la búsqueda de personas y empresa en Internet, pero ya existen otras herramientas de búsqueda de personas. A continuación se realizará una descripción de las más populares:

1. **123People.es**[75]: Es un motor de búsqueda de personas en tiempo real que busca en la Web. Utiliza una algoritmo propio registrado con el que se puede encontrar datos relacionados con nombres que contienen imágenes, vídeos, números de teléfono, direcciones de correo electrónico, redes sociales, perfiles en Wikipedia, etc...

Figura 4.87: 123People.es

2. **Pipl**[76]: Filtra las búsquedas por ciudad o país, muestra más de lo que Google enseña. Además se pueden encontrar hasta direcciones de personas con solo su nombre y apellido.

Figura 4.88: Pilp

3. **ZabaSearch**[77]: Solo funciona para Estados Unidos, con solo saber el nombre de la persona se pueden encontrar direcciones y números telefónicos.

Figura 4.89: ZabaSearch

4. **Wink**[78]: Si la persona buscada utiliza Twitter, MySpace, entre otras

redes sociales, con este buscador se podrán filtrar los contenidos por ubicación geográfica o intereses como libros, música, películas, etc...

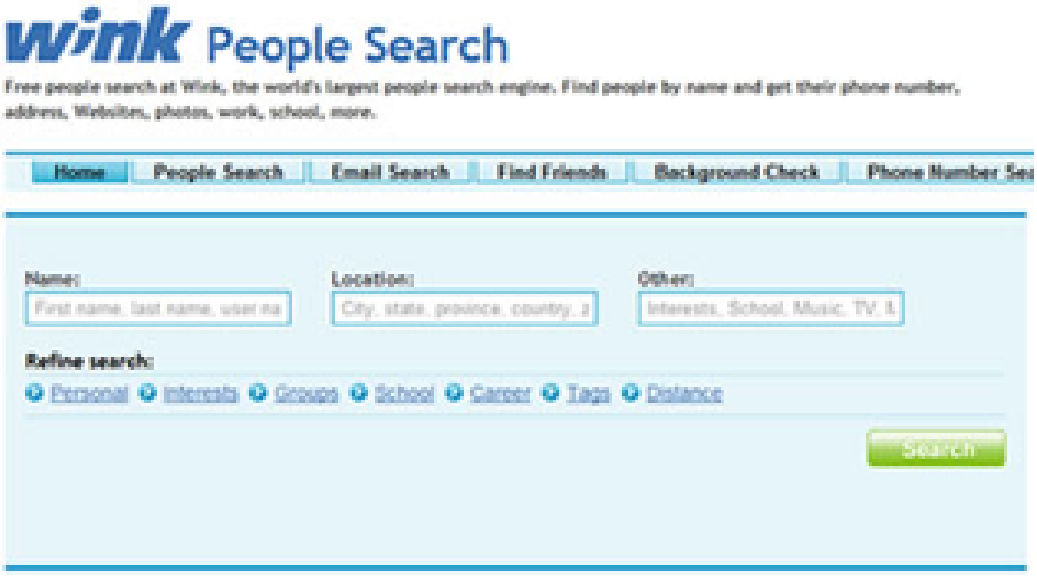


Figura 4.90: Wink

5. **Google**: Solo con escribir su nombre y correo electrónico se puede obtener una gran cantidad de información.

Figura 4.91: Google

6. **Who Is This Person?**[79](Extensión para Firefox): Es un complemento del navegador, en su menú se encuentra una información de diversos buscadores que irán directamente a opciones como Facebook o MySpace para buscar a la persona indicada.

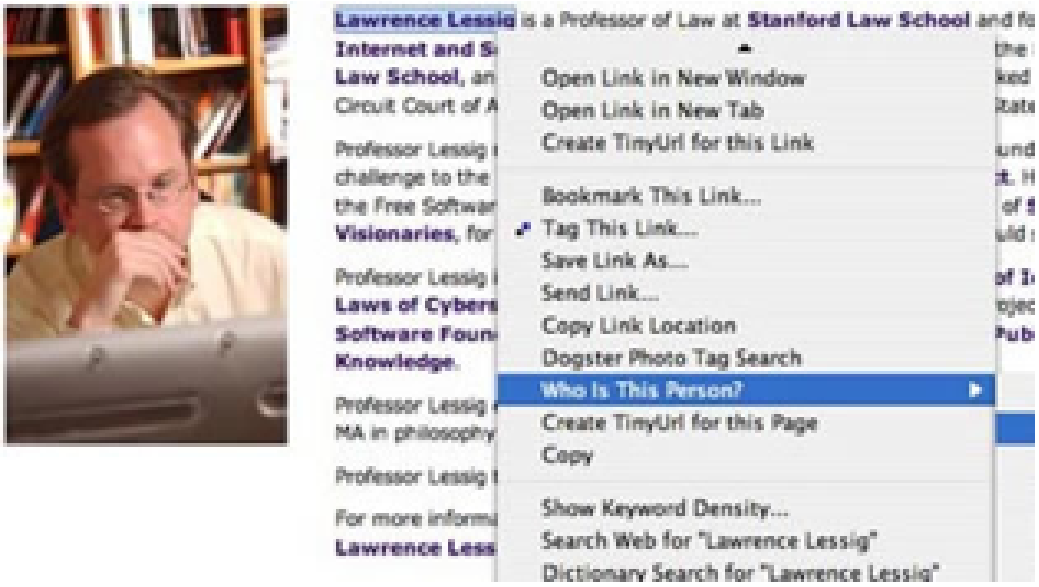


Figura 4.92: Who Is This Person?

¿Qué ofrece Skiptracing que no ofrecen estas herramientas?

Todas estas herramientas proporcionan información de Internet sobre la persona buscada, la herramienta más robusta y que mejor funciona es 123People, ya que realiza la búsqueda en varias fuentes diferentes. Viendo estas herramientas se puede pensar ¿qué ventajas aporta Skiptracing? Se enumeran a continuación:

- Skiptracing proporciona un resultado ordenado por **Validez**: Las herramientas anteriormente mostradas obtienen información devolviéndola sin post-procesado, sin embargo, Skiptracing realiza un post-procesado de la información y evalúa como de **buena** es dicha información, asociando un número de validez a cada resultado devuelto.

- Skiptracing permite seleccionar fuentes de búsqueda: Como ya se ha comentado 123People devuelve la información sobre una persona en distintas fuentes, pero es posible que no interese buscar en todas las fuentes que dispone el buscador. Skiptracing permite elegir las fuentes de búsqueda para así realizar la búsqueda en las fuentes deseadas.

- Ofrece dos tipos de búsqueda, búsqueda simple y profunda, realizando varias combinaciones con los datos pasados como parámetro.
- Además, podemos destacar que Skiptracing incorpora un diccionario de "Nicknames", es decir, diminutivos o nombre populares que las personas pueden utilizar en Internet.
- Skiptracing además de permitir realizar una búsqueda por el nombre de la persona, también permite insertar más información para afinar la búsqueda.
- Skiptracing permite la automatización de la búsqueda a través de la carga de un fichero.
- Las soluciones anteriormente mencionadas, son muy útiles para la búsqueda **puntual** de personas. En un entorno empresarial Skiptracing ofrece la posibilidad de una fácil adaptación a las necesidades del cliente, es decir, Skiptracing puede añadir nuevas fuentes de información de manera sencilla añadiendo nuevas fuentes a VDP y en caso de que sea necesario, creando un nuevo fichero de configuración asociado.
- En cuanto al punto anterior, Skiptracing puede ampliar su búsqueda, es decir, actualmente Skiptracing no solamente busca personas, sino que también empresas, con lo cual sería posible añadir una nueva búsqueda, de cualquier otra cosa, de manera sencilla.

Capítulo 5

Historia del proyecto

Este proyecto fin de carrera fue realizado en la empresa **Denodo Technologies** como prácticas en empresa. El proyecto **skiptracing** fue un proyecto piloto para un cliente de Denodo Technologies. Inicialmente el proyecto no estaba concebido para la búsqueda de empresas, únicamente realizaba la búsqueda de personas. Los requisitos iniciales del proyecto, hablados con el cliente eran:

- Búsqueda de personas por nombre, apellidos y provincia.

  En este caso, los apellidos iban todos en un campo, no se separan como ocurre ahora. La separación se realizó para poder conseguir mayor número de combinaciones en las búsquedas y además para un cálculo más preciso en la validez.

- Una vez realiza la búsqueda individual de personas, se permitían cambiar los parámetros de búsqueda, tal y como ocurre actualmente. Lo que no existía era el formulario de búsqueda avanzada que se tiene actualmente, por lo que si se querían ampliar las opciones de búsqueda era necesario realizar previamente una búsqueda simple, y utilizar el formulario de consolidación de resultados.

- Existía una búsqueda en actividad, no solo buscaba en redes sociales y en Internet, sino que también realizaba la búsqueda en el Segunda mano y páginas amarillas.

- Búsqueda a través de fichero, esta opción apenas ha cambiado. Este tipo de búsqueda funciona de la misma forma, exceptuando la búsqueda en actividad que se ha eliminado por completo.

- No se permitía elegir las fuentes de búsqueda.

- Solo se realizaba la búsqueda en Google, es decir, no se realizaba ni el blogs ni en noticias.

Este proyecto se presentó al cliente, y una vez terminó se pudieron realizar mejoras para el presente proyecto fin de carrera. Las mejoras introducidas con respecto al proyecto inicial fueron las siguientes:

- Cambio en el formulario inicial, se inserta un nuevo **segundo apellido** y se elimina el campo provincia. Esto conlleva una mejora a la hora de las comparaciones para el cálculo de la validez, y si se quiere insertar la provincia, se puede realizar directamente en la búsqueda avanzada.
- Se añade la búsqueda en blogs y en noticias.
- Se elimina la búsqueda en actividad. Inicialmente se elimina de la búsqueda principal y se añade al perfil de usuario de redes sociales. Así se realizaba una búsqueda más precisa, se eliminó ya que no devolvía resultados satisfactorios en la mayoría de los casos y ralentizaba mucho la aplicación.
- Se añade un formulario de búsqueda avanzada, que contiene más campos para la búsqueda.
- En la búsqueda avanzada se añade la posibilidad de elegir las fuentes a buscar.
- Se añade la opción, en la búsqueda avanzada, de elegir búsqueda simple y búsqueda profunda.
- Se añade la búsqueda de empresas con las mismas características de personas.

En cuando a la búsqueda se añaden las siguientes mejoras:

- Wrapper de facebook_google para evitar el captcha[73] que tiene facebook cuando se realizan muchas búsquedas seguidas.
- Se añade la búsqueda por diminutivos y nombre populares, por ejemplo, si el usuario que se busca se llama Francisco se realiza la búsqueda por Kiko, Paco, etc.
- Se añaden nuevas funciones que permiten obtener más parámetros para el cálculo de la validez. Esto se modifica al añadir un nuevo campo Segundo Apellido.

Capítulo 6

Conclusiones

Como ya se comentó en la introducción, el objetivo de este proyecto es triple:

- Crear una aplicación que será capaz de extraer información de Internet.
- Demostrar el buen funcionamiento de la plataforma Denodo.
- Demostrar que es posible obtener bastante información de redes sociales y de Internet en general a pesar de las medidas de privacidad que poseen las redes sociales.

Crear una aplicación que será capaz de extraer información de Internet.

Skiptracing es capaz de obtener información de Internet según un *nombre y apellido* o *empresa* dada. Es capaz, no solo de obtener información, sino también de clasificar los resultados, es decir, es capaz de devolver la búsqueda ordenando los resultados por orden de validez.

En este punto, Skiptracing es capaz de obtener información relevante de una persona en Internet, cuanta más paciencia se tenga realizando la búsqueda profunda, más aumentan las posibilidades de encontrar lo que se está buscando.

Aunque Skiptracing funciona razonablemente bien, tiene un gran problema de uso a largo plazo. Si la fuente de información cambia mucho será imposible realizar la extracción de la información, es decir, depende en gran medida de la estructura de las fuentes. Aunque para que esto ocurra, la página fuente tiene que cambiar radicalmente su estructura. Esto es así, porque Skiptracing extrae la información teniendo en cuenta la estructura HTML (a través de la plataforma Denodo) de la página de la que se quiere extraer la información. En dicha extracción no se tiene en cuenta la página completa, sino únicamente los componentes HTML que rodean la información que se

quiere extraer, por lo que para que dejara de funcionar la página debería cambiar radicalmente.

Un ejemplo de que esto, es que Google cambió su apariencia añadiendo un menú de búsqueda a la izquierda. Aún cambiado parte del HTML de la página Skiptracing siguió siendo capaz de extraer toda la información necesaria. En caso de que una fuente cambie radicalmente, y no se pueda extraer información, simplemente será necesario modificar el Wrapper de extracción de información teniendo en cuenta la nueva estructura. Una vez vuelve a funcionar, se exporta el Wrapper a VDP con el mismo nombre que tenía anteriormente, lo que hará que en VDP se actualizará todo teniendo en cuenta el nuevo Wrapper. Por lo tanto, aún teniendo que cambiar el Wrapper de alguna fuente, la modificación de código es nula, teniendo que cambiar únicamente el Wrapper en la herramienta de administración de ITP.

Por lo que se puede afirmar que este objetivo ha sido conseguido, obteniendo una aplicación útil y robusta en la extracción de información en Internet. Una aplicación que no solo permite la extracción de huella digital de personas y empresas, sino que además es una aplicación de fácil expansión, es decir, está realizada de manera genérica utilizando ficheros de configuración, para que sea fácil realizar una futura ampliación de fuentes.

Demostrar el buen funcionamiento de la plataforma Denodo

Como ya se ha comentado anteriormente, este proyecto se realizó como prácticas en empresas en la empresa Denodo Technologies, por lo tanto el actual proyecto pertenece a un Piloto real de dicha empresa. En el actual proyecto se ha construido teniendo en cuenta el potencial de la plataforma de Denodo, realizando el desarrollo en torno a ella. Por tanto, en este proyecto no solamente se explica el funcionamiento de la herramienta final, sino también se quería presentar la utilidad de la plataforma de Denodo, y su potencial. Aunque hay que tener en cuenta que la plataforma Denodo posee muchas más utilidades y formas de uso, en este proyecto se presentan sus herramientas más potentes y una forma de utilización.

Además, utilizando esta herramienta, se desarrolla de forma más sencilla y rápida el código de la aplicación. La aplicación Web, es una aplicación muy sencilla que únicamente realiza peticiones y muestra resultados, teniendo toda la funcionalidad en el procedimiento almacenado contenido en VDP. Además, la generación de las fuentes y Wrappers se realiza de manera sencilla utilizando las interfaces gráficas que Denodo Platform proporciona.

Demostrar que es posible obtener información de redes sociales y de Internet en general.

Este punto es el más difícil de demostrar, ya que para realizar dicha demostración sería necesario un uso continuado de la plataforma. Aún así, se va a realizar una lista de posibles *trucos*, que ya se han comentado, para

evitar medidas de *seguridad*:

- Facebook presenta un buscador completamente accesible a terceros. Cuando realizas la búsqueda muestra 3 páginas de resultados. En cada página posee un listado de posibles personas que estás buscando. Se puede acceder a un perfil *restringido* de dicha persona. Este perfil proporciona cierta información del perfil que consultamos, ofreciendo información de sus amigos, ambiciones y redes a las que está asociado, además permite ponerse en contacto con dicha persona.

