



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

## FACULTAD DE INFORMÁTICA

# TESINA DE LICENCIATURA

**TÍTULO:** Extendiendo el browser web con soporte para búsquedas peer-to-peer

**AUTORES:** Fermín Baudino Zoya

**DIRECTOR:** Sergio Firmenich, Alejandro Fernandez

**CODIRECTOR:**

**ASESOR PROFESIONAL:**

**CARRERA:** Licenciatura en Informática

### Resumen

Los navegadores web son aplicaciones de software que permiten, entre otras cosas, interactuar con páginas web y con la información que estas contienen. Los navegadores web no siempre cumplen con todas las necesidades que tienen los usuarios ya que estas cambian con el tiempo. Es así que llegan las Extensiones Web, que son pequeñas aplicaciones de software para personalizar un navegador web. Lo que se propone en este trabajo es agregar a la extensión web desarrollada en ANDES la posibilidad de integrarla a una red peer-to-peer, y de esta manera lograr que los usuarios puedan comunicarse entre ellos aprovechando una arquitectura peer-to-peer. Para lograr este objetivo se usarán tecnologías existentes creadas en otros trabajos de investigación, contribuyendo a estas de tal manera que puedan ser integradas en una y brindar un mayor potencial a las mismas.

### Palabras Clave

WebExtensions, Peer-to-peer, Burbuja de Filtrado, Peerify, Web Semántica.

### Conclusiones

Se logró desarrollar una potente herramienta de definición de APIs de búsqueda (search APIs) basadas en web scraping que permita brindar servicios de datos basándose en una arquitectura peer-to-peer, además, sumado a esta herramienta se logró plantear casos o escenarios de uso como pruebas de concepto para evidenciar el potencial de la misma y algunos posibles escenarios donde podría ser de utilidad.

### Trabajos Realizados

Se analizó el middleware y el framework, el primero proporcionó la necesaria capa de comunicación peer-to-peer, mientras que el segundo facilitó la conexión entre las extensiones web, dotándolas de funcionalidad P2P. Se llevó a cabo un análisis y estudio de ANDES. Se desarrolló la herramienta "Peerify". Se desarrollaron pruebas de concepto con casos de uso en los que podría ser de utilidad la herramienta "Peerify".

### Trabajos Futuros

Agregar a "Peerify" la posibilidad de guardar un registro o historial donde se almacenen las búsquedas donde participó cada peer y los servicios de búsqueda involucrados, entre otras cosas. Agregar una capa de seguridad a la interacción con la red peer-to-peer. Realizar una evaluación de "Peerify" con personas en un entorno preparado para distintos tipos de prueba.

Extendiendo el browser web con soporte para  
búsquedas peer-to-peer.

Fermín Baudino Zoya

Universidad Nacional de La Plata  
Facultad de Informática

Directores:  
Sergio Firmenich  
Alejandro Fernandez

23 de noviembre de 2023

## Resumen

Los navegadores web son aplicaciones de software que permiten, entre otras cosas, interactuar con páginas web y con la información que estas contienen. Como es usual con las aplicaciones de software, los navegadores web no siempre cumplen con todas las necesidades que tienen los usuarios ya que estas cambian con el tiempo. Es frecuente que un usuario quiera realizar algún tipo de modificación ya sea en el diseño o estilo del navegador, o agregar alguna funcionalidad o herramienta a este, o complementar y mejorar algún sitio web. Es así que llegan las Extensiones Web (Web Extensions [3]), que son pequeñas aplicaciones de software para personalizar un navegador web. Permiten extender y modificar las capacidades de un navegador web agregando características y funciones al mismo. En ANDES [1] se propone un enfoque basado en extensiones de navegador para dar a los usuarios la posibilidad de integrar nuevos servicios de búsqueda en los navegadores. Estos servicios se crean personalizando los servicios de búsqueda existentes en la World Wide Web (Web). La herramienta propuesta en ANDES se basa en dos conceptos: Web Augmentation (WA) y End User Development (EUD). El primer concepto se refiere a tecnologías que modifican sitios web para incluir características requeridas por los usuarios pero que no son provistas por el sitio. Por otro lado, el segundo concepto hace referencia a tecnologías que permiten a los usuarios crear sus propias herramientas para satisfacer sus requisitos.

Lo que se propone en este trabajo es agregar a la extensión web desarrollada en [1] la posibilidad de integrarla a una red peer-to-peer, y de esta manera lograr que los usuarios de ANDES puedan comunicarse entre ellos aprovechando una arquitectura peer-to-peer. Para lograr este objetivo se usarán tecnologías existentes creadas en otros trabajos de investigación, contribuyendo a estas de tal manera que puedan ser integradas en una y brindar un mayor potencial a las mismas. Estas tecnologías existentes fueron desarrolladas en los trabajos [1] y [2].

# Índice general

<b>1. Introducción</b>	<b>2</b>
1.1. Motivación . . . . .	4
1.2. Estructura y organización de la tesis . . . . .	6
<b>2. Background</b>	<b>7</b>
2.1. Web Scraping . . . . .	7
2.2. Web Semántica . . . . .	9
2.3. WOA y Andes . . . . .	10
2.4. Volunteer Computing . . . . .	11
2.5. P2P y sus topologías . . . . .	12
<b>3. Trabajo Relacionado</b>	<b>14</b>
3.1. Composición de API de ANDES . . . . .	15
<b>4. Enfoque y Arquitectura</b>	<b>19</b>
4.1. Comunicación e interacción p2p . . . . .	21
4.2. Middleware P2P . . . . .	22
<b>5. Herramientas</b>	<b>24</b>
5.1. “A data service layer for web browser extension	24
5.2. Middleware P2P . . . . .	31
5.3. Configuración inicial . . . . .	32
<b>6. Casos de uso</b>	<b>36</b>
6.1. Integración de búsquedas P2P . . . . .	38
6.2. Burbuja de filtrado . . . . .	43
<b>7. Conclusiones y trabajo futuro</b>	<b>49</b>

# Capítulo 1

## Introducción

La evolución de la web y de los navegadores ha sido un proceso constante y dinámico, marcado por la incorporación de diversas tecnologías que han transformado la forma en que los usuarios interactúan con los contenidos en línea. En sus inicios cuando Tim Berners-Lee creó la World Wide Web (WWW), también llamada Web 1.0, los usuarios no podían interactuar con el contenido de esta, sino que era una web unidireccional de solo lectura. Luego con la llegada de la Web 2.0 los usuarios ya no podían únicamente leer el contenido de la web sino que también podían crear nuevo contenido, modificarlo, y actualizarlo. [3] Relacionado a la web, el crecimiento y evolución también lo tuvieron los navegadores web. Desde sus inicios en 1991 con la creación del primer navegador web hasta la actualidad, se han creado decenas de navegadores y a su vez se fueron incorporando nuevas tecnologías y herramientas a estos para enriquecerlos y brindar un mayor potencial a los mismos. Un claro ejemplo de esto son las extensiones para los navegadores, herramientas que agregan características y funciones a un navegador. Con estas un usuario podría agregar características o funciones como por ejemplo mejorar o complementar una página web, modificar el aspecto del navegador, agregar o quitar contenido de las páginas web, entre otras cosas. Como referencia, hoy en día existen más de 120.000 extensiones de navegador<sup>1</sup> publicadas en internet para poder ser descargadas y utilizadas por usuarios en el uso cotidiano del navegador. Y se sabe que cientos de estas extensiones superan el millón de usuarios activos, y otras miles de las extensiones existentes superan los miles de usuarios activos.

La tecnología dentro de los navegadores web ha avanzado significativamente en los últimos años, incorporando una variedad de protocolos para

---

<sup>1</sup><https://chrome-stats.com/insights>

permitir comunicaciones seguras y auténticas entre navegadores sin depender de una infraestructura central. [4] Estos protocolos permiten la comunicación de navegador a navegador sin sacrificar la autenticidad, la confidencialidad o la integridad de la comunicación .

Uno de estos protocolos se conoce como Comunicación Web en Tiempo Real (WebRTC <sup>2</sup>, por sus siglas en inglés), es una tecnología que permite la comunicación directa entre dos navegadores web sin la necesidad de un servidor intermediario. Esta tecnología emplea una arquitectura peer-to-peer. En un modelo tradicional de cliente-servidor toda la comunicación pasa por un servidor central, lo cual puede llevar a mayor latencia en la comunicación y posibles cuellos de botella. Por otro lado, con WebRTC y gracias al uso de la arquitectura peer-to-peer los datos pueden intercambiarse directamente entre los navegadores, reduciendo la latencia en las comunicaciones y mejorando el rendimiento. De este modo, una extensión web podría conectarse a una red peer-to-peer mediante el uso de protocolos como WebRTC para establecer conexiones y comunicarse con otros navegadores.

WebRTC hace uso de un proceso llamado “señalización” para establecer una conexión entre dos browsers. La señalización implica el intercambio de ciertos datos sobre los browsers, como por ejemplo la IP, para establecer la ruta más eficiente para la comunicación. Si bien este proceso de señalización es facilitado por un servidor de señalización, una vez que se establece la conexión la transferencia de datos ocurre directamente entre los navegadores.

En resumen, WebRTC aprovecha la arquitectura peer-to-peer para permitir una comunicación directa, eficiente y segura entre dos navegadores web, sin la necesidad de un intermediario.

Además del crecimiento y evolución de la web y los navegadores web, los buscadores dentro de la web también fueron evolucionando y con ellos nuestra manera de buscar información en la web. Al término “Information Retrieval” (IR) se lo define en *Introduction to Information Retrieval (p.1)* como “encontrar material de naturaleza no estructurada que satisfaga una necesidad de información dentro de grandes colecciones”. En la actualidad todos nos vemos involucrados en el proceso de IR al utilizar la web. [5]

Cuando se desea buscar información sobre algún tópico lo primero que solemos pensar es en la web (World Wide Web), pocas personas hoy en día deben tener como primera idea buscar información en, por ejemplo, un libro. Además de ser en la gran mayoría de los casos mas veloz, estamos ya muy acostumbrados a que por cualquier inquietud que tengamos y necesitemos buscar información sobre algo, al primer lugar que acudimos es a la web.

---

<sup>2</sup>[https://developer.mozilla.org/en-US/docs/Web/API/WebRTC\\_API](https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API)

Por esto es que es tan importante el proceso de “recuperación de la información” o búsqueda de información en la web. En [1] se describen dos tipos de búsquedas: las “búsquedas primarias” (primary searches) y las “búsquedas secundarias” (ancillary searches). Define a las búsquedas primarias como tareas de búsqueda que se realizan iterativamente y múltiples veces hasta encontrar un resultado esperado. Por otro lado, define a las búsquedas secundarias como tareas que surgen a partir de búsquedas primarias con el fin de obtener información necesaria para completar la búsqueda primaria

Con respecto a las búsquedas en la web, existe un término nombrado “burbuja de filtrado” (ó “filter bubble”) que fue acuñado por Eli Pariser en su libro “The Filter Bubble: What The Internet Is Hiding From You”, en él intenta capturar su preocupación de que el aumento del uso de la personalización en la clasificación de los resultados de los motores de búsqueda y las redes sociales crearían “un universo único de información para cada uno de nosotros” (2011, p. 10), erosionando la posibilidad de un terreno común relativamente compartido. Esto se debe a que podríamos ver más y más cosas que nos gustan, mientras que las cosas que no somos propensos a que nos gusten se nos ocultan. En otras palabras, se puede imaginar a las burbujas de filtrado como “barrios informáticos” donde se entregan mensajes que se ajustan a nuestras preferencias, lo que puede resultar en una visión del mundo cada vez más personalizada y menos objetiva. [6]

## 1.1. Motivación

Dado este contexto de la web y los navegadores web, la motivación que lleva a realizar este trabajo es la de complementar, con una arquitectura peer-to-peer y con herramientas como la desarrollada en [1], una red de extensiones web (muchas extensiones web conectadas entre sí) que en conjunto hacen una extracción de información (scraping) usando servicios de búsqueda previamente creados por los usuarios, con las cuales se obtienen datos. Complementar la extensión con la arquitectura peer-to-peer es lo que da la posibilidad de crear esta red de extensiones, pudiendo de esta manera realizar las búsquedas para la extracción de información de manera distribuida y que los resultados de ellas puedan ser distribuidos utilizando mecanismos P2P.

El objetivo que se quiere lograr es el de utilizar estas tecnologías mencionadas para construir una extensión web y que con esta cualquier usuario tenga la posibilidad de definir servicios de búsqueda para los sitios web que sean concernientes a sus diferentes intereses, y que luego, mediante el uso

de la arquitectura peer-to-peer, pueda conectarse a otros usuarios y permitir realizar una búsqueda para ellos. Esta búsqueda se dispararía a todos los pares conectados, inclusive él mismo, y de esta manera cada usuario estaría extrayendo información de sus servicios de búsqueda definidos para finalmente reunir toda esa información en un solo lugar, todo esto sin la necesidad de escribir a mano la búsqueda en cada sitio web, recolectar los datos arrojados por los mismos, y enviarlos al usuario que lo necesitaba. Por lo tanto, a modo de resumen se tendría una extensión web con la cuál se pueda recolectar datos sobre algún tópico en específico utilizando varias computadoras a la vez, para después unir toda esa información recolectada y entregarla de una manera consistente al usuario que inició la búsqueda. Sumado a esto está el hecho de que se podría evitar la burbuja de filtrado gracias a la unión de los resultados de cada peer ya que estos resultados no estarían “cerrados” a los intereses de uno mismo sino que estarían vinculados a cada “universo único de información” de cada peer y esto lograría un mayor rango de resultados diferentes, librándolo a uno de su “burbuja”.

Dicho esto, se propone utilizar las extensiones web de los navegadores web integrándolas a una red peer-to-peer para que se puedan comunicar aprovechando la arquitectura P2P. Se usarán tecnologías existentes creadas en otras investigaciones contribuyendo a éstas de tal manera que puedan ser integradas en una y brindar un mayor potencial a las mismas. Entre los desarrollos propuestos se definen:

- Analizar y estudiar el Middleware que provee una capa de comunicación P2P.
- Analizar y estudiar el Framework que se utiliza en la construcción de las extensiones web para hacer uso del Middleware y convertir las extensiones en P2P.
- Analizar y estudiar la herramienta desarrollada en [1].
- Desarrollar una extensión web, que llamaremos “Peerify”, que mediante el uso de una arquitectura P2P provea la posibilidad de definir servicios de búsqueda para cualquier sitio web y realizar, con la ayuda de estos servicios, una búsqueda sobre un tópico específico utilizando los peers conectados a la red P2P.