  Por lo tanto, es muy fácil saber al menos *algo* de información de cualquier persona que posea un perfil en Facebook. Además, hay que destacar que Facebook devuelve una respuesta diferente cada vez, es decir, te muestra una lista de amigos diferente y redes a la que estas conectado, por lo que realizando varias búsquedas se puede obtener bastante información de la persona. Existen filtros y configuraciones avanzadas que permiten que esto no ocurra, pero para ello tiene que ser configurado, Facebook por defecto no lo realiza.

- Además, Facebook posee el denominado *Captcha*, que intenta evitar las búsquedas masivas en esta red social. Se evita este sistema de seguridad realizando, además que en Facebook, una búsqueda en Google. Para más información sobre esta solución, en el apartado de *Skiptracing* dentro de *Problemas abordados* se explica cómo es posible evitar este y otros problemas encontrados al desarrollar la herramienta.

- Otro problema al que nos enfrentamos al buscar a una persona en Internet, es que pueden registrarse con un nombre diferente del nombre de pila. Para intentar mejorar la búsqueda se emplea un *diccionario de nicknames* para agudizar la búsqueda.

- Linkedin no ofrece buscador de empresas, aún así con Google hemos sido capaces de obtener información sobre empresas.

Muchas personas que utilizan redes sociales como Facebook o Sonico, publican sus datos sin ningún tipo de privacidad sin ser conscientes de que se puede acceder a ella de una manera sencilla.

Como ya se ha comentado en la introducción de este proyecto, existe una problemática en cuanto a la información que publicamos en las redes sociales. Esto hace que ciertos grupos, tales como desarrolladores o los cesionarios de los datos recogidos a través de estas plataformas, tengan acceso a ciertos datos que se consideran personales y en algunos casos pueden ir más allá de nombres y direcciones de correo, pudiendo tratarse de datos que la actual normativa de protección de datos sitúa en un nivel alto de protección y que sólo pueden ser tratados en muy contadas ocasiones y aquí circulan libremente.

Pese a lo expuesto anteriormente, no hay que ver las redes sociales como una fuente de incontables peligros y llenas de usuarios malintencionados, siempre hay que analizar las situaciones y las nuevas herramientas con una cierta perspectiva , aunque llamen mucho la atención, sobre todo en los medios de comunicación.

Las redes sociales pueden ser muy beneficiosas y útiles, para ciertas empresas y para ciertos usuarios, es más cómodo interactuar a través de la propia red o plataforma, ya que al híper-contextualizar la publicidad se consigue que sólo se muestre la que pueda ser de su interés, según los gustos expresados en la propia red o plataforma. También hay que tener en cuenta que determinados tipos de redes promueven la realización de contactos profesionales y fomentan la interactuación entre profesionales del mismo sector.

Pero, no solo hay que tener en cuenta la responsabilidad de las redes sociales, los usuarios también son, en parte, culpables de la información pública al alcance de todos. Todas las redes tienen distintos grados de privacidad entre los que el usuario puede optar. Pero ninguno de ellos protege al 100 % al usuario. Facebook lo reconoce dentro de sus cláusulas de privacidad: *A pesar de los controles de seguridad Facebook no puede garantizar que esté totalmente libre de todo contenido ofensivo, o ilegal ni que sus miembros no se encontrarán con conductas ilegales.* Además las redes sociales, no son perfectas, existen agujeros por los que se filtra información, sobre todo si el usuario no es consciente al 100 % de lo que está publicando y con qué nivel de privacidad.

Lo que queremos destacar es que aunque las redes sociales tienen sus perfiles de privacidad mucha gente no sabe utilizarlos, y muchos individuos o *extractores de información* puede aprovecharse de los *huecos* que existen entre la redes de amigos que hay en las redes sociales.

Nadie pone en duda los beneficios de este tipo de redes, pero siempre hay que tener en cuenta ciertas conductas que hay que evitar:

- No dar más datos personales de los estrictamente necesarios.
- Ajustar los perfiles de privacidad para que sólo sean visibles por los contactos admitidos.
- No agregar contactos a los que no se conoce.
- En caso de detectar una conducta que puede ser ofensiva, ponerla en conocimiento del responsable de la red.
- Si se entiende que tal conducta es constitutiva de un delito, ponerla en conocimiento de las Fuerzas y Cuerpos de Seguridad del Estado.

En este proyecto nos hemos centrado mucho en las redes sociales, esto ha sido así porque, quizás, son las que más información personal nos puede aportar. Por último, destacar, que mas allá de las redes sociales, con un motor de búsqueda como Google, que nos permite obtener todo tipo de información en Internet, si además somos capaces de obtenerla y clasificarla, se pueden realizar aplicaciones de extracción de información muy potentes.

Capítulo 7

Trabajos Futuros

La aplicación **Skiptracing** puede ser utilizada para multitud de búsquedas diferentes, gracias a su sencillez y a la utilización de la plataforma Denodo. A continuación se enumeran posibles mejoras de la aplicación, conservando su funcionalidad:

- Añadir la búsqueda en más redes sociales. Esto ayudaría a obtener más información sobre la persona, además como se puede elegir la fuente de búsqueda se podría elegir una u otra en cada caso, así la búsqueda no es tan lenta. Para añadir esta ampliación sería necesario modificar las siguientes partes del proyecto:
  - Añadir una nueva fuente que apunte a la nueva red social, creando un Wrapper de ITP.
  - Añadir el Wrapper a VDP y añadirlo a la vista derivada de búsqueda social, a través de una unión.
  - Modificar la aplicación Web, en la búsqueda avanzada, para poder seleccionar la nueva fuente.
- Añadir la búsqueda en otros buscadores. Actualmente se está utilizando el buscador **Google**, sería una mejora añadir la búsqueda no solo en este buscador sino también en otros populares como por ejemplo **Bing**[74]. Para añadir esta ampliación sería necesario modificar las siguientes partes del proyecto:
  - Añadir la nueva fuente que apunte al nuevo buscador, creando un Wrapper de ITP.
  - Como pertenece a búsqueda en Web, añadimos dicho Wrapper a VDP y lo añadimos, mediante una unión, a la vista derivada de búsqueda en Google Web.

  - Modificar la aplicación Web, en la búsqueda avanzada, para poder seleccionar las fuentes en la búsqueda Web.

- Añadir la búsqueda por fichero para empresas, ya que actualmente está solo hecha para personas. Para crear esta ampliación, habría que realizar las siguientes modificaciones:

  - Añadir una nueva pestaña, para añadir la opción de subida de fichero para empresas.
  - Crear la estructura del fichero que se va a subir.
  - Crear el Job que realizará la búsqueda en redes sociales de empresas
  - Modificar la aplicación Web para que muestre los resultados de la búsqueda.

- Cuando se realiza la búsqueda en redes sociales, se obtiene una lista de amigos, sería útil poder identificar a los *posibles familiares*, teniendo en cuenta los apellidos de la persona. La realización de esta ampliación se puede llevar a cabo modificando, únicamente, la aplicación Web. Cuando tengamos un usuario que tenga los apellidos iguales o uno de ellos iguales, podemos añadir un icono que indique posible parentesco.

- Añadir la configuración del umbral de validez. Actualmente está fijado por nosotros, pero sería interesante que el usuario determinara el umbral de validez al realizar la búsqueda a través de fichero. La realización de esta ampliación sería compleja, ya que actualmente utilizamos ficheros de configuración estáticos, habría que modificar la filosofía de la aplicación para que en lugar de usar los ficheros de configuración, utilizara los parámetros insertados por el usuario.

- Se podría personalizar más aún la herramienta, para que el usuario tenga más control sobre las búsquedas y los ficheros de configuración. No solo modificar el umbral de validez, como se comenta en el apartado anterior, sino poder configurar las combinaciones de los parámetros de entrada para realizar la búsqueda, poder configurar también la puntuación que se da en cada caso para el cálculo de la validez, en definitiva, que el usuario tenga total poder en la realización de la búsqueda y muestra de los resultados. Esta ampliación, al igual que en el caso anterior, depende de los ficheros de configuración. Para realizar la modificación habría que cambiar la aplicación para que en lugar de usar los ficheros de configuración, utilizara los parámetros insertados por el usuario.

- Se podría crear un complemento Firefox que sea un extractor de huella digital. Para realizar esta ampliación habría que modificar:

- Primero habría que publicar como un Webservice el procedimiento almacenado que procesa las consultas y devuelve los resultados. Así el complemento Firefox simplemente tendría que realizar consultas al Webservice creado y recoger los resultados. La publicación del procedimiento almacenado como un Webservice es muy sencilla, ya que la plataforma Denodo nos proporciona un mecanismo para realizarlo de manera sencilla
- Crear el complemento Firefox, utilizando javascript. El complemento debería realizar peticiones al Webservice publicado en el apartado anterior.

Hay que destacar que teniendo en cuenta la forma en la que está implementado Skiptracing, se pueden realizar búsqueda de muchas cosas diferentes. Es decir, se podría adecuar a las necesidades del cliente, realizando el desarrollo de manera similar a como se realiza actualmente.

APÉNDICES

Apéndice A

Manual de utilización de la aplicación Skiptracing

A.1. Introducción

La aplicación Web de Skiptracing permite buscar la huella digital de personas en la Web a través de la búsqueda en distintas fuentes. Esta búsqueda puede ser individual o a través de un fichero realizando la búsqueda de un número mayor de personas. Además también permite buscar información de empresas en la Web.

A.2. Manual de usuario

A.2.1. Inicio

Figura A.1: Pantalla principal Skiptracing

En la pantalla, mostrada en la Figura A.1, tenemos el formulario principal de la aplicación. Esta pantalla contiene un menú que se encuentra a la derecha de la pantalla. En él se puede seleccionar, la búsqueda de personas como se muestra en la Figura A.1, búsqueda de empresas como se muestra en la Figura A.2 o subir un fichero para realizar más búsquedas como se muestra en la Figura A.3.

Figura A.2: Formulario búsqueda de empresas

Figura A.3: Formulario de subida de ficheros

A.2.2. Búsqueda simple de personas

Para realizar dicha búsqueda se introducen los datos en los campos de la Figura A.4.

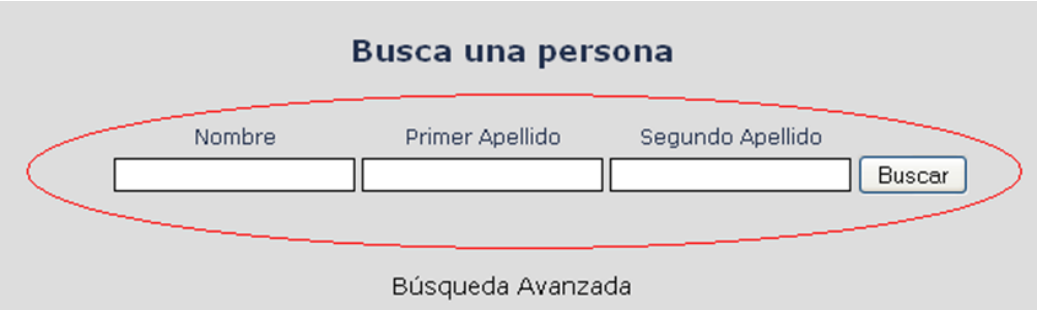


Figura A.4: Formulario Skiptracing

Para realizar la búsqueda es preciso introducir el nombre de la persona y primer apellido de forma obligatoria, el segundo apellido es un parámetro opcional.

A.2.2.1. RESULTADO EN LA WEB

Por defecto, el resultado inicial que se puede observar es el resultado en la Web. Dicha búsqueda se realiza en el buscador Google. A continuación, en la Figura A.5 se observa un resultado de una búsqueda:

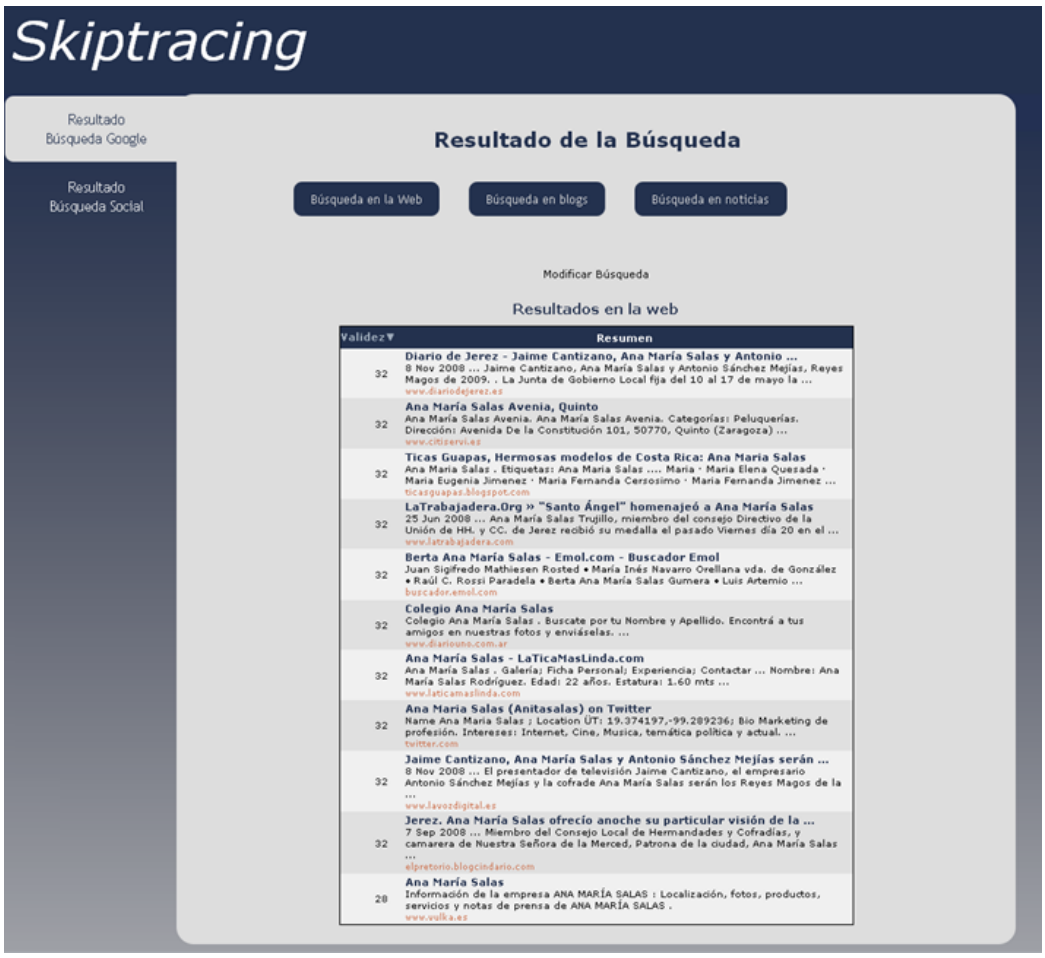


Figura A.5: Búsqueda Web

Se observa como resultado dos columnas en la Figura A.5. La primera de ellas muestra la validez del resultado y la segunda el resumen. Dicho resumen está formado por tres partes:

- Título: proporciona un enlace a la página que contiene datos sobre la persona que se está buscando.
- Resumen: con el contenido de la página a la que enlaza el título.
- URL: URL principal de donde se obtiene la información.

A.2.2.2. RESULTADOS EN BLOGS

Pulsando sobre el botón *Búsqueda en blogs*, se accede al resultado mostrado en la Figura A.6:

Figura A.6: Resultado en blogs

Se observa como resultado, al igual que el apartado anterior, tenemos dos columnas. La primera de ellas muestra la validez del resultado y la segunda el resumen. Dicho resumen está formado por tres partes:

- Título: proporciona un enlace a la página que contiene datos sobre la persona que se está buscando.