A modo de conclusión de la propuesta, brindar una extensión como la propuesta en esta tesis puede llegar a ser de mucha utilidad para una gran cantidad de usuarios, teniendo en cuenta el total de usuarios activos que llegan a tener las extensiones existentes anteriormente mencionadas. Además,



estas extensiones no solo son usadas por usuarios para un uso personal, sino que también muchas de estas extensiones son usadas por instituciones académicas para trabajos de investigación a las que les serviría el tener esta capa de datos peer-to-peer.

## 1.2. Estructura y organización de la tesis

Esta sección tiene la finalidad de explicar como esta organizado el documento; como se lee, y que es lo que se va a encontrar en cada capítulo.

El presenta trabajo se distribuye en 7 capítulos, a continuación se describen de forma breve los mismos:

- Capítulo 2: introduce al background de este trabajo, los conceptos que se relacionan con el mismo.
- Capítulo 3: trata trabajos relacionados a la herramienta desarrollada en este trabajo.
- Capítulo 4: es sobre una visión general del enfoque y como se estructura la herramienta desarrollada.
- Capítulo 5: explica las herramientas utilizadas en el desarrollo del trabajo.
- Capítulo 6: se plantean los casos de uso o escenarios a los cuales se podría aplicar la herramienta desarrollada.
- Capítulo 7: conclusiones y posibles adiciones en futuros trabajos.

## Capítulo 2

# Background

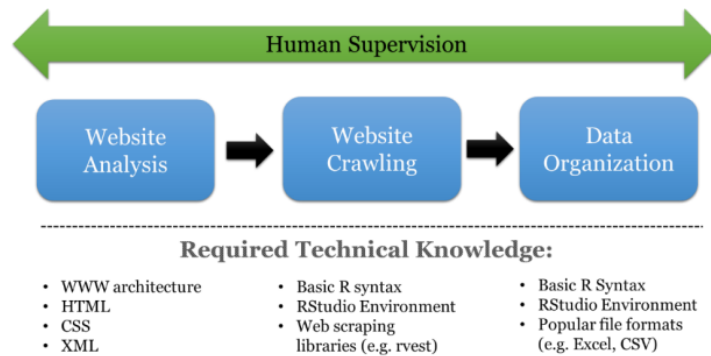
### 2.1. Web Scraping

El término “Web Scraping” (también conocido como “web data extraction”, “web data scraping”, “web harvesting”) refiere al proceso de extracción de información de un sitio web de manera automática. [7] Este proceso se lleva a cabo mediante el uso de algún tipo de software como por ejemplo, herramientas especialmente diseñadas que simulan la “navegación” humana por una página web en la que analizan su contenido, lo extraen, y lo almacenan en un formato especial (puede ser JSON, CSV, HTML, etc) para luego ser procesado y utilizado. [8]

En el vasto océano de datos que hay en la Web hoy en día esta se ha convertido en un enorme depósito de datos que se expande constantemente en el que tanto personas como sistemas informáticos están continuamente absorbiendo y generando nueva información. Este incesante flujo de datos hace que el Web Scraping sea realmente útil para el proceso de recolección de información.

Actualmente existen diversas técnicas de scraping [7], algunas de ellas son :

- El tradicional “copiar y pegar”
- Expresiones regulares
- Programación HTTP
- Analisis del DOM
- Software para Web Scraping



Web Scraping (Adapted from Krotov & Tennyson, 2018; Krotov & Silva, 2018; Krotov et al., 2020)

Figura 2.1: Web Scraping (Adaptado de Krotov & Tennyson, 2018; Krotov & Silva, 2018; Krotov et al., 2020)

Por último, este proceso de Web Scraping puede ser útil para distintas prácticas tales como:

- Comparación de precios en línea
- Monitoreo de cambios en un sitio
- Monitoreo del clima
- Recopilación de datos para una investigación
- “Mash up web”, una combinación de distintas paginas web
- Análisis de mercados

[7] sin embargo, estos son solo algunos ejemplos de los posibles usos que se le puede dar al Web Scraping ya que las posibilidades de uso que tiene el Web Scraping son aún mucho más amplias.

En [8] se describe al proceso de Web Scraping en 3 etapas:

1. Etapa de búsqueda: en esta etapa se accede a la página (mediante el uso del protocolo HTTP) de la que se desea extraer la información.
2. Etapa de extracción: esta etapa refiere al momento de “seleccionar” la información deseada luego de haber accedido a la página web. En esta etapa se puede utilizar técnicas como expresiones regulares, consultas de XPATH, librerías para analizar contenido HTML, entre otras cosas.

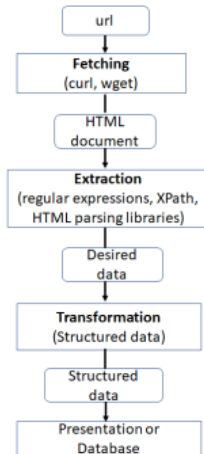


Figura 2.2: Proceso de Web Scraping (Persson, 2019).

3. Etapa de transformación: En esta última etapa ya se posee la información deseada por lo que será transformada a un formato estructurado para ser visualizada o almacenada.

en la figura 2.2 se puede observar este proceso en un diagrama.

## 2.2. Web Semántica

Iniciaremos esta sección dando una definición de lo que se conoce como “agente”, un agente es una pieza de software que se ejecuta sin una interacción directa por parte de un humano o una supervisión constante para lograr un fin provisto por un usuario. Los agentes recolectan, filtran, y procesan información que se encuentra en la web. [9] Una gran cantidad de información en la web hoy en día está diseñada para que la lean los humanos, no para ser manipulada por programas informáticos. Las computadoras no tienen una manera confiable de entender el significado de la información en la web. Según Tim Berners-Lee en [9] la web semántica es una extensión de la web que conocemos en la que se le da a la información un significado claro para que las computadoras y las personas puedan trabajar cooperativamente. “La web semántica permitirá a las máquinas comprender documentos y datos semánticos”.

Una alternativa de definición se encuentra en el trabajo de referencia

[10], que conceptualiza la web semántica como “la web de los significados”. Esta perspectiva se fundamenta en los conceptos de sintaxis y semántica. La sintaxis se refiere a la forma en que se expresa un concepto, mientras que la semántica se enfoca en el significado de dicho concepto. De esta manera, se plantea la posibilidad de concebir la web semántica desde esta óptica.

### 2.3. WOA y Andes

En esta sección se hablará sobre WOA [11] y ANDES [1], dos herramientas desarrolladas en trabajos de investigación que permiten abstraer contenidos en la web y materializarlos como objetos, y tener una forma de buscar estos objetos en diferentes fuentes de información, respectivamente. Con WOA se busca lograr una experiencia web personal donde se pueda materializar información de la web en objetos dentro del browser, permitiendo interactuar con estos de diversos modos. ANDES lleva esta idea a un punto mas lejano, presenta una manera de poder buscar en la web a través de la creación de modelos de servicios de búsqueda sobre los cuales se pueden realizar búsquedas de los objetos de información previamente creados. La creación de los modelos y la búsquedas combinan conceptos de End User Development y Web Scraping, logrando una herramienta que no necesita de conocimientos técnicos de programación y una extracción de la información de manera automática.

En [11] se presenta Web Objects Ambient (WOA), una plataforma que recopila contenido web y lo materializa en objetos de información concretos que luego residen en el navegador web en un entorno de objetos comunes para ser utilizados y mejorarlos para que los usuarios puedan interactuar con ellos.

En WOA el concepto de materialización se define en 3 pasos: “identificar los objetos de información de interés para el usuario en el contexto de una página web”, “estructurar la abstracción”, y “extraer las instancias disponibles de los objetos”.

En resumen, WOA es una plataforma que recopila información de la web y la convierte en objetos concretos que pueden ser utilizados por los usuarios en un entorno común.

Conociendo que es WOA y su funcionamiento podemos encontrar un problema en esta herramienta. Éste es que no existe un modo de poder buscar utilizando estas materializaciones de objetos en fuentes existentes. Es decir, poder buscar esos objetos en diferentes fuentes en la web. Teniendo esto en

cuenta, existe la necesidad de crear una herramienta para poder materializar datos de la web en objetos y utilizarlos como modelos de búsquedas en servicios de internet.

En [1] se propone una herramienta para mejorar las tareas de búsquedas de los usuarios de la web mediante la personalización de los servicios de búsqueda para enriquecer la experiencia de búsqueda de los usuarios e intentar hacerlas más eficientes y efectivas. Con esta herramienta propuesta, llamada ANDES, se propone un enfoque basado en extensiones de navegador para dar a los usuarios la posibilidad de integrar nuevos servicios de búsqueda en los navegadores. Estos servicios se crean personalizando los servicios de búsqueda existentes en la Web.

Esta herramienta se basa en dos conceptos: Web Augmentation (WA) y End User Development (EUD). El primer concepto se refiere a tecnologías que modifican sitios web para incluir características requeridas por los usuarios pero que no son provistas por el sitio. Por otro lado, el segundo concepto hace referencia a tecnologías que permiten a los usuarios crear sus propias herramientas para satisfacer sus requisitos.

ANDES permite principalmente dos cosas: 1) Crear servicios de búsqueda (mencionados como “search APIs”) para la búsqueda de objetos de información en fuentes específicas; 2) Poder utilizar los servicios de búsqueda definidos con la herramienta para cumplir con el objetivo de satisfacer “ancillary searches”. La creación de las “search APIs” se lleva a cabo por los usuarios sin la necesidad de tener conocimientos técnicos de programación gracias al EUD (End User Development), por otro lado, la utilización de estas “search APIs” se logra simplemente seleccionando un texto dentro del navegador, luego abriendo el menú de contexto del navegador con el click derecho, y por último haciendo click izquierdo en una de las opciones de los servicios de búsqueda disponibles mostrados en este menú.

Esta tesina extenderá la herramienta presentada en [1] para brindar a las extensiones web una capa de servicio de datos en la que se tengan servicios de búsqueda, generados manualmente por cada usuario sin la necesidad de tener conocimiento de programación. Se tomará de ANDES la creación de las “search APIs” pero se propondrá otro uso sobre estas, no irán enfocadas al propósito de satisfacer las “ancillary searches”.

## 2.4. Volunteer Computing

Volunteer computing o cómputo voluntario es un tipo de “metacomputing” que se enfoca en facilitar la manera en la que las personas puedan usar

sus dispositivos electrónicos (como computadoras o teléfonos inteligentes) para ser parte de una “metacomputadora”. [12]

Es una forma de cómputo distribuido basado en red, que permite a los participantes compartir sus recursos informáticos inactivos y ayuda a ejecutar proyectos computacionalmente costosos.

Volunteer Computing se puede clasificar en diferentes criterios, estos son:

1. Autonomía
2. Anonimidad
3. Altruismo

Autonomía refiere a si los voluntarios que prestan su poder de cómputo pueden unirse y salir libremente en cualquier momento que lo deseen o si son forzados a ser voluntarios. El segundo criterio, anonimidad, está relacionado al conocimiento que se tenga sobre los voluntarios. Es decir, si son conocidos y confiables o si son desconocidos, o no confiables. Por último, el tercer criterio trata sobre si los voluntarios esperan algo a cambio de prestar su poder de cómputo ya sea una retribución monetaria u otro tipo de compensación.

En el trabajo propuesto en esta tesina se adaptará este concepto permitiendo a navegadores conectados a una red peer-to-peer “prestar” sus servicios para realizar búsquedas para otros usuarios, además, en el sentido altruista los usuarios tendrán como retribución el servicio que ellos prestan a través de los demás usuarios.

## 2.5. P2P y sus topologías

Las redes Peer-to-Peer (P2P) han evolucionado significativamente a lo largo de los años, adaptándose a un número creciente de participantes y a las cambiantes demandas de los usuarios. Una característica fundamental de estas redes es su topología.

En las redes P2P, los nodos (o “peers”) se conectan entre sí formando una topología que puede ser estructurada o no estructurada. En una topología no estructurada, los peers se conectan de manera aleatoria, sin seguir un patrón específico. Un ejemplo de esto es la red Gnutella<sup>1</sup>, una de las primeras redes de intercambio de archivos P2P.

Por otro lado, en una topología estructurada la red se organiza de tal manera que cualquier nodo puede encontrar eficientemente la información

---

<sup>1</sup><https://www.gnu.org/philosophy/gnutella.es.html>

que busca, incluso si la red es muy grande. Esto se logra asignando a cada nodo y a cada dato en la red un identificador único, y luego organizando estos identificadores en una estructura de datos específica. [13]

También es importante hablar sobre los tipos de arquitecturas de redes peer-to-peer, una red peer-to-peer puede ser categorizada en alguna de las siguientes tipos de arquitecturas:

- Red peer-to-peer pura
- Red peer-to-peer híbrida
- Red peer-to-peer mediada

Se habla de una red p2p pura cuando “los participantes comparten parte de sus propios recursos de hardware (potencia de procesamiento, capacidad de almacenamiento, capacidad de enlace de red, impresoras, ...). Estos recursos compartidos son necesarios para proporcionar el Servicio y el contenido ofrecido por la red (por ejemplo, intercambio de archivos o espacios de trabajo compartidos para la colaboración): Son accesibles por otros pares directamente, sin pasar por entidades intermediarias” y cualquier participante puede ser eliminado de la red sin causar ninguna pérdida de servicio en la red. [14]

Una red p2p híbrida es una red en la que los participantes comparten parte de sus propios recursos de hardware pero existe al menos un participante central que necesita estar conectado para proveer partes del servicio ofrecido por la red. En estas redes existe lo que se conocen como “ultrapeers” [15] o “super-peers” [16] (“nodos que actúan como un servidor centralizado para un subconjunto de clientes” dentro de la red).

Por último, las redes peer-to-peer mediadas son las redes tales que utilizan un servidor para manejar las operaciones de control, por ejemplo para tener un registro de los archivos que están disponibles para consultar en la red. [15]



## Capítulo 3

# Trabajo Relacionado

La sección de “Trabajo relacionado” de esta tesina examina la investigación previa y las contribuciones académicas relevantes en el ámbito de búsquedas peer-to-peer y extensiones de navegador. En esta sección, se analizarán los trabajos anteriores que han abordado temas y problemas similares, con el objetivo de situar el presente estudio en el contexto de las investigaciones anteriores y resaltar su originalidad y aportes novedosos.