- Resumen: con el contenido de la página a la que enlaza el título.

- URL: URL principal de donde se obtiene la información.

A.2.2.3. RESULTADO EN NOTICIAS

Si se pulsa sobre el botón *Búsqueda en noticias*, se accede al resultado mostrado en la Figura A.7:

Figura A.7: Resultado en noticias

Se observa como resultado, al igual que el apartado anterior, tenemos dos columnas. La primera de ellas muestra la validez del resultado y la segunda el resumen. Dicho resumen está formado por tres partes:

- Título: proporciona un enlace a la página que contiene datos sobre la persona que se está buscando.
- Resumen: con el contenido de la página a la que enlaza el título.
- URL: URL principal de donde se obtiene la información.

A.2.2.4. RESULTADO EN REDES SOCIALES

Pulsando en la opción *Resultado búsqueda social*, del menú de la izquierda se accede al resultado en redes sociales. A continuación, se muestra el resultado en la Figura A.8:

Figura A.8: Resultado en redes sociales

Se puede observar que la tabla está formada por tres columnas:

- Validez: Esta columna proporciona una puntuación indicando la confianza del resultado.
- Imagen: En esta columna se pueden ver las imágenes que el usuario tenga en las redes sociales a las que pertenece.
- Datos: En esta columna se muestran los datos principales de cada persona: Nombre completo, Nick, ubicación y empresa en la que trabaja.

Podemos observar en la parte derecha de cada tupla que tenemos un icono, marcada en rojo en la Figura A.8. Si se pulsa dicha flecha se abre la pantalla mostrada en la Figura A.9:

Figura A.9: Pantalla Detalle

En la se puede ver más información sobre la persona encontrada. En este caso tenemos información sobre su localización, Nick, puesto, empresa y universidad. Si se encontraran más información se rellenarían los campos oportunos.

En la opción de Información Adicional se tiene un enlace con la página de la que se ha extraído la información anteriormente mostrada.

Es posible que si se encuentran varios usuarios con varios campos iguales, la aplicación, los fusione teniendo el resultado que muestra la Figura A.10. Esta fusión se produce ya que se ha encontrado a la misma persona en dos redes sociales.

Figura A.10: Resultado búsqueda en facebook fusionada

En la pantalla mostrada en la Figura A.9 hay que destacar, que si se extrae información de una persona y se obtiene la empresa en la que trabaja, se puede pinchar sobre el nombre de la empresa. En ese caso se abrirá una nueva ventana y se realizará la búsqueda, como en el apartado de búsqueda de empresas.

Además, también se puede hacer clic sobre cualquiera de las personas que se enumeran en el apartado *Amistades*, abriéndose una nueva ventana y realizando la búsqueda de forma normal, con los datos obtenidos al pinchar.

A.2.2.5. FORMULARIO DE CONSOLIDACIÓN

Se puede observar que la pantalla de resultados muestra un enlace *Modificar búsqueda* que muestra un nuevo formulario con más campos y más opciones. El formulario permite almacenar los datos que vayamos encontrando de la persona para afinar los resultados obtenidos.

Dicho formulario es el que se presenta en la Figura A.11.

Figura A.11: Formulario de Consolidación I

En el formulario de la Figura A.11 contiene los siguientes campos:

- DNI de la persona que se está buscando
- Nombre de la persona que se está buscando.
- Apellidos de la persona que se está buscando. Se puede introducir sólo el primer apellido o el primer y segundo apellido.
- Nick que emplea la persona que se está buscando en Internet.
- Teléfono personal de la persona que se está buscando.
- Dirección de la persona que se está buscando.
- Provincia de la persona que se está buscando.
- Ciudad de la persona que se está buscando.
- País de la persona que se está buscando.
- Información Adicional de la persona que se está buscado, que pueda servir para su búsqueda.
- Estudios: Lugar de estudios de la persona que se está buscando.
- Empresa en la que trabaja la persona que se está buscando.
- Tlfn. de Empresa donde trabaja la persona que se está buscando.

En el formulario de la Figura A.11 se pueden realizar las siguientes acciones:

- **Validar Datos**: pulsando sobre este botón se guardan en la base de datos local la información introducida en los campos del formulario mostrado en la Figura A.11. Inicialmente se rellenarán automáticamente el campo nombre, apellidos y, opcionalmente, la provincia. Si se quieren añadir más campos para realizar una búsqueda más precisa, se deberán introducir los datos nuevos en los campos correspondientes del formulario de la Figura A.11 y pulsar el botón *Validar Datos*. Si no se validan los datos y se pulsa cualquiera de los botones de Búsqueda Simple o Búsqueda Profunda se eliminaran los datos anteriormente introducidos y se realizará la búsqueda con la información validada hasta ese momento. Destacar que esta búsqueda se corresponde con la búsqueda simple explicada a continuación.

- **Búsqueda Simple**: Esta búsqueda es como la que se ha explicado en los puntos anteriores. Realiza la búsqueda sobres los datos validados. Es decir, realiza la búsqueda con los datos introducidos en el formulario inicial o con los datos obtenidos del formulario avanzado una vez el usuario los haya validado. En caso de pulsar esta opción y no haber validado los datos se realizará la búsqueda borrando aquellos nuevos datos.

- **Búsqueda Profunda**: Esta búsqueda realiza una búsqueda sobre los datos validados al igual que en el caso anterior pero más profundamente. Es decir, realiza más combinaciones en los parámetros de entrada realizando una consulta más compleja. Hay que destacar que esta búsqueda al realizar más consultas sobre las fuentes también es más lenta, es decir, los resultados se muestran con mayor retardo. Además existe otro enlace en la parte inferior del formulario *Opciones de Búsqueda*, pulsando dicho enlace aparecen las opciones mostradas en la Figura A.12:

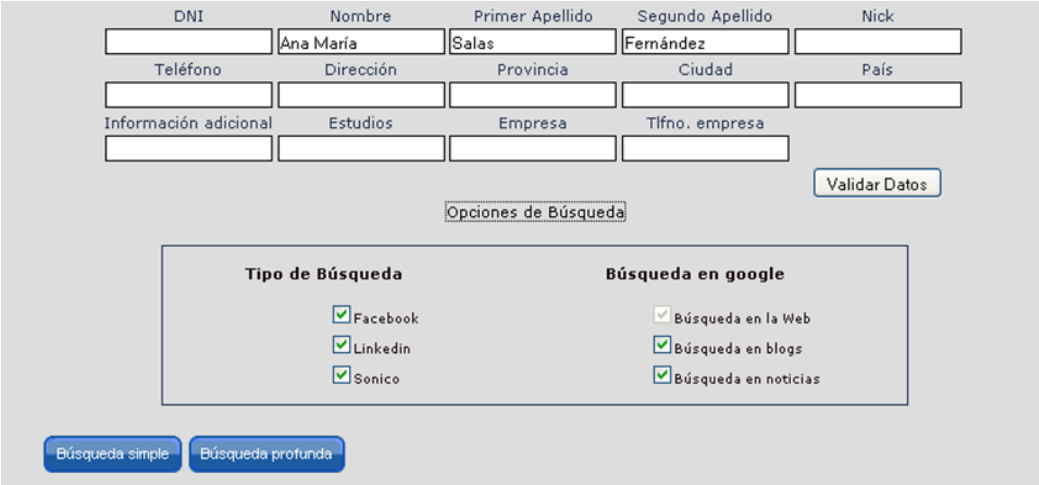


Figura A.12: Formulario de consolidación II

Este enlace, ofrece la oportunidad de perfilar más la búsqueda eligiendo las fuentes en las que se quiere buscar.

A.2.3. Búsqueda avanzada de personas

En la pantalla mostrada en la Figura A.1, se puede observar un enlace *Búsqueda Avanzada* si se pulsa dicho enlace redirige a la pantalla mostrada en la Figura A.13:

Figura A.13: Búsqueda Avanzada de Personas

Vemos que este formulario es igual al formulario de consolidación. Una vez seleccionemos el tipo de búsqueda profunda o simple y pinchando sobre el botón *Buscar* muestra los resultados de la misma forma que se ha comentado anteriormente.

Además, cabe destacar, que existe la opción de acceder a la búsqueda avanzada de empresas a través del menú situado a la izquierda de la pantalla.

A.2.4. Búsqueda normal de empresas

Para realizar la búsqueda normal de empresas, se pulsa sobre el enlace *Búsqueda de empresas* que tenemos en el menú de la izquierda mostrado en la Figura A.1. A continuación mostramos el formulario de búsqueda de empresas, Figura A.14:

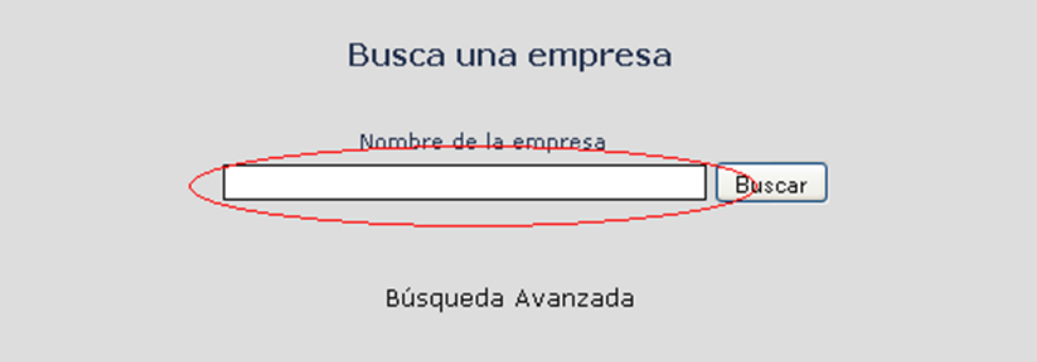


Figura A.14: Formulario de búsqueda de empresas

Una vez insertemos la empresa se producen los resultados que se enumeran a continuación.

A.2.4.1. RESULTADO EN WEB

A continuación, en la Figura A.15, se muestra el resultado de la búsqueda de empresas. En este primer caso se muestra el resultado Web.

Se observa como resultado, al igual que ocurría en la búsqueda de personas tenemos dos columnas. La primera de ellas muestra la validez del resultado y la segunda el resumen. Dicho resumen está formado por tres partes:

- Título: proporciona un enlace a la página que contiene datos sobre la empresa.
- Resumen: con el contenido de la página a la que enlaza el título.
- URL: URL principal de donde se obtiene la información.

Figura A.15: Resultado Web de empresas

A.2.4.2. RESULTADO EN BLOGS

Para acceder a esta opción, pinchamos en el enlace *Búsqueda en noticias*, así se mostrará la pantalla que muestra la Figura A.16. En este caso se muestra el resultado de la búsqueda en blogs. Al igual que en el caso anterior, tenemos dos columnas. La primera de ellas muestra la validez del resultado y la segunda el resumen. Dicho resumen está formado por tres partes:

- Título: proporciona un enlace al blog que contiene datos sobre la empresa.

- Resumen: con el contenido de la entrada del blog de la página a la que enlaza el título.

- URL: URL principal de donde se obtiene la información.

Validez ▼	Resumen
10	**Drupal: Cómo mostrar el submenú activo dentro de un nodo** 7 Sep 2010 Por Gabriel Cuesta Drupal es un CMS que separa por completo la gestión del contenido de los menús de navegación, esta manera de hacer las cosas implica que cuando mostramos el contenido de un nodo en pantalla no nos pone fácil mostrar además otros datos ... Mi pequeño rincón - Gabriel Cuesta - http://gabicuesta.blogspot.com/ gabicuesta.blogspot.com
10	**La Cartera Sanitaria pone en marcha Nodo de la Biblioteca Virtual ...** 19 Ago 2010 Por agonzalez Para la construcción de bibliografía chubutense de información en salud. Con la puesta en marcha de un nodo de la BVS Biblioteca Virtual en Salud, con base operativa en Esquel y una Jornada de capacitación en el manejo de tecnologías de ... Noticias de la Secretaría de Salud - http://organismos.chubut.gov.ar /salud_noticias/ organismos.chubut.gov.ar
10	**Regreso a Nuestro Origen** 29 Ago 2010 Por Nodo Bío Bío, RAP CHILE Con alegría os saludo, luego de terminar nuestra Reunión Oficial de Nodo BIO BIO, en Baktun 07, Silio 07, Luna Lunar, Año de la Luna Entonada Roja. Con éxito y eficacia, logramos designar los cargos, funciones y tareas a seguir, ... NODO BIO BIO - http://nodobiobio.blogspot.com/ nodobiobio.blogspot.com
10	**Implementan un nodo de la Biblioteca Virtual en Salud, en Esquel ...** 22 Ago 2010 Por Diana Rodríguez En la ciudad de Esquel, Chubut, con la puesta en marcha de un nodo de la Biblioteca Virtual en Salud (BVS) y con una jornada de alfabetización informacional y digital para los profesionales de la salud proviiciales, culminó "La Semana ... Corazón patagónico. - http://corazonpatagonico.blogspot.com/ corazonpatagonico.blogspot.com
10	**Hawala: descripción y funcionalidades** 17 Ago 2010 Por David de Ugarte El mecanismo es sencillo: una serie de nodos {1,2,3,...,N} implantados en distintos territorios, reconocen cada uno a una serie de apoderados. El nodo 1 reconocería por ejemplo {a1,b1,c1,...,n1}, cada uno con distintos límites de crédito y ... El Correo de las Indias - http://lasindias.coop/ lasindias.coop
10	**Nuevas novedades!** 9 Ago 2010 Por Eugenio "Tute" Costa Las tareas de Nodo Comunitario fueron aceptadas como proyecto de voluntariado universitario de la UNNOBA. Llevado al castellano esto significa que tendremos un presupuesto para cubrir los costos de las tareas y poder realizar tareas que ... Nodo Comunitario - http://www.nodocomunitario.com.ar/ www.nodocomunitario.com.ar

Figura A.16: Resultado en blogs de empresas

A.2.4.3. RESULTADO EN NOTICIAS

Para acceder a esta opción, pinchamos en el enlace *Búsqueda en noticias*, así se mostrará la pantalla que muestra la Figura A.17.

Figura A.17: Resultado en noticias de empresas

Por último, mostramos el resultado de la búsqueda en noticias. Al igual que ocurre en apartados anteriores tenemos dos columnas como resultado. La primera de ellas muestra la validez del resultado y la segunda el resumen. Dicho resumen está formado por tres partes:

- Título: proporciona un enlace al blog que contiene datos sobre la empresa.
- Resumen: con el contenido de la entrada del blog de la página a la que enlaza el título.
- URL: URL principal de donde se obtiene la información.

A.2.4.4. Resultado en Redes sociales

Por último, y al igual que para la búsqueda de personas, tenemos el resultado de redes sociales, que se muestra en la Figura A.18.