La aplicación MinervaLight busca los documentos en una red P2P formada por los ordenadores de los usuarios que tienen instalado el software. Cada ordenador tiene un índice local de los documentos que ha descargado e indexado mediante un rastreador web enfocado. Cuando un usuario introduce una consulta, MinervaLight recupera los resúmenes estadísticos del directorio, selecciona los nodos más relevantes, les envía la consulta y combina los resultados que le devuelven. [17].

En [17] las búsquedas se llevan a cabo en una fuente de información, en el repositorio local de los peers para el caso de MinervaLight. Es diferente el caso de este trabajo, las fuentes de búsqueda no se limitan a un repositorio local de cada peer conectado a la red sino que se permite a los usuarios crear “search APIs” de los sitios que ellos deseen y categorizarlos semánticamente para luego poder hacer una búsqueda global en estos.

En [18] se propone una forma de crear extensiones web que permitan a los usuarios colaborar entre sí sin depender de un servidor central, basándose en una arquitectura P2P y en el uso de WebRTC. Presenta un middleware que gestiona las comunicaciones P2P y ofrece una interfaz sencilla para enviar y recibir mensajes entre los nodos. Este middleware, diseñado e implementado en [2], fue utilizado para la realización de esta tesina.

Estos trabajos mencionados sirvieron como referencia para desarrollar el

trabajo presentado en esta tesina.

### 3.1. Composición de API de ANDES

En [1] se propone ANDES, una herramienta para mejorar las búsquedas en la web, enfocada en las búsquedas auxiliares. Esta herramienta permite a los usuarios crear servicios de búsqueda personalizados para satisfacer sus necesidades específicas. Permite a los usuarios redefinir el alcance de los servicios de búsqueda, e indicar el formato esperado de los resultados de la búsqueda. Utiliza técnicas de minería de texto para extraer información relevante de la página web actual y presentarla al usuario en una ventana superpuesta. ANDES también permite a los usuarios guardar y organizar la información extraída para su uso posterior, así como para compartir con otros usuarios los servicios de búsqueda creados. En resumen, ANDES es una herramienta útil para mejorar la eficiencia y efectividad de la búsqueda de información en la web.

En ANDES un servicio de búsqueda es la “abstracción de un motor de búsqueda”, la creación de estos servicios de búsqueda funciona de la siguiente manera. Para crear un servicio de búsqueda, el usuario primero navega a una página de resultados del motor de búsqueda del que desea crear su servicio. Una vez allí, el usuario hace click en el botón de la barra de herramientas de ANDES para iniciar. Se abre una barra lateral que ayuda al usuario a identificar los componentes del servicio de búsqueda. Se presenta una pantalla de bienvenida al usuario. Para crear un nuevo servicio, el usuario hace click en el botón “New service” y se le pide que proporcione un nombre para el nuevo servicio. El usuario debe indicar qué elemento de la página es el campo de entrada para la búsqueda. Luego hace click izquierdo en el campo de entrada deseado y este se resalta para proporcionar retroalimentación. Durante este paso, todos los elementos del formulario de búsqueda se desactivan temporalmente (por ejemplo, hacer click en el botón de búsqueda no activa la búsqueda). A continuación, el usuario debe indicar una estrategia para activar la consulta de búsqueda (por ejemplo, a través de un click en un botón o presionando la tecla “enter”). Por último, el usuario debe indicar qué elementos de la página de resultados representan los resultados. Las siguientes figuras muestran este proceso: <sup>1</sup>

---

<sup>1</sup>Las imágenes 3.1, 3.2 y 3.3 de las herramientas presentadas en esta sección fueron desarrolladas por el equipo de investigación del Lifa integrado por los directores de esta tesina

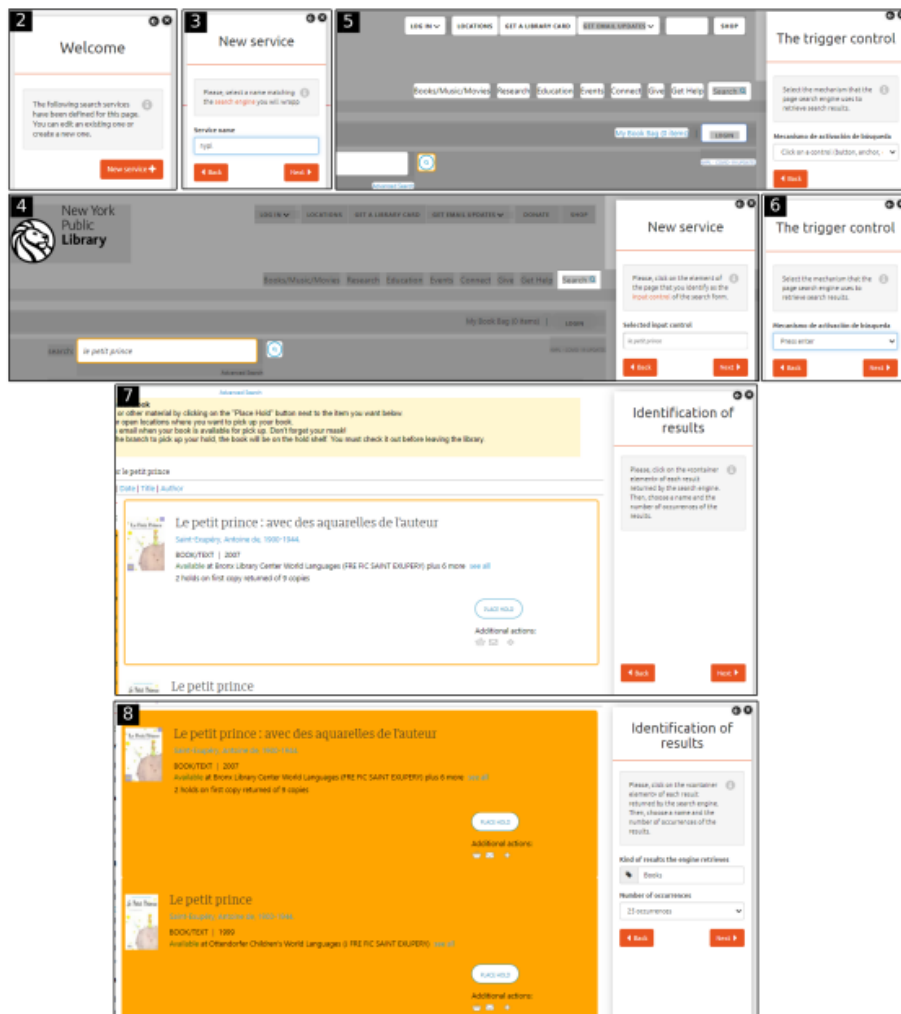


Figura 3.1: Definición de un servicio: selección del campo de entrada, del modo de invocación de la búsqueda, y la selección del contenedor de resultados[1]