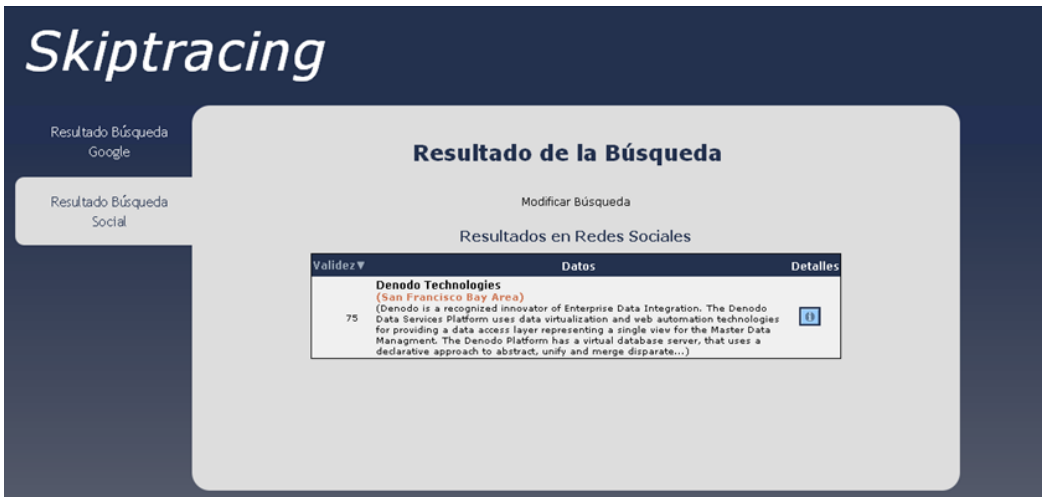


Figura A.18: Búsqueda en redes sociales y páginas amarillas

Se puede observar que la tabla está formada por tres columnas, al igual que en el caso de búsqueda de personas:

- Validez: Esta columna proporciona una puntuación indicando la confianza del resultado.

- Resumen: con un resumen del contenido de la información encontrada en la red social.

A.2.5. Búsqueda a través de fichero

Para realizar la carga masiva a través de un fichero se emplea el formulario mostrados en la figura A.19.

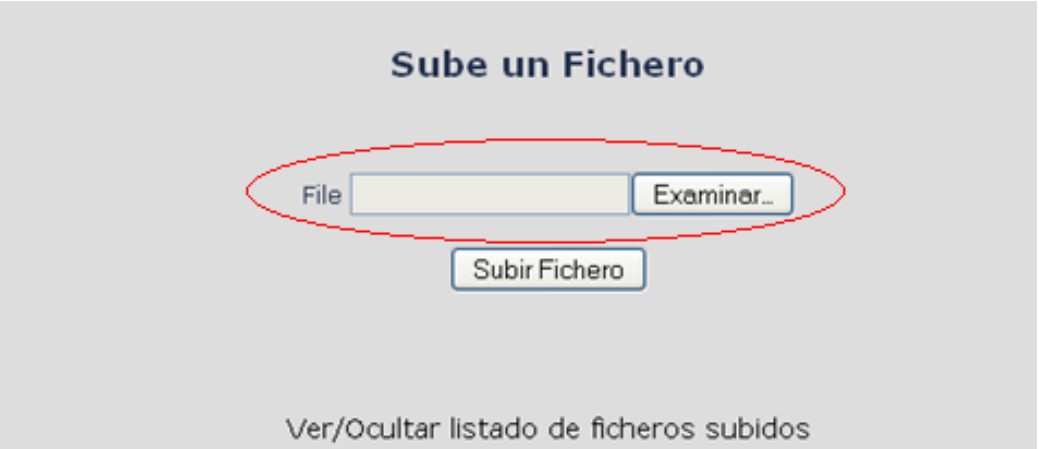


Figura A.19: Formulario de subida de ficheros

En este formulario tenemos que cargar un fichero con el siguiente formato:

dni;cuenta;direc;plaza;telef;nombre;apel1;apel2;tipo

En la A.20, mostramos un ejemplo de dicho fichero.

```
dni;cuenta;direc;plaza;telef;nombre;apel1;apel2;tipo
123456;1111-1111-11-1111111111;;A Coruña;;Daniel;Diaz;De la Iglesia;
12345;1111-1111-11-1111111111;;;;Juan Jose;Iglesias;Gonzalez;
123456789H;1111-1111-11-1111111111;c/ Real 20 1ºC;Madrid;;Alejandro;Gonzalez;Nuñez;
35566827V;1111-1111-11-1111111111;c/ Real 20 1ºC; Tui;asda;Fidel;Agrafojo;De Santiago;
12345;1111-1111-11-1111111111;;A Coruña;;José Antonio;Lamas;;
123456789H;1111-1111-11-1111111111;c/ Real 20 1ºC;;;Alejandro;González;Núñez;
876543260B;3333-2222-22-2222222222;c/ falsa 1 2º;Coruña;;Inmaculada;Domínguez;Mira;
70867802H;1111-1111-11-1111111111;c/ Real 20 1ºC;Madrid;;Javier;Diez;Gonzalez;
123456;1111-1111-11-1111111111;;A Coruña;;Marcos;López;Barbeito;
987654321A;2222-2222-22-2222222222;c/ falsa 1 2º;Pontevedra;;Alberto;Sanmartin;Camuñas;
876543210B;3333-2222-22-2222222222;c/ falsa 1 2º;Coruña;;Inmaculada;Domínguez;Mira;
12345;1111-1111-11-1111111111;;A Coruña;;Pedro;Moreira;;
70867805H;1111-1111-11-1111111111;c/ Real 20 1ºC;Madrid;;Javier;Diez;Gonzalez;
12345;1111-1111-11-1111111111;;Moaña;;María;Sousa;;
987654321A;2222-2222-22-2222222222;c/ falsa 1 2º;Pontevedra;;Alberto;Sanmartin;Camuñas;
876543210B;3333-2222-22-2222222222;c/ falsa 1 2º;Coruña;;Inmaculada;Domínguez;Mira
70867802H;1111-1111-11-1111111111;c/ Real 20 1ºC;Madrid;;Javier;Diez;Gonzalez
12345;1111-1111-11-1111111111;;A Coruña;;María;Lareo;
987654321A;2222-2222-22-2222222222;c/ falsa 1 2º;Pontevedra;;Alberto;Sanmartin;Camuñas
12345;1111-1111-11-1111111111;;A Coruña;;Alberto;Pan;Bermúdez
976543210B;3333-2222-22-2222222222;c/ falsa 1 2º;Coruña;;Inmaculada;Domínguez;Mira
12345;1111-1111-11-1111111111;;A Coruña;;Verónica;Fresco;Franco
70867802H;1111-1111-11-1111111111;c/ Real 20 1ºC;Madrid;;Javier;Diez;Gonzalez
12345;1111-1111-11-1111111111;;A Coruña;;Marcos;Muiño;García
987654321A;2222-2222-22-2222222222;c/ falsa 1 2º;Pontevedra;;Alberto;Sanmartin;Camuñas
```

Figura A.20: Fichero de carga.

Además, pinchando sobre la opción "Ver/Ocultar listado de ficheros subidos", se pueden ver los ficheros cargados:

Figura A.21: Listado de ficheros cargados

Si pinchamos sobre cualquiera de los ficheros cargados nos lleva a la página de resultados mostrada en la figura A.22

Figura A.22: Resultado de ficheros cargados

En la Figura A.22 se muestra una tabla con el resultado de la búsqueda. Existen dos resultados posibles:

- **Resultado positivo:** Indica que se ha encontrado actividad del usuario en la búsqueda de Redes sociales o de Actividad y que además el campo validez de dicha búsqueda es mayor de 45.
- **Resultado negativo:** No se ha encontrado actividad del usuario en la red o si se han encontrado resultados estos tienen el campo validez menor que 45. Si se pulsa sobre cualquier usuario, redirige a la página de resultados anteriormente explicada.

Si se pincha sobre una persona con resultado positivo, nos llevará a la página de resultados que se muestra al hacer una búsqueda normal, con la única diferencia de que la información de redes sociales se obtiene directamente de base de datos y se lanza en tiempo real la búsqueda de blogs, foros y noticias.

Si por el contrario se pincha sobre una persona con resultado negativo, al igual que en el caso anterior, nos llevará a la página de resultados que se muestra al hacer la búsqueda normal. En este caso no habrá resultados para la búsqueda en redes sociales y se lanza en tiempo real la búsqueda de blogs, foros y noticias.

Apéndice B

Ficheros de configuración

B.1. Estructura del fichero de configuración

Los ficheros de configuración poseen la siguiente estructura:

- Configuración VDP: Todos los ficheros de configuración contienen los parámetros.
  - vdpDs.queryTimeout: Tiempo de espera para obtener los resultados al realizar la consulta a VDP. Si tarda más se lanzará un Time Out.
  - vdpDs.chunkTimeout: Tiempo que tarda el procedimiento almacenado en realizar preguntas a las vistas de VDP para ver si poseen resultados.
  - vdpDs.chunkSize: En caso de que la vista haya devuelto en número de resultados que marca este campo, serán devueltos al procedimiento almacenado.

  Estos parámetros configuran la Query que vamos a realizar a VDP. No suelen modificarse.
- Configuración de la clase Processor:
  - search.processor.instanceClassName: Este parámetro sirve para indicar la clase java que utilizará los datos de los parámetros de este fichero. Este valor no se suele modificar de un fichero a otro.
- Configuración de la Query
  - search.processor.deep: Indica el número de querys que el Processor tiene que buscar en el fichero de configuración.

- search.output.fields.metadata: indica los campos de salida de las querys a realizar. Este campo es común en todas las consultas de la aplicación para unificar los campos de salida, ya que al realizar diferentes querys tenemos diferentes campos de salida.
- search.deep.1.query: En este parámetro se escribe la query que queremos realizar, por ejemplo: SELECT * FROM FINAL_GOOGLE_BY_KEYWORDS WHERE KEYWORDS = @keys@. Siendo el valor keys el valor del parámetro siguiente. El nombre del parámetro contiene un número que indica el número de query. Tendremos tantas querys como indique el parámetro search.processor.deep.
- search.deep.1.parameter.keys: Este parámetro indica los valores que contendrá el valor keys contenido en la consulta del apartado anterior. Al igual que en el parámetro anterior, el nombre contiene un número que indica a qué consulta pertenecen los parámetros.

- Configuración del procesamiento de los resultados:

  - search.normalizer.instanceClassName: Este parámetro indica la clase java que procesará los resultados con los siguientes parámetros.
  - loc.denodo.skiptracing.normalizer.MinimizeNormalizer.discard: Este parámetro determina si se descarta o no un resultado en función del valor de confianza. Es decir, al realizar la búsqueda se procesa cada resultado para darle un valor que representará su validez, si este valor supera el valor que contiene este parámetro se aceptará, en caso contrario se rechazará el resultado.
  - loc.denodo.skiptracing.normalizer.MinimizeNormalizer.global.compare.1.function.0: Este parámetro indica la función que va procesar los resultados. En este caso tenemos dos números incluidos en el nombre, el primero indica la query a la que pertenece y el segundo el número de función. Para su ejecución empleará los parámetros que se explican a continuación.
  - loc.denodo.skiptracing.normalizer.MinimizeNormalizer.global.compare.dupl.threshold: Indica el número de columnas que deben coincidir entre dos resultados para considerarlos duplicados.
  - loc.denodo.skiptracing.normalizer.MinimizeNormalizer.global.compare.dupl.keyfields: Los campos contenidos en este parámetro son considerados cuando tenemos dos resultados con el mismo valor en ellos. En este caso se marcan como duplicados independientemente de si el número de campos iguales es inferior al parámetro anterior.
  - loc.denodo.skiptracing.normalizer.MinimizeNormalizer.row.reliability.1: El valor que contiene este parámetro se usa para saber si el resultado obtenido se rechaza o se muestra como parte de la tabla de

resultados. El número que contiene en el nombre está referido a una de las querys definidas anteriormente en el documento.

- loc.denodo.skiptracing.normalizer.MinimizeNormalizer.row.compare.1.function.0: Este parámetro contiene la función que realizará la comparación entre el parámetro de la derecha y el de la izquierda. Por ejemplo: out.ST_ACTIVITY_TITLE contains in.FIRSTNAME, en este caso, se llama a la función contains, que mira si el parámetro FIRSTNAME de entrada está contenido en el parámetro ST_ACTIVITY_TITLE contenido en el resultado de la búsqueda. Este parámetro contiene dos números en su nombre, el primero indica la query anteriormente definida y el segundo es un identificador de la función dentro del fichero .properties. En los parámetros de este tipo tenemos las siguientes funciones implementadas:
  - contains: Esta función comprueba si el parámetro de la derecha está contenido en el parámetro de la izquierda.
  - contains_ig: Esta función comprueba si el parámetro de la derecha está contenido en el parámetro de la izquierda, sin tener en cuenta las mayúsculas ni minúsculas.
  - Notnull: Esta función solo tiene el parámetro de la izquierda, y comprueba que este no sea nulo.
  - Eq: Esta función comprueba si el parámetro de la derecha es igual que el parámetro de la izquierda.
  - Eq_ig: Esta función comprueba si el parámetro de la derecha es igual que el parámetro de la izquierda, sin tener en cuenta las mayúsculas ni minúsculas.

- loc.denodo.skiptracing.normalizer.MinimizeNormalizer.row.compare.1.value.0: Este parámetro contiene un valor que se obtendrá en caso de que se cumpla la función anterior. El valor contenido en los parámetros de este tipo se irá sumando hasta conseguir un valor que indicará la validez del resultado. Vemos que en el nombre del parámetro actual, tenemos dos números, ambos corresponden a la query definida anteriormente y a la función anterior

B.2. Ficheros de configuración contenidos en el proyecto

Los ficheros de configuración que emplea el procedimiento almacenado son:

- Query-google.properties: Contiene los parámetros para realizar y procesar la búsqueda en la página del buscador Google.

- Query-google-deep.properties: Contiene los parámetros para realizar y procesar la búsqueda, de forma profunda, en la página del buscador Google.
- Query-google-blogs-deep.properties: Contiene los parámetros para realizar y procesar la búsqueda, de forma profunda, en la página del buscador Google Blogs.
- Query-google-news-deep.properties: Contiene los parámetros para realizar y procesar la búsqueda, de forma profunda, en la página del buscador Google.
- Query-google-blogs.properties: Contiene los parámetros para realizar y procesar la búsqueda en la página del buscador Google Blogs.
- Query-google-news.properties: Contiene los parámetros para realizar y procesar la búsqueda en la página del buscador Google Noticias.
- Query-social.properties: Contiene los parámetros para realizar y procesar la búsqueda de personas en redes sociales.
- Query-social-deep.properties: Contiene los parámetros para realizar y procesar la búsqueda, de forma Profunda, en redes sociales.
- Query-comp-google-blogs.properties: Contiene los parámetros para realizar y procesar la búsqueda en la página del buscador Google Blogs para la búsqueda de empresas.
- Query-comp-google-blogs-deep.properties: Contiene los parámetros para realizar y procesar la búsqueda en la página del buscador Google Blogs, de forma profunda, para la búsqueda de empresas.
- Query-comp-google-news.properties: Contiene los parámetros para realizar y procesar la búsqueda en la página del buscador Google Noticias para la búsqueda de empresas.
- Query-comp-google-news-deep.properties: Contiene los parámetros para realizar y procesar la búsqueda en la página del buscador Google Noticias, de forma profunda, para la búsqueda de empresas.
- Query-comp-social.properties: Contiene los parámetros para realizar y procesar la búsqueda en redes sociales para la búsqueda de empresas.
- Query-comp-social-deep.properties: Contiene los parámetros para realizar y procesar la búsqueda, de forma Profunda, en redes sociales para la búsqueda de empresas.

Apéndice C

Instrucciones de despliegue

A continuación se enumera los distintos pasos a seguir para desplegar cada uno de los módulos:

C.1. Desplegar Wrapper en ITP

En el repositorio cvs este módulo se denomina skiptracing-itp-wrappers. Para desplegar los Wrappers de ITP se realizan los siguientes pasos:

1. Se copian los ficheros xml de cada uno de los Wrapper en la ruta Denodo Platform/metadata/itp-admin-tool/nombreProyecto.