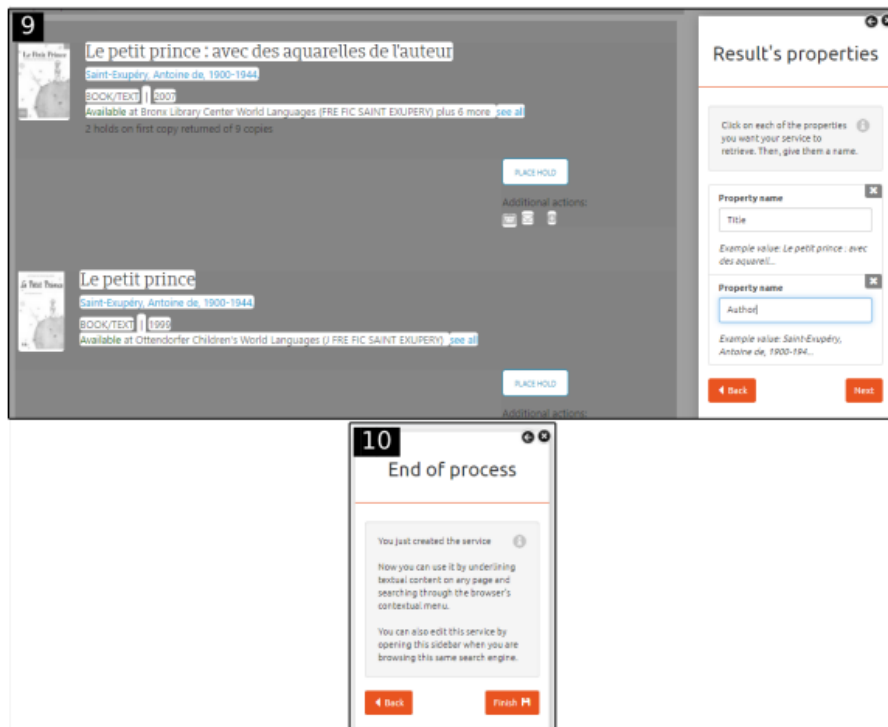


Figura 3.2: Definición de un servicio: selección de las propiedades de los resultados [1]

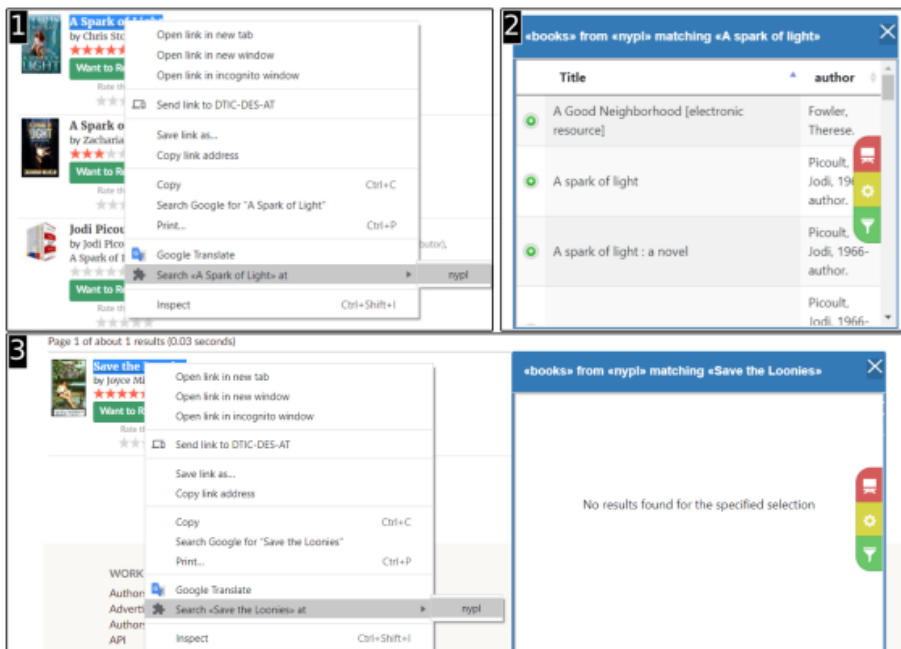


Figura 3.3: Utilizando el servicio de NYPL augmentado [1]

## Capítulo 4

# Enfoque y Arquitectura

“Los usuarios encuentran información en la web ya sea buscando o navegando.” [1] Cuando estamos haciendo una búsqueda lo hacemos con el fin de encontrar algo en particular, ya sea que deseamos resolver un problema, que tenemos una inquietud, o simplemente queremos saber como va a estar el clima, nos tomamos la tarea de hacer las búsquedas para encontrar eso que deseamos. Estas búsquedas son las tareas que mas se realizan en la web. Según “similarweb” <sup>1</sup> google.com es el sitio mas visitado, donde los usuarios pasan en promedio 00:10:38 minutos y ven en promedio 8.66 páginas antes de salir del sitio.

El proceso de búsqueda en la web sigue un conjunto de pasos claramente definidos. En primer lugar, los usuarios eligen el servicio de búsqueda de su preferencia, ya sea Google, Bing o un sitio web especializado en reseñas de libros, películas o compras en línea, entre otros. Luego, escriben su consulta para encontrar la información que buscan. Posteriormente, exploran los resultados, filtrando y seleccionando los más relevantes o interesantes. Finalmente, concluyen su búsqueda.

Este constante flujo de búsquedas ha generado la necesidad de brindar a las extensiones web una capa de servicios de datos que permita a los usuarios generar, gestionar y compartir servicios de búsqueda de manera sencilla, sin requerir conocimientos de programación. Además, estas extensiones pueden actuar como nodos en una red peer-to-peer, lo que facilita la distribución de servicios de búsqueda y colaboración entre usuarios. En las secciones siguientes, exploraremos diversos casos de uso que han surgido en este contexto, aprovechando al máximo la importancia de las búsquedas en la web.

En la siguiente figura (Figura 4.1) se muestra una idea general de como

---

<sup>1</sup><https://www.similarweb.com/top-websites/> (Último acceso Agosto 2023)

es el diseño del trabajo, cada usuario va a tener en su browser la extensión web donde se pueden administrar los servicios de búsqueda conectada a un middleware p2p que a su vez es otra extensión web. Por último, el middleware se encargará de hacer la conexión a la red y se va a encargar también de la lógica para el envío y recepción de datos entre los peers.

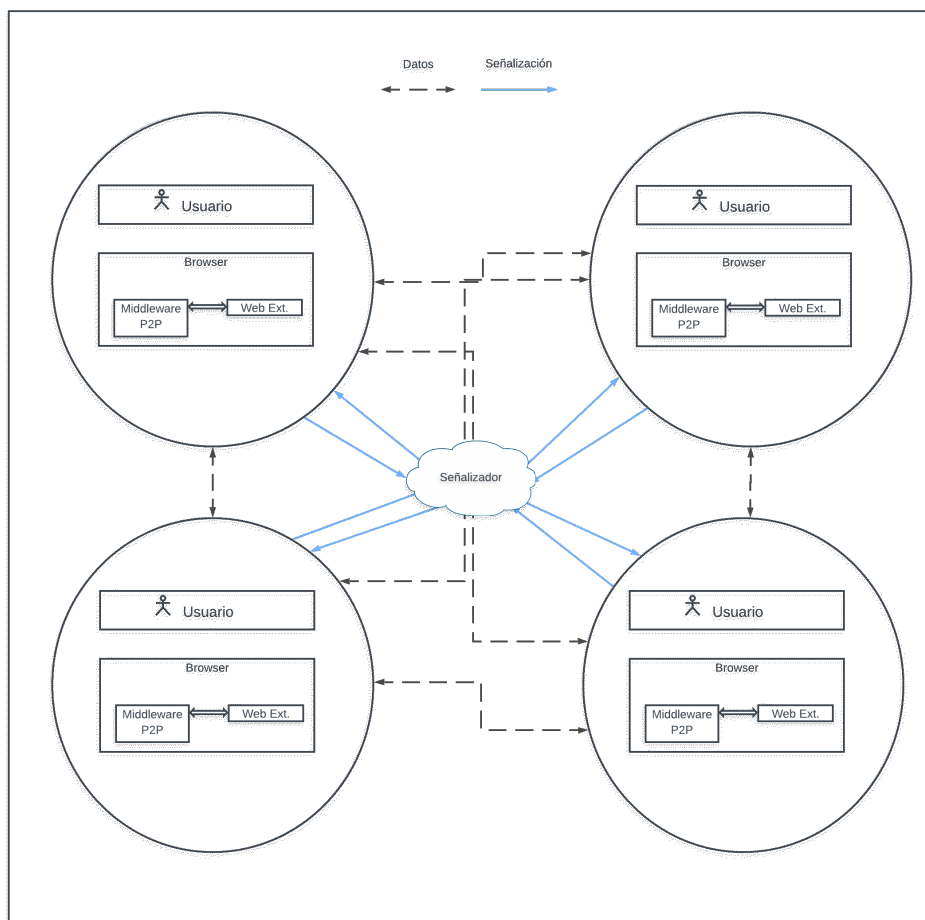


Figura 4.1: Arquitectura general del proyecto

## 4.1. Comunicación e interacción p2p

A continuación se va a representar y detallar como se da la interacción entre los peers conectados a la red. En otra sección se detallará esto con mayor énfasis en cada caso de uso en particular presentado para este trabajo. Los casos de uso propuestos son: 1) Realizar búsquedas en la red P2P, 2) Compartir servicios y comparar resultados de búsquedas pensando en el concepto de Burbuja de filtrado.

Inicialmente los usuarios instalan la extensión web del middleware y a su vez la extensión web para la creación de los servicios de búsqueda, llamada “Peerify”. Estas dos extensiones web se conectan entre sí a través de un puerto <sup>2</sup> <sup>3</sup>, un objeto que permite una conexión bidireccional que puede usarse para intercambiar mensajes. De esta manera, ambas extensiones pueden intercambiar mensajes entre si permitiendo que un usuario desde “Peerify” pueda invocar búsquedas, compartir servicios, recibir respuestas, entre otras cosas, desde y hacia los demás usuarios en la red peer-to-peer.



Figura 4.2: Draftp2p: Extensión P2P

<sup>2</sup><https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API/Runtime/Port>

<sup>3</sup><https://developer.chrome.com/docs/extensions/reference/runtime/#type-Port>



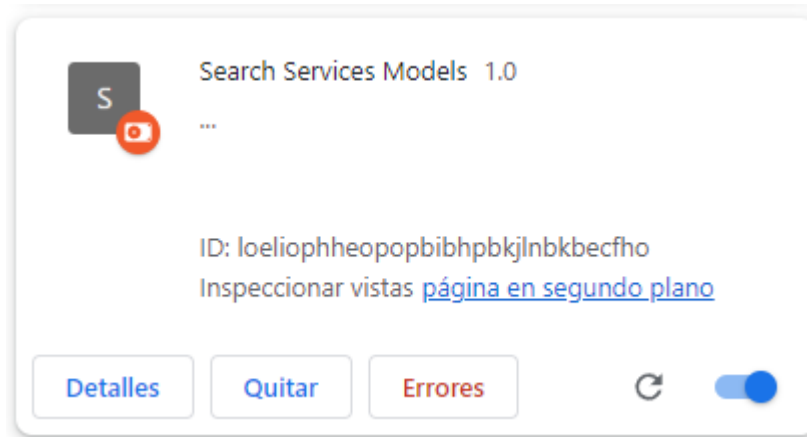


Figura 4.3: Extensión para la creación de servicios de búsqueda

El funcionamiento de ambas extensiones y la red peer-to-peer en simples pasos sería el siguiente:

1. Instalar ambas extensiones en el navegador.
2. Se genera una conexión entre las extensiones instaladas.
3. La extensión web del middleware se conecta a la red peer-to-peer.
4. El usuario, desde “Peerify”, invoca una búsqueda.
5. “Peerify” envía un mensaje por el puerto a la extensión del middleware, y esta envía el pedido de búsqueda a la red peer-to-peer (a todos los peers o solo a algunos elegidos previamente).
6. El middleware recibe respuestas y las reenvía por el puerto a “Peerify”

## 4.2. Middleware P2P

Para implementar la comunicación P2P se utilizó la extensión web que funciona como un middleware y se encarga de las conexiones entre los usuarios que la tengan instalada. Esta extensión web fue desarrollada en “Colaboración P2P para ambientes de desarrollo de usuario final basados en navegadores web” [2].

La conexión entre el middleware y “Peerify”, como se mencionó anteriormente se da a través de un puerto que permite una conexión entre extensión

web a extensión web por el cual se intercambian mensajes. Sin embargo, la lógica donde se crea este puerto y se gestionan los envíos de mensaje, entre otras cosas de la comunicación, se encuentra en un “conector”.

Este “conector” es simplemente un entorno de trabajo o marco de trabajo<sup>4</sup> (“framework”) que facilita el trabajo para la comunicación entre las extensiones web. Este framework es un archivo javascript<sup>5</sup> también propuesto en el trabajo [2]. La figura 4.2 muestra la extensión del middleware instalada.

---

<sup>4</sup>[https://developer.mozilla.org/en-US/docs/Learn/Tools\\_and\\_testing/Client-side\\_JavaScript\\_frameworks/Introduction](https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/Introduction)

<sup>5</sup>[https://developer.mozilla.org/es/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript)

## Capítulo 5

# Herramientas

En esta sección se describirán con mayor detalle las herramientas mencionadas en la sección previa, haciendo una demostración de su uso a través de un ejemplo simple. Se busca mostrar el alcance y las funcionalidades de las herramientas. Posteriormente en la siguiente sección se darán ejemplos mas puntuales de posibles casos de uso propuestos en este trabajo , mostrando al lector el posible potencial y permitiéndole de esta manera imaginar otros casos de uso probables.

### 5.1. “A data service layer for web browser extension”

En [19] se desarrolla e implementa una evolución de ANDES, esta herramienta desarrollada se podría considerar al “núcleo” de este trabajo. En esta herramienta se basa el trabajo. La evolución de ANDES es una extensión de navegador que posee “una capa de servicio de datos diseñada específicamente para extensiones de navegador”, con ella se logra encapsular los servicios de búsqueda de sitios web en “search APIs” reutilizables orientadas a objetos. Esta herramienta te permite abstraerte de la complejidad de las búsquedas, la recopilación de resultados, y su posterior procesamiento, a su vez gracias al uso de EUD (End User Development) permite crear estas search APIs sin necesidad de conocimientos técnicos, se crean desde una interfaz y haciendo clicks.