2. Se abre la aplicación Wrapper Generator Tool de Denodo Platform.

3. Se pulsa con el botón derecho sobre el proyecto donde se han almacenado los ficheros xml guardados y se selecciona la opción Add processes, tal y como se muestra en la Figura C.1:

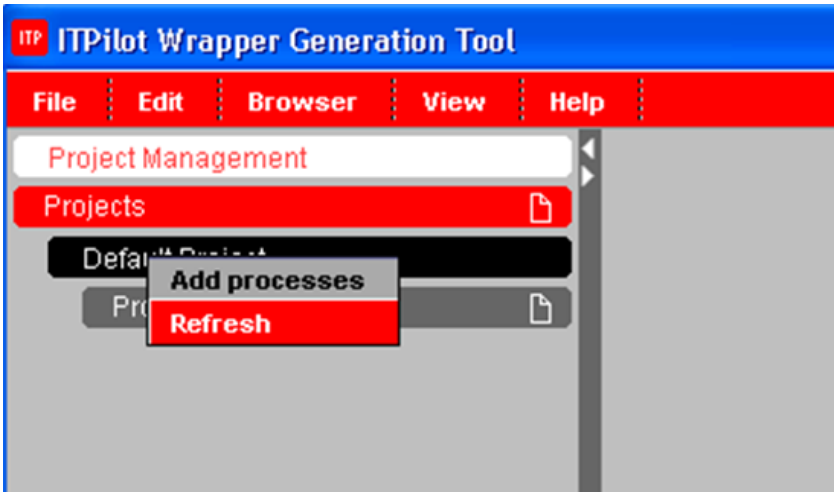


Figura C.1: Pantalla para añadir Wrapper a ITP

4. Una vez realizado el paso anterior se seleccionan los Wrapper que queremos importar al proyecto. Tal y como muestra la Figura C.2:

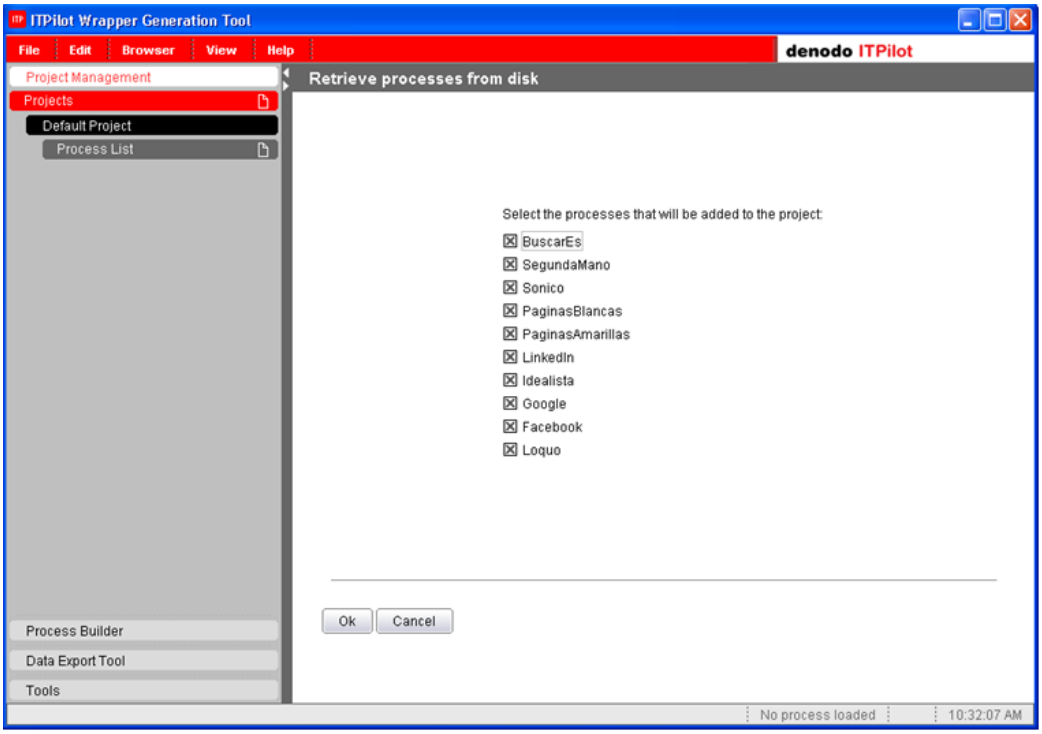


Figura C.2: Pantalla para añadir Wrapper a ITP

5. Se pulsa el botón OK añadiendo los wrappers a la herramienta.

6. Ahora ya es posible ejecutarlos y editarlos. A continuación se muestra una figura de los Wrapper añadidos a la herramienta.

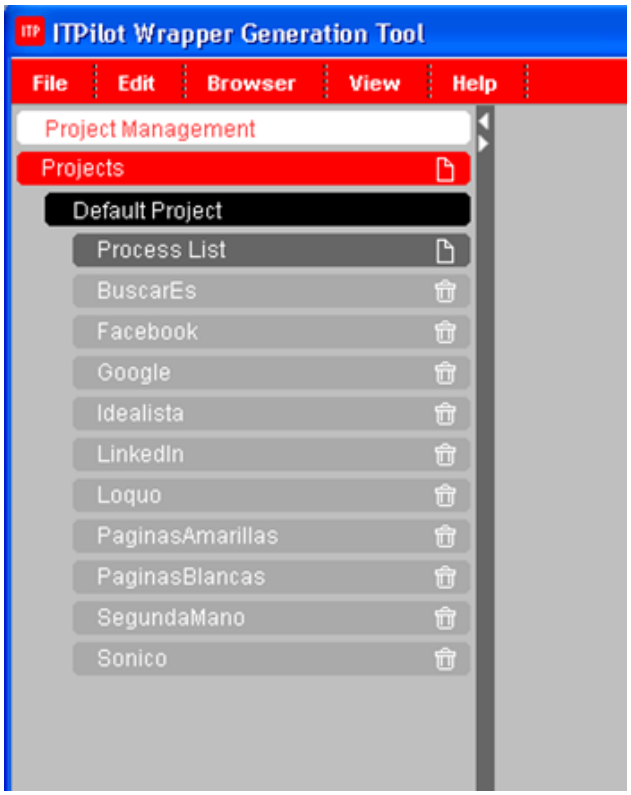


Figura C.3: Wrappers en ITP

En caso de realizar alguna modificación del Wrapper se debe desplegar en VDP para poder crear las vistas necesarias. Para ello, desde ITP, se puede desplegar el Wrapper modificado en VDP. Para ello seguiremos los siguientes pasos:

7. Seleccionamos el wrapper que queremos desplegar, abriéndolo en la herramienta.

8. Pulsamos la opción ”Data export Tool”de la herramienta.

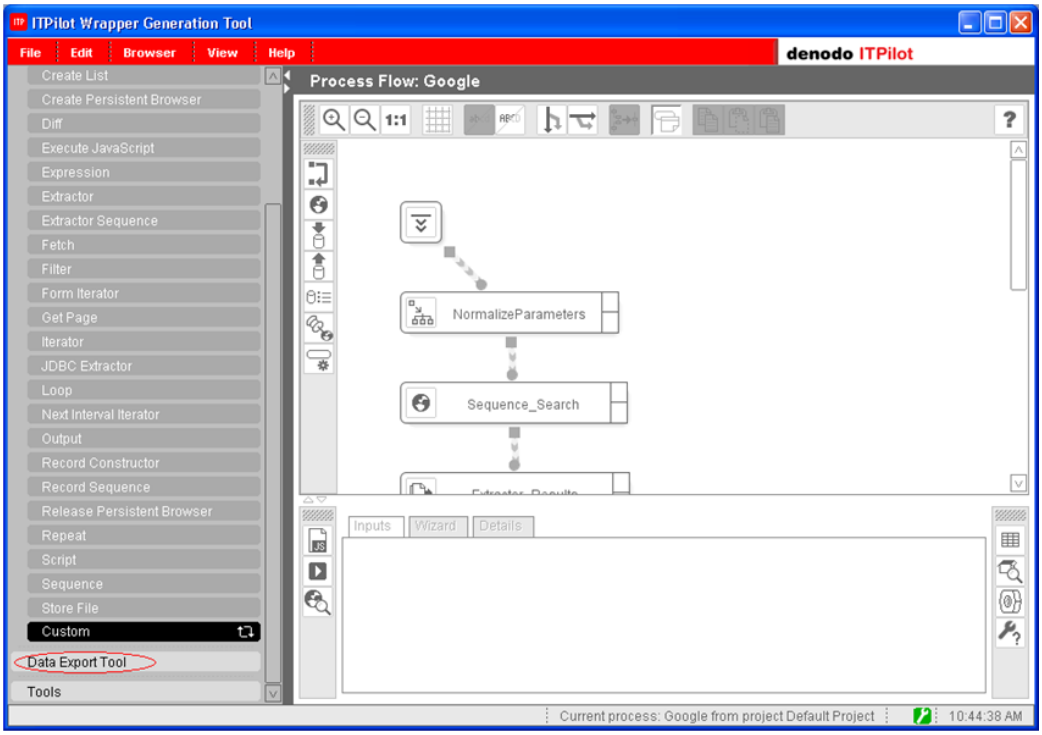


Figura C.4: Pantalla ITP

9. Seleccionamos la opción, ”Server Deploy”. Dentro de dicha opción, se introduce el nombre deseado. Si el Wrapper ya estaba desplegado previamente, es necesario seleccionar la opción, Replace Wrapper if it already exists”. Se introducen los datos relativos al servidor VDP y se pulsa el botón OK.

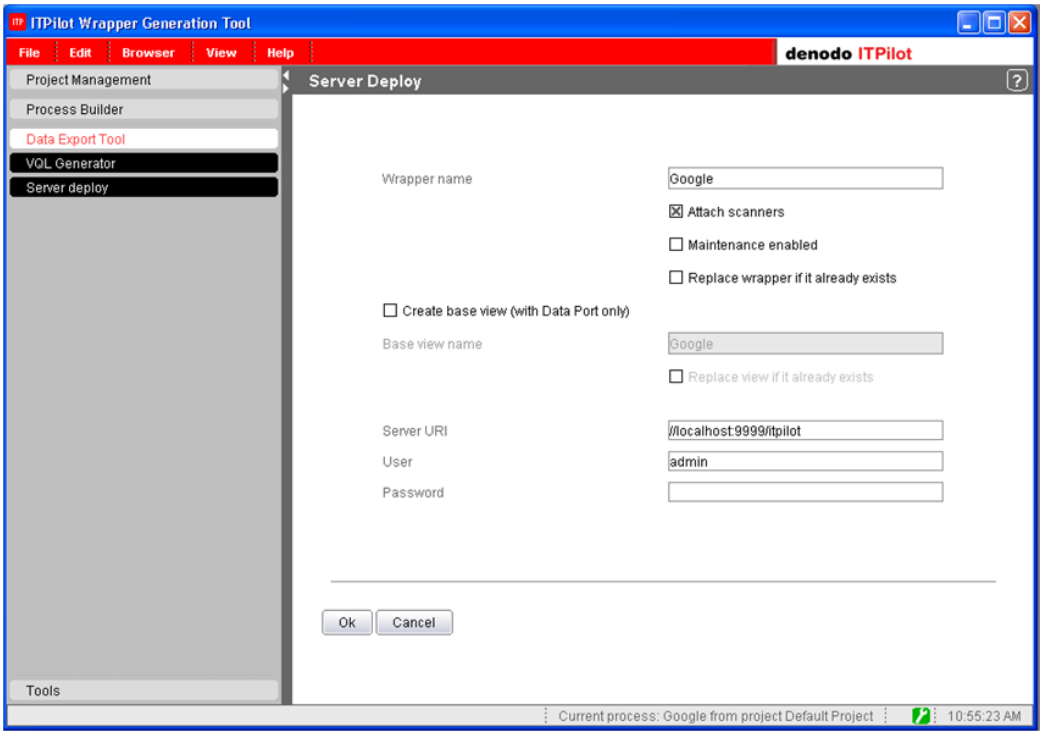


Figura C.5: Pantalla de despliegue de ITP

C.2. Desplegar los Jobs del scheduler

En el repositorio cvs este módulo se denomina skiptracing-scheduler. Para desplegar este módulo se realizan los siguientes pasos:

1. Se arranca el servidor y la herramienta de administración del Scheduler.

2. Nos conectamos a la página de la herramienta de administración a través de la dirección http://localhost:9090/webadmin/denodo-scheduler-admin/ShowAuthentication.ajax.

3. Realizamos la autenticación en la página anterior.

4. Una vez realizada la autenticación. Se selecciona la pestaña Configuration situada en la parte derecha superior de la ventana, tal y como se muestra en la Figura C.6:

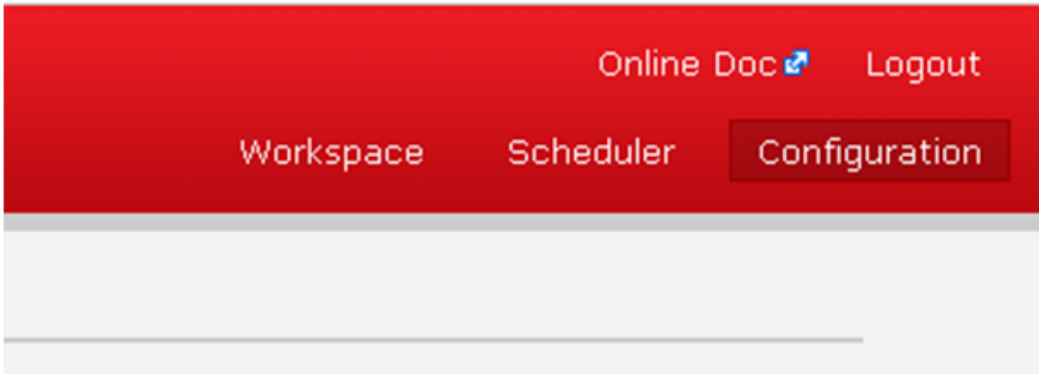


Figura C.6: Opciones del Scheduler

5. A continuación se pulsa la opción import.

6. Una vez seleccionada adjuntamos el fichero .zip generado por este módulo tal como muestra la Figura C.7:

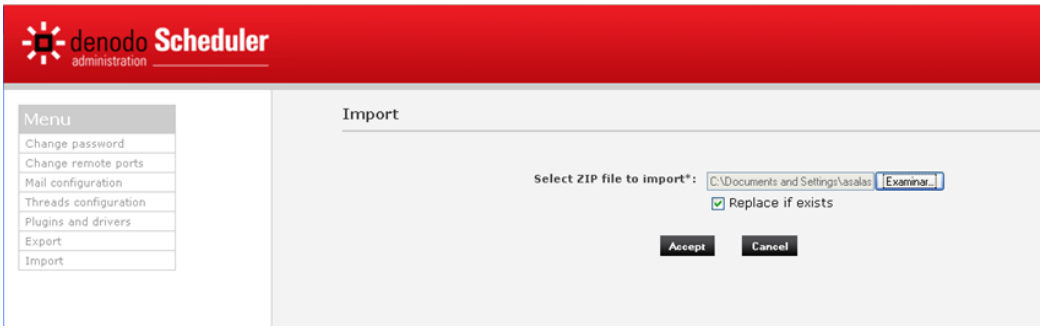


Figura C.7: Pantalla para importar en Scheduler

7. Una vez completado este paso correctamente, en la pestaña de Workspace, dentro del proyecto Test, aparecerá la pantalla que muestra la Figura C.8:

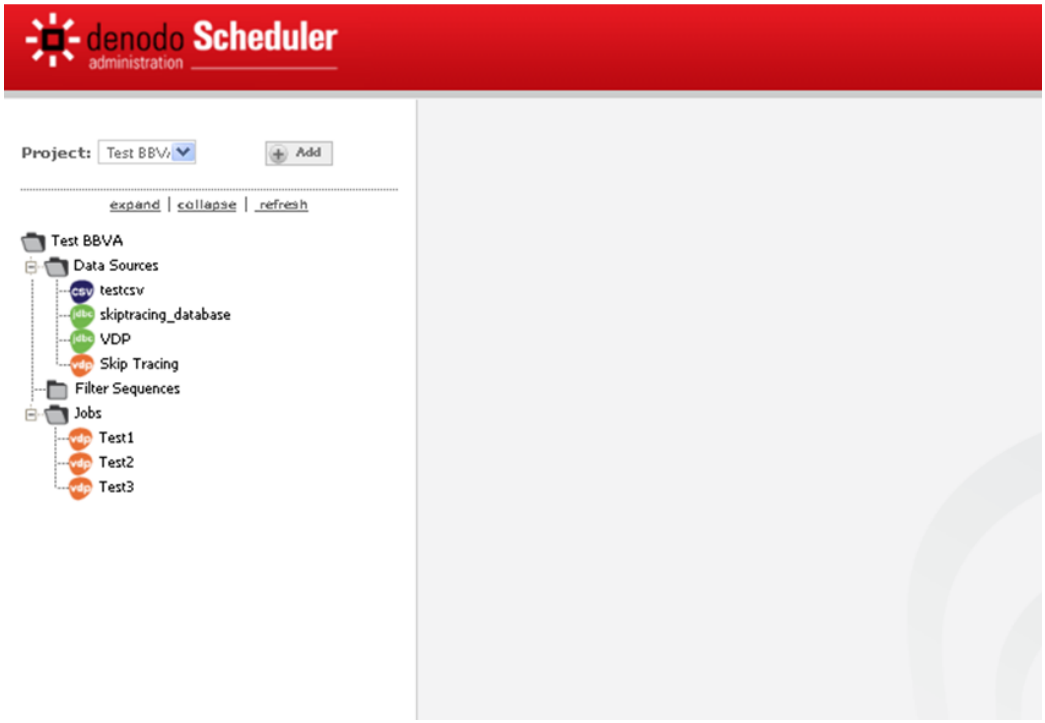


Figura C.8: Pantalla Workspace Scheduler

C.3. Despliegue del procedimiento almacenado en VDP

En el repositorio cvs este módulo se denomina skiptracing-proc.

Para desplegar el procedimiento almacenado tenemos que copiar el documento .jar generado por dicho módulo. Este .jar debe almacenarse en la ruta Denodo Platform/extensions/thirdparty/lib.

C.4. Despliegue del Exporter

En el repositorio cvs este módulo se denomina skiptracing-exporter.

Este módulo genera el documento skiptracing-exporter-1.0-jar-with-dependencies.jar. Para desplegar este módulo hay que copiar el fichero anterior en la ruta Denodo Platform/extensions/thirdparty/lib.

Para que el exporter funcione correctamente hay que copiar el fichero SkiptracingExporter.xml que se encuentra en el directorio src/main/resources/META-INF en el directorio de la plataforma Denodo metadata/scheduler/elements/exporters.

Además para el correcto funcionamiento de la aplicación es necesario adjuntar el fichero .jar del driver jdbc. Para ello se arranca el servidor del scheduler (Server) y su herramienta de administración (Administration Tool). Una vez realizada la conexión con la herramienta de administración, se pulsa la pestaña Configuration y la opción Plugins and Drivers, mostrando la pantalla que muestra la Figura C.9:

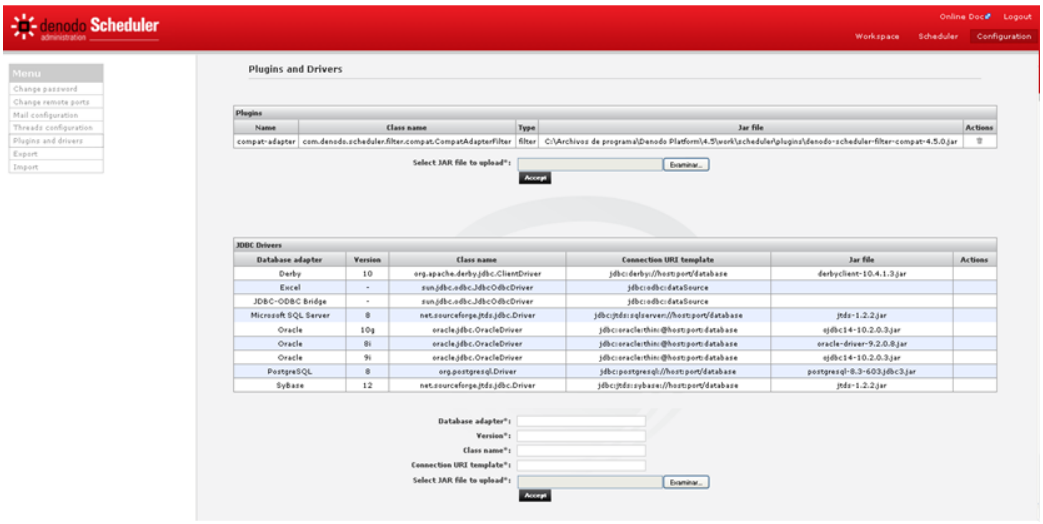


Figura C.9: Pantalla Plugins and Drivers del Scheduler

Hay que rellenar el formulario que muestra la Figura C.10:

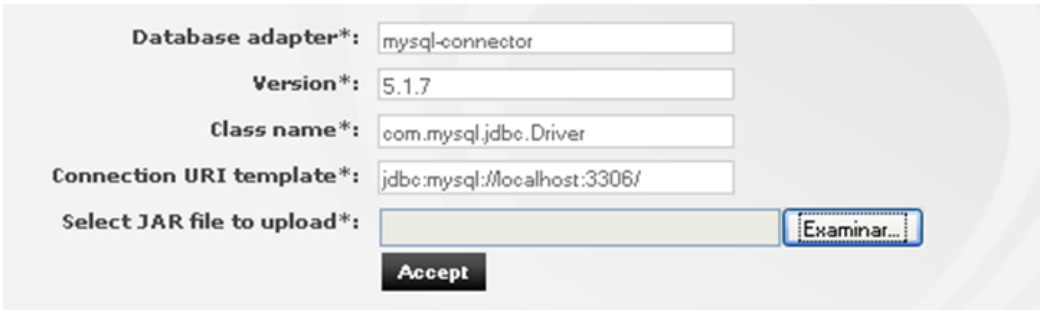


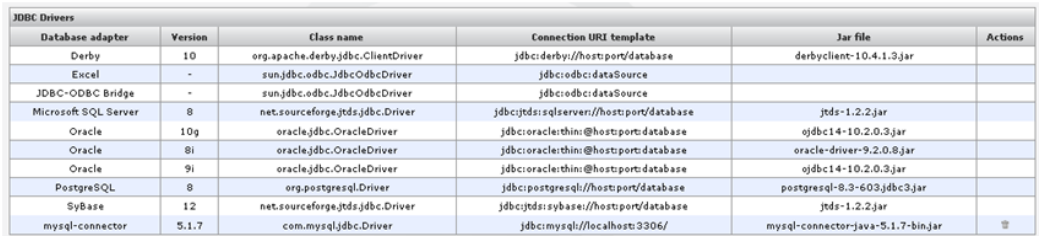
Figura C.10: Formulario de inserción de Drivers del Scheduler

A continuación se rellenan los datos de la siguiente forma:

- Database adapter: mysql-connector
- Version: 5.1.7

- Class name: com.mysql.jdbc.Driver
- Connection URI template: jdbc:mysql://localhost:3306/bbdd
- Select JAR file to upload: Seleccionamos el fichero .jar del driver.

Una vez se pulsa el botón Accept la tabla de la figura C.9 queda como muestra la Figura C.11:



JDBC Drivers

Database adapter	Version	Class name	Connection URI template	Jar file	Actions
Derby	10	org.apache.derby.jdbc.ClientDriver	jdbc:derby://host:port/database	derbyclient-10.4.1.3.jar	
Excel	-	sun.jdbc.odbc.JdbcOdbcDriver	jdbc:odbc:dataSource		
JDBC-ODBC Bridge	-	sun.jdbc.odbc.JdbcOdbcDriver	jdbc:odbc:dataSource		
Microsoft SQL Server	8	net.sourceforge.jtds.jdbc.Driver	jdbc:jtds:sqlserver://host:port/database	jtds-1.2.2.jar	
Oracle	10g	oracle.jdbc.OracleDriver	jdbc:oracle:thin:@host:port:database	ojdbc14-10.2.0.3.jar	
Oracle	8i	oracle.jdbc.OracleDriver	jdbc:oracle:thin:@host:port:database	oracle-driver-9.2.0.8.jar	
Oracle	9i	oracle.jdbc.OracleDriver	jdbc:oracle:thin:@host:port:database	ojdbc14-10.2.0.3.jar	
PostgreSQL	8	org.postgresql.Driver	jdbc:postgresql://host:port/database	postgresql-8.3-603.jdbc3.jar	
SyBase	12	net.sourceforge.jtds.jdbc.Driver	jdbc:jtds:sybase://host:port/database	jtds-1.2.2.jar	
mysql-connector	5.1.7	com.mysql.jdbc.Driver	jdbc:mysql://localhost:3306/	mysql-connector-java-5.1.7-bin.jar	

Figura C.11: Drivers adjuntados del Scheduler

Para que el exporter funcione correctamente hay que configurar el acceso a la base de datos con la que se va a trabajar. Para ello se inserta en el fichero skiptracing.business.properties, la información relativa a la base de datos.

Nota: Para que funcione correctamente el acceso del exporter a la base de datos hay que copiar el .jar del driver jdbc en la carpeta Denodo Platform/extensions/thirdparty/lib.

C.5. Despliegue de vistas en VDP

En el repositorio cvs este módulo se denomina skiptracing-vdp-vql.

Para desplegar este módulo hay que arrancar el servidor VDP (Server) y su herramienta de administración (Administration Tool).

Una vez abierta la herramienta Administration Tool, se realiza la conexión a Skip_tracing y se selecciona la opción VQL Shell.

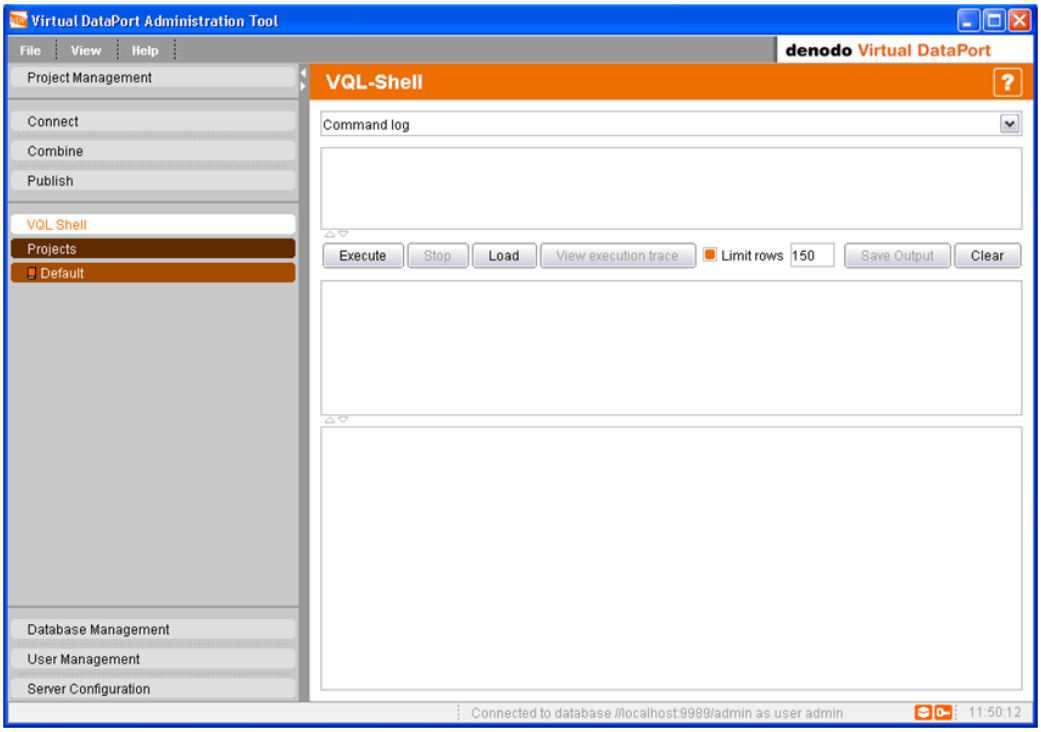


Figura C.12: Pantalla VDP

Una vez seleccionada, pulsamos el botón Load y seleccionamos los ficheros skiptracing-model y skiptracing_upload_csv.

Una vez se carguen ambos ficheros tendremos generadas todas las Vistas de la aplicación.

Para que las vistas CSV funcionen correctamente se tienen que colocar los documentos CSV en el directorio apropiado. El directorio que viene en el fichero vql es: C:/Archivos de programa/DenodoPlatform/metadata/itp-admin-tool/SkipTracing/

Se debe modificar dicho directorio del fichero skiptracing-model si la ruta es diferente o queremos almacenar la información en otro fichero. Los documentos csv son:

- **input_data.csv**: Este fichero csv sirve para guardar la dirección de inserción a través de la carga de un fichero. Se almacena como fichero csv para facilitar su procesado.
- **pblancas_provinces.csv**: Es un fichero csv que almacena todas las provincias. Este fichero se utiliza para compararlas con las que el usuario introduce y obtener el nombre correcto de la provincia. Por ejemplo, si se introduce Islas Baleares nos quedaremos con Illes Balears.

Nota: Es posible que falle la conexión a la base de datos a través de VDP, en este caso habría que seguir los siguientes pasos:

1. Abrir la aplicación VDP Administration Tool y entrar con el usuario admin.

2. Pinchar en la opción connect>JDBC>skiptr

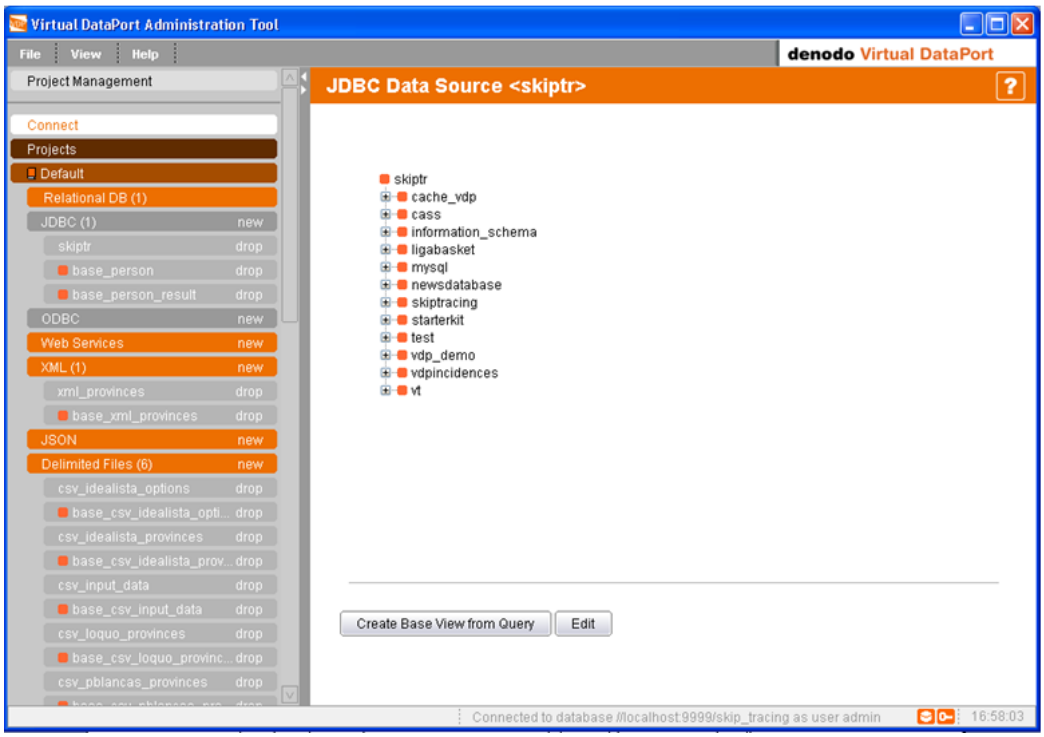


Figura C.13: Opción Connect VDP

3. Pulsar la tecla edit.

4. En la opción Driver JAR file añadir el driver JDBC.

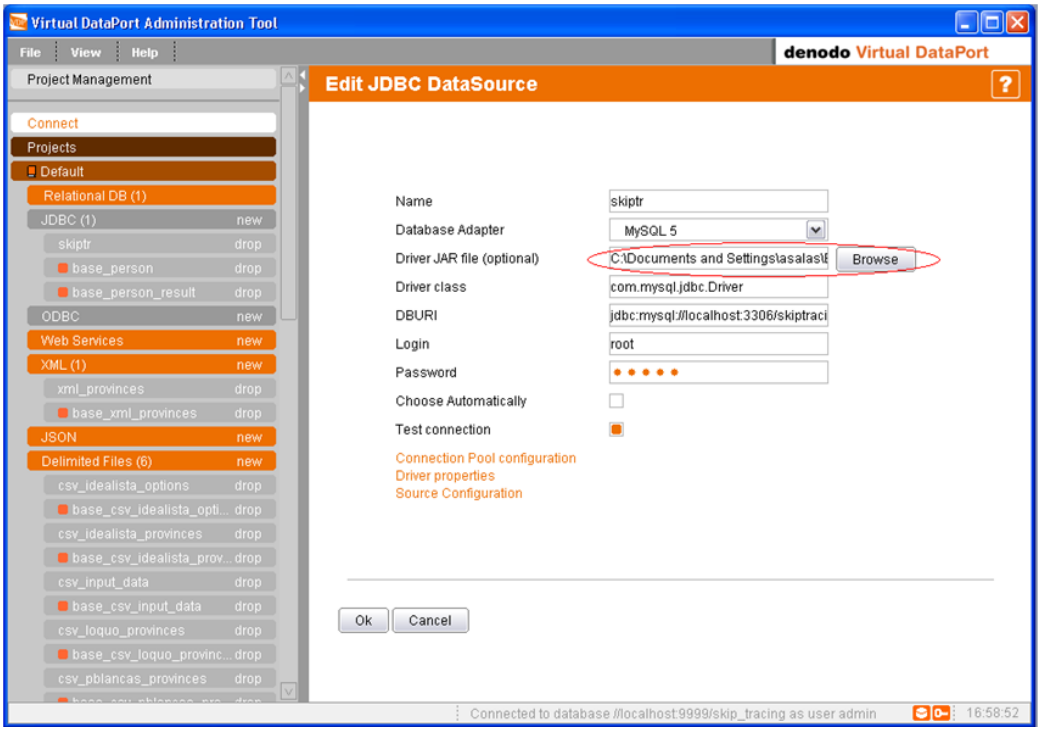


Figura C.14: Pantalla Edit JDBC

C.6. Despliegue de la aplicación web

En el repositorio cvs este módulo se denomina skiptracing-web.

Para desplegar la aplicación WEB se tiene que configurar inicialmente la base de datos. Para ello se debe ejecutar el fichero skiptracing-consolidated.sql.

Una vez se tenga configurada la base de datos, se debe generar el fichero .war que será colocado en la carpeta WEB-INF del servidor tomcat (o directamente ejecutado desde eclipse).

Apéndice D

Análisis de fuentes: Vistas intermedias

D.0.0.1. View_google

La vista derivada view_google es una proyección de la vista base base_itp_googe , como muestra el siguiente diagrama:

Figura D.1: Tree view view_google

Los campos de dicha vista son:

VIEW_GOOGLE	
G_URL	text
G_SUMMARY	text
G_SOURCE	text
G_TITLE	text
KEYWORDS	text
SOURCES	text

Figura D.2: Campos vista view_google

D.0.0.2. View_facebook

La vista derivada view_facebook, es una unión de la vistas base base_itp_facebook y base_itp_facebookgoogle.

Figura D.3: Campos vista view_facebook

Los campos de dicha vista son:

VIEW_FACEBOOK	
FB_COUNTRY	text
FB_NAME_DETAILS	text
FB_NICK	text
FB_IMAGE_URL	text
FB_NAME	text
FB_FRIENDS	itp_facebook_fb_friends
FB_FRIEND_NAME	text
FIRSTNAME	text
LASTNAME	text
FB_DETAIL_URL	text
FB_REDES	text
FB_TYPE	text
SOURCE	text

Figura D.4: Campos vista view_facebook

D.0.0.3. View_company

La vista derivada view_company, es una proyección de la vista base base_itp_pamarillas.

Figura D.5: Treeview base_itp_pamarillas

Los campos de dicha vista son:

VIEW_COMPANY	
COM_COMPANY	text
COM_COMPANY_DESCRIPTION	text
COM_COMPANY_WEB	text
COM_COMPANY_ADDRESS	text
COM_PHONE_LIST	itp_pamarillas_pa_phones_list
PA_PHONE	text
NAME	text
ACTIVITY	text
PROVINCE	text
CITY	text

Figura D.6: Campos de la vista view_company

D.0.0.4. Final_company_by_name

La vista derivada final_company_by_name, es una proyección de la vista derivada view_company.

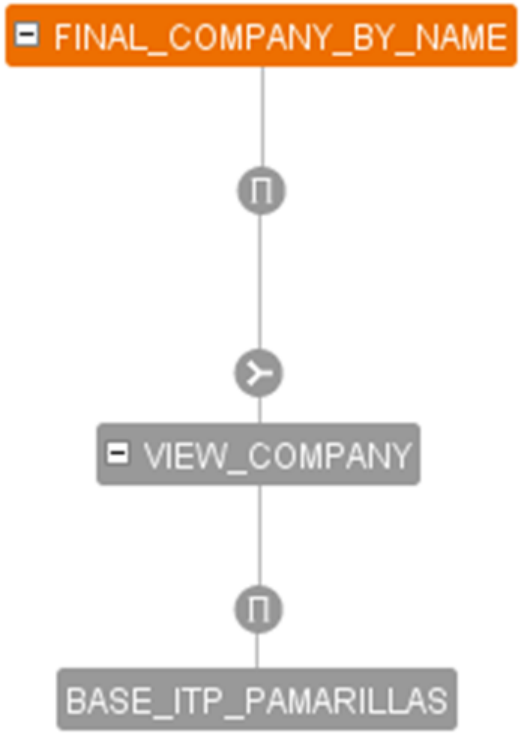


Figura D.7: Treeview Final_company_by_name

Los campos de dicha vista son:

FINAL_COMPANY_BY_NAME	
ST_COMPANY	text
ST_COMPANY_DESCRIPTION	text
ST_COMPANY_WEB	text
ST_COMPANY_ADDRESS	text
ST_COMPANY_PHONE	text
NAME	text
ACTIVITY	text
PROVINCE	text
CITY	text

Figura D.8: Campos de la vista final_company_by_name

A continuación se explican más detalladamente los campos de la Tabla 31:

- ST_COMPANY: Nombre de la empresa.
- ST_COMPANY_DESCRIPTION: Descripción de la empresa.

- ST_COMPANY_WEB: Página Web de la empresa.
- ST_COMPANY_ADDRESS: Dirección de la empresa.
- ST_COMPANY_PHONE: Teléfono de la empresa.
- NAME: Nombre de la empresa.
- ACTIVITY: Actividad a la que se dedica la empresa.
- PROVINCE: Provincia de la empresa.
- CITY: Ciudad de la empresa.

D.0.0.5. Inter_xml_provinces

La vista derivada inter_xml_provinces, es una proyección de la vista base base_xml_provinces.

Figura D.9: Treeview inter_xml_provinces

Los campos de dicha vista son:

INTER_XML_PROVINCES	
PARAM	text
ARRAY_ANYTYPE	xml_provinces_nombres_array_an...
ANYTYPE	text

Figura D.10: Campos de la vista inter_xml_provinces

D.0.0.6. Inter_xml2_provinces

La vista derivada inter_xml2_provinces, es una proyección de la vista inter_xml_provinces (47). En esta nueva vista se emplea el operador Flattern. Esta nueva vista elimina el campo array de la vista inter_xml_provinces y creamos una tupla por cada elemento del array que teníamos antes, es decir, en lugar de tener una sola entrada con un array, tenemos varios resultados teniendo en el campo ANYTYPE un único campo de texto.

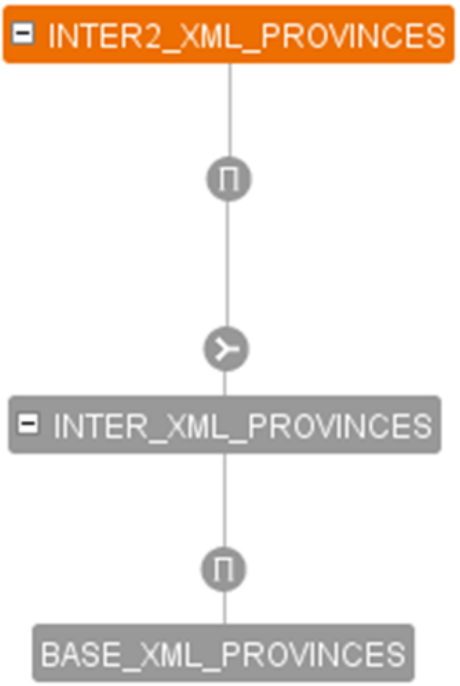


Figura D.11: Treeview inter2_xml_provinces

Los campos de la nueva vista son:

INTER2_XML_PROVINCES	
PARAM	text
ANYTYPE	text

Figura D.12: Campos de la vista inter2_xml_provinces

D.0.0.7. Final_xml_provinces

La vista derivada final_xml_provinces, es una proyección de la vista inter_2xml_provinces (48).

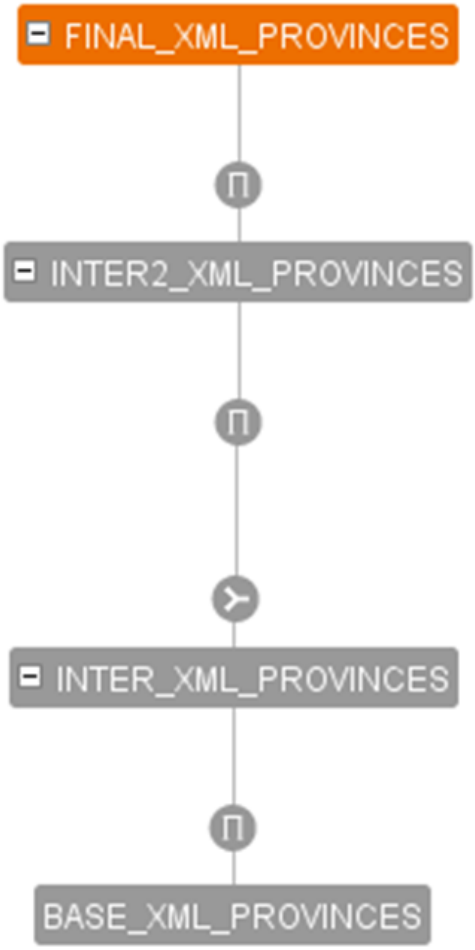


Figura D.13: Treeview final_xml_provinces

Los nuevos campos de la vista son:

FINAL_XML_PROVINCES	
PARAM	text
PROVINCIA	text

Figura D.14: Campos de la vista final_xml_provinces

D.0.0.8. Final_xml_distinct_province

La vista derivada final_xml_distinct_province, es una proyección de la vista final_xml_province (49).

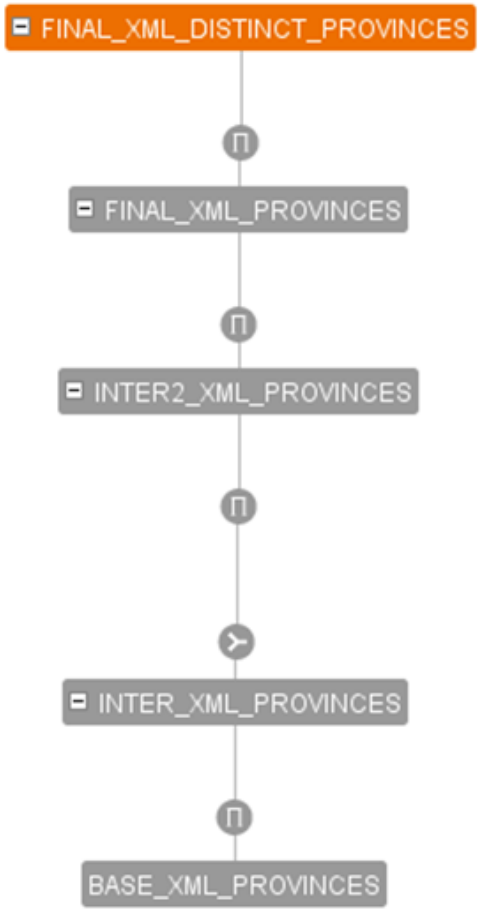


Figura D.15: Treeview final_xml_distinct_province

Los campos de la nueva tabla son:

FINAL_XML_DISTINCT_PROVINCES	
PROVINCIA	text
PARAM	text

Figura D.16: Campos de la vista final_xml_distinct_provinces

D.0.0.9. Inter_csv_task_input_nonull

La vista derivada inter_csv_task_input_nonull, es una selección de la vista base base_csv_task_input (42).

Figura D.17: Treeview inter_csv_task_input_nonull

Los campos de la nueva vista son:

INTER_CSV_TASK_INPUT_NONULL	
NIF	text
CUENTA	text
DIREC	text
PLAZA	text
TELEF	text
NOMBRE	text
APEL1	text
APEL2	text
TIPO	text
FILENAME	text

Figura D.18: Campos vista Campos de la vista inter_csv_task_input_nonull

D.0.0.10. Inter_csv_distinct_provinces

La vista derivada inter_csv_distinct_provinces, es una unión a través de una condición join de las vistas inter_csv_task_input_nonull (51) y final_xml_distinct_provinces

(50).

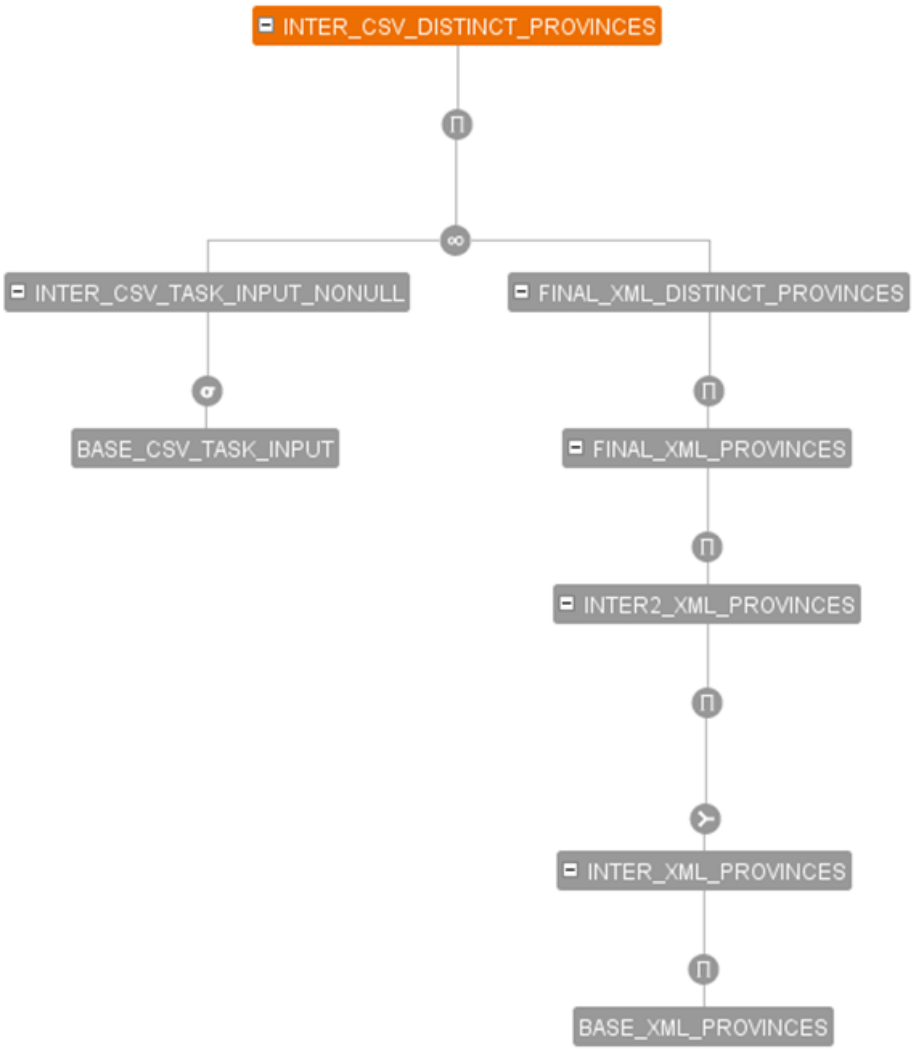


Figura D.19: Treeview inter_csv_distinct_provinces

Los campos de la nueva tabla son:

INTER_CSV_DISTINCT_PROVINCES	
NIF	text
CUENTA	text
DIREC	text
PLAZA	text
TELEF	text
NOMBRE	text
APEL1	text
APEL2	text
TIPO	text
FILENAME	text
PROVINCIA	text
PARAM	text

Figura D.20: Campos de la vista inter_csv_distinct_provinces

D.0.0.11. Inter_csv_task_input_null

La vista derivada inter_csv_task_input_null, es una selección de la vista base base_csv_task_input (42).

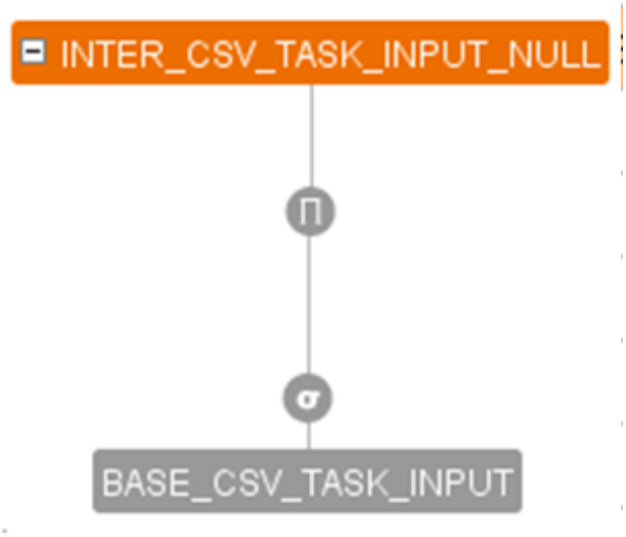


Figura D.21: Treeview inter_csv_task_input_null

Los campos de la nueva tabla son:

INTER_CSV_TASK_INPUT_NULL	
NIF	text
CUENTA	text
DIREC	text
PLAZA	text
TELEF	text
NOMBRE	text
APEL1	text
APEL2	text
TIPO	text
FILENAME	text
PROVINCIA	text

Figura D.22: Campos de la tabla inter_csv_task_input_null

D.0.0.12. Inter_csv_task_input_province

La vista derivada inter_csv_task_input_province, es una unión de las vistas inter_csv_distinct_provinces (51) e inter_csv_task_input_null (53).

Figura D.23: Treeview inter_csv_task_input_provinces

Los datos de la nueva vista son:

INTER_CSV_TASK_INPUT_PROVINCES	
NIF	text
CUENTA	text
DIREC	text
PLAZA	text
TELEF	text
NOMBRE	text
APEL1	text
APEL2	text
TIPO	text
FILENAME	text
PROVINCIA	text
PARAM	text

Figura D.24: Campos de la vista inter_csv_task_input_provinces

Bibliografía

[1] Web 2.0. http://www.microsoft.com/business/smb/es-es/internet/web_2.mspx

[2] Web 2.0. http://www.slideshare.net/lu154f/la-web-20-4109930

[3] Información Web en la actualidad. http://internet.suite101.net/article.cfm/la-obtencion-de-la-informacion-en-la-actualidad

[4] Redes Sociales. http://www.whatissocialnetworking.com/

[5] Redes Sociales. http://www.slideshare.net/igroflo/redes-sociales-2066545

[6] Redes Sociales. http://www.slideshare.net/Kuka2010/el-uso-de-redes-sociales

[7] Problemática en redes sociales. http://www.legaltoday.com/opinion/articulos-de-opinion/los-problemas-derivados-de-las-redes-sociales

[8] Problemática en redes sociales. http://www.x-novo.com/los-problemas-derivados-de-las-redes-sociales-2/

[9] Facebook. www.facebook.com

[10] Sonico. www.sonico.com

[11] Twitter. www.twiter.com

[12] Linkedin. www.linkedin.com

[13] Definition of Blogging. http://www.enterpriseblogs.info/definition-blogging

[14] ¿Qué es un foro?. http://www.creatuforo.com/que_es_un_foro.html

[15] Denodo Technologies. http://www.denodo.com

[16] Denodo Platform 4.6. http://help.denodo.com/platform-docs/4.6/

[17] Mashup Developer Community: http://www.jackbe.com/enterprise-mashup/

[18] http://www.webmashup.com: Mashups and Web 2.0 API

[19] Programmableweb: http://www.programmableweb.com/

[20] Ejemplo Mashup. http://es.wikipedia.org/wiki/Mashup_aplicaci %C3 %B3n_web_h %C3 %ADbrida)

[21] http://rss.softwaregarden.com/aboutrss.html

[22] Apache Wicket: http://wicket.apache.org/

[23] Java: http://java.sun.com/

[24] Java AWT Api: http://java.sun.com/j2se/1.4.2/docs/api/java/awt/package-summary.html

[25] Java Swing Api: http://java.sun.com/j2se/1.4.2/docs/api/javax/swing/package-summary.html

[26] Eclipse: http://www.eclipse.org/

[27] JSP. http://java.sun.com/products/jsp/

[28] Apache Struct. http://struts.apache.org/

[29] JSF. http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html

[30] Apache Tapestry: http://tapestry.apache.org/

[31] Echo: http://echo.nextapp.com/site/

[32] Plain Old Java Object: http://java.sun.com/developer/Books/javaprogramming/pojos/

[33] JDO: http://java.sun.com/jdo/

[34] Hibernate. http://www.hibernate.org/

[35] Manual Hibernate. http://www.javahispano.org/contenidos/archivo/77/Manual-Hibernate.pdf

[36] Spring Source. http://www.springsource.org/

[37] Evaluación de Frameworks. The Spring Framework - Reference Documentation. http://www.javatx.cn/spring/reference/html_single/

[38] http://bdigital.eafit.edu.co/bdigital/PROYECTO/P005.12CDP127/MarcoTeorico.pdf

[39] C Programming. http://www.cprogramming.com/

[40] JDBC. http://java.sun.com/javase/technologies/database/

[41] SQL. http://www.sql.org/

[42] DAO Data Acces Object. http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html

[43] Oracle. http://www.oracle.com/index.html

[44] DB2. http://www-01.ibm.com/software/data/db2/

[45] MYSQL. http://www.mysql.com/

[46] JavaScript Object Notation. http://www.json.org/

[47] Microsoft .NET. http://www.microsoft.com/net/

[48] Apache Version 2.0. http://httpd.apache.org/docs/2.0/

[49] Enterprise JavaBean. http://java.sun.com/products/ejb/

[50] Java Naming and Directory Interface (JNDI). http://java.sun.com/products/jndi/

[51] Ibatis. http://ibatis.apache.org/

[52] BeanFactory. http://static.springsource.org/spring/docs/2.0.x/api/org/springframework/beans/factory/BeanFactory.html

[53] Java Servlets. http://java.sun.com/products/servlet/

[54] AOP Alliance. http://aopalliance.sourceforge.net/

[55] DENODO VIRTUAL DATAPORT 4.6 ADMINISTRATOR GUIDE. http://help.denodo.com/platform-docs/4.6/DenodoVirtualDataPort.AdministratorGuide.pdf

[56] DENODO VIRTUAL DATAPORT 4.6 DEVELOPER GUIDE. http://help.denodo.com/platform-docs/4.6/DenodoVirtualDataPort.DeveloperGuide.pdf

[57] DENODO ITPILOT 4.6 USER MANUAL. http://help.denodo.com/platform-docs/4.6/DenodoITPilot.UserGuide.pdf

[58] DENODO ARACNE 4.6 ADMINISTRATOR GUIDE. http://help.denodo.com/platform-docs/4.6/DenodoAracne.AdministratorGuide.pdf

[59] DENODO SCHEDULER 4.6 ADMINISTRATOR GUIDE. http://help.denodo.com/platform-docs/4.6/DenodoScheduler.AdministratorGuide.pdf

[60] Google Mini. http://www.google.com/enterprise/mini/

[61] Apache Derby. http://db.apache.org/derby/

[62] MySQL Open Source Database. http://www.mysql.com/

[63] Oracle Database. Oracle Corporation. http://www.oracle.com/database/index.html

[64] Microsoft Word. http://office.microsoft.com

[65] Adobe Portable Document Format. http://www.adobe.com/products/acrobat/adobepdf.html

[66] DENODO ITPILOT 4.6 DEVELOPER GUIDE. http://help.denodo.com/platform-docs/4.6/DenodoITPilot.Developer.pdf

[67] DENODO ITPILOT 4.6 NSEQL MANUAL. http://help.denodo.com/platform-docs/4.6/DenodoITPilot.NSEQLManual.pdf

[68] DENODO ITPILOT 4.6 DEXTL MANUAL. http://help.denodo.com/platform-docs/4.6/DenodoITPilot.DEXTLManual.pdf

[69] DENODO VIRTUAL DATAPORT 4.6 ADVANCED VQL GUIDE. http://help.denodo.com/platform-docs/4.6/DenodoVirtualDataPort.AdvancedVQLGuide.pdf

[70] Microsoft Exchange Server. http://www.microsoft.com/exchange/

[71] Salesforce.com. On-demand Customer Relationship Management. http://www.salesforce.com/

[72] JavaScript. http://www.javascript.com/

[73] The Official Captcha Site. http://www.captcha.net.

[74] Bing. http://www.bing.com/

[75] 123People.es. http://www.123people.es/

[76] Pipl. http://pipl.com/

[77] ZabaSearch. http://www.zabasearch.com/

[78] Wink. http://wink.com/

[79] Who Is This Person? https://addons.mozilla.org/es-ES/firefox/addon/1912/?id1912&application=firefox%22%20%5Ct%20%22_blank

[80] The Forrester Wave: Information-As-A-Service, Q1 2010. Noel Yuhanna and Mike Gilpin for Application Development & Program Management Professionals

[81] Forrester Research. http://www.forrester.com/rb/research

[82] Informatica The Data Integration Company. http://www.informatica.com/Pages/index.aspx

[83] Informatica PowerCenter. http://www.informatica.com/products_services/powercenter/Pages/index.aspx

[84] Informatica B2B Data Transformation. http://www.informatica.com/products_services/b2b_data_transformation/Pages/index.aspx

[85] Informatica PowerExchange. http://www.informatica.com/products_services/powerexchange/Pages/index.aspx

[86] Informatica Data Archive. http://www.informatica.com/products_services/data_archive/Pages/index.aspx

[87] Informatica Data Explorer. http://www.informatica.com/products_services/data_explorer/Pages/index.aspx

[88] Informatica Data Subset. http://www.informatica.com/products_services/data_subset/Pages/index.aspx

[89] Biblioteca del Software AddressDoctor. http://www.informatica.com/products_services/address_validation/Pages/index.aspx

[90] Informatica Data Privacity. http://www.informatica.com/products_services/data_privacy/Pages/index.aspx

[91] Informatica Data Quality. http://www.informatica.com/products_services/data_quality/Pages/index.aspx

[92] Informatica Cloud. http://www.informaticacloud.com/

[93] Informatica Identity Resolution. http://www.informatica.com/products_services/identity_resolution/ Pages/index.aspx

[94] Informatica Data Service. http://www.informatica.com/products_services/data_services/Pages/index.aspx

[95] Gestión de Datos Maestros. http://www.informatica.com/products_services/mdm/Pages/index.aspx

[96] Informatica B2B Data Exchange. http://www.informatica.com/products_services/b2b_data_exchange/Pages/index.aspx

[97] Informatica RulePoint. http://www.informatica.com/products_services/rulepoint_operational_intelligence/Pages/index.aspx

[98] Composite. http://www.compositesw.com/

[99] Composite Discovery. http://www.compositesw.com/index.php/products/composite-discovery/

[100] Composite Information Server. http://www.compositesw.com/index.php/products/composite-information-server/

[101] Composite Applications Data Services. http://www.compositesw.com/index.php/products/composite-application-data-services/

[102] Composite Monitor. http://www.compositesw.com/index.php/products/composite-monitor/

[103] Composite Active Cluster. http://www.compositesw.com/index.php/products/composite-active-cluster/

[104] IBM. http://www.ibm.com/us/en/

[105] IBM InfoSphere DataStage. http://www-01.ibm.com/software/data/infosphere/datastage/

[106] DataStage Designer. http://www-01.ibm.com/support/docview.wss?uid=swg21381848

[107] DB2 Connect. http://www-01.ibm.com/software/data/db2/db2connect/

[108] DataStage Director. http://www-01.ibm.com/support/docview.wss?uid=swg21372885

[109] Microsoft. http://www.microsoft.com/en/us/default.aspx

[110] Microsoft SQL Server 2008. http://www.microsoft.com/sqlserver/2008/en/us/

[111] Microsoft Data Engine. http://msdn.microsoft.com/en-us/library/ms811092.aspx

[112] Microsoft BizTalk Server. http://www.microsoft.com/biztalk/en/us/default.aspx

[113] Red Hat. http://www.es.redhat.com/

[114] MetaMatrix Enterprise Data Services Platform. https://www.jbossgroup-europe.ch/ products/platforms/dataservices/

[115] JBoss Enterprise SOA Platform. http://www.jboss.com/products/platforms/soa/

[116] Fedora Project. http://fedoraproject.org/

[117] Jboss Enterprise. http://www.jboss.com/products/