En las siguientes figuras se puede observar a la evolución de ANDES planteada en [19] con mayor detalle y como es el proceso de la creación de las search APIs y su posterior uso, dando como ejemplo uno de los casos de usos planteados en su desarrollo. Para mayor información y detalle se puede

recurrir a [19].

En la figura 5.1 se pueden observar los pasos del proceso para poder crear una nueva search API de un servicio de búsqueda, en el ejemplo se usa Springer<sup>1</sup> como servicio de búsqueda para la search API. Los pasos son los siguientes:

1. Estar en la pantalla de los resultados del servicio de búsqueda para el cual se quiere crear la nueva search API, una vez allí hacer click en el ícono de la extensión web.
2. Se abrirá un modal de bienvenida donde se tendrán dos opciones, hacer click en la opción de “Services”.
3. Se cambiará a una pantalla en el modal donde se mostrarán los servicios de búsqueda almacenados en el navegador del usuario junto las opciones disponibles para estos servicios, y un botón para crear un nuevo servicio (“New service”).
4. Hacer click en el botón que dice “New service” y darle un nombre al nuevo servicio.
5. Lo próximo es indicarle a la herramienta desarrollada en [19] en que parte del sitio web se encuentra el campo de entrada de texto para la búsquedas en el servicio a agregar.
6. Luego hay que indicarle donde se encuentran los resultados de la búsqueda que hicimos.
7. Por último, ya habiendo marcado donde se encuentran los resultados, la herramienta pide indicar las propiedades de los resultados (al menos una) junto con el tag semántico al que estarán relacionados los resultados de este nuevo servicio de búsqueda.
8. Finalmente, se presiona en “Finish” y se guarda el nuevo servicio de búsqueda.

---

<sup>1</sup><https://link.springer.com/>

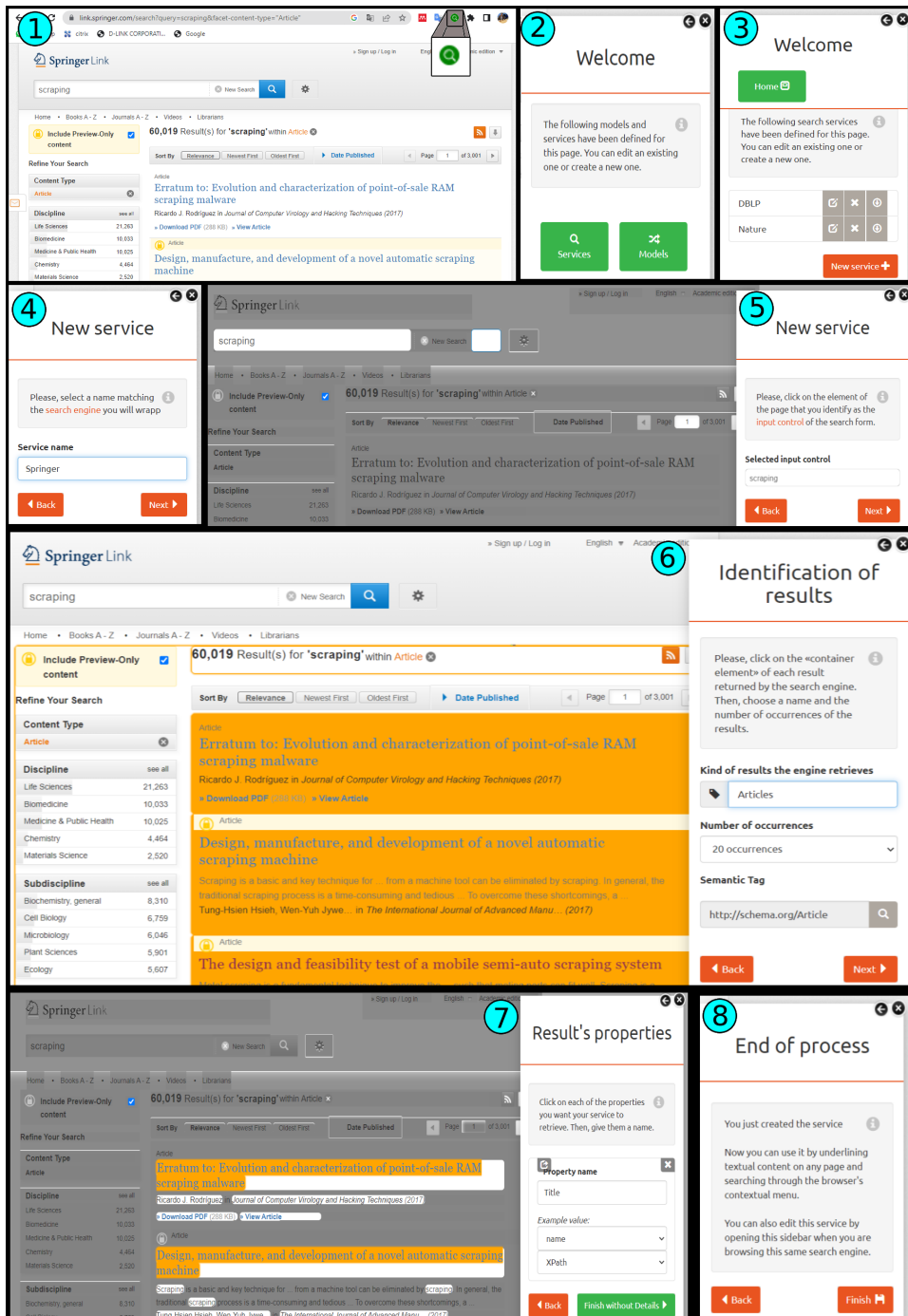


Figura 5.1: Creación de una search API para el servicio de búsqueda de Springer

Ya se tiene un servicio de búsqueda nuevo en la herramienta, lo que se mostrará a continuación es como usarlo a través de la creación de modelos de búsqueda que utilicen los servicios de búsqueda creados. Posteriormente se realizará una búsqueda de prueba con este modelo de búsqueda y se mostrarán resultados de la misma.

Los pasos son los siguientes:

1. Hacer click en el ícono de la extensión web.
2. Se abrirá un modal de bienvenida donde se tendrán dos opciones, hacer click en la opción de “Models”.
3. Se cambiará a una pantalla en el modal donde se mostrarán los modelos almacenados en el navegador del usuario junto las opciones disponibles para estos, un botón para crear un nuevo modelo (“New model” o “Nuevo Modelo”), además habrá una opción para importar modelos.
4. Hacer click en el botón que dice “Nuevo Modelo” y darle un nombre al nuevo modelo.
5. Lo próximo es indicarle a la herramienta que servicios de búsqueda se quieren utilizar para el nuevo modelo. Estos servicios de búsqueda se pueden relacionar entre sí por distintos atributos que tengan definidos.
6. Para agregar un servicio de búsqueda al modelo se puede arrastrar y soltar al servicio deseado en la “caja” pintada de gris.
7. Luego, ya habiendo arrastrado los servicios deseados para este nuevo modelo se pasa al siguiente paso haciendo click en “siguiente”.
8. Por último, desde este último paso se puede probar el nuevo modelo. Los resultados de la búsqueda se mostrarán ahí mismo en el modal del último paso.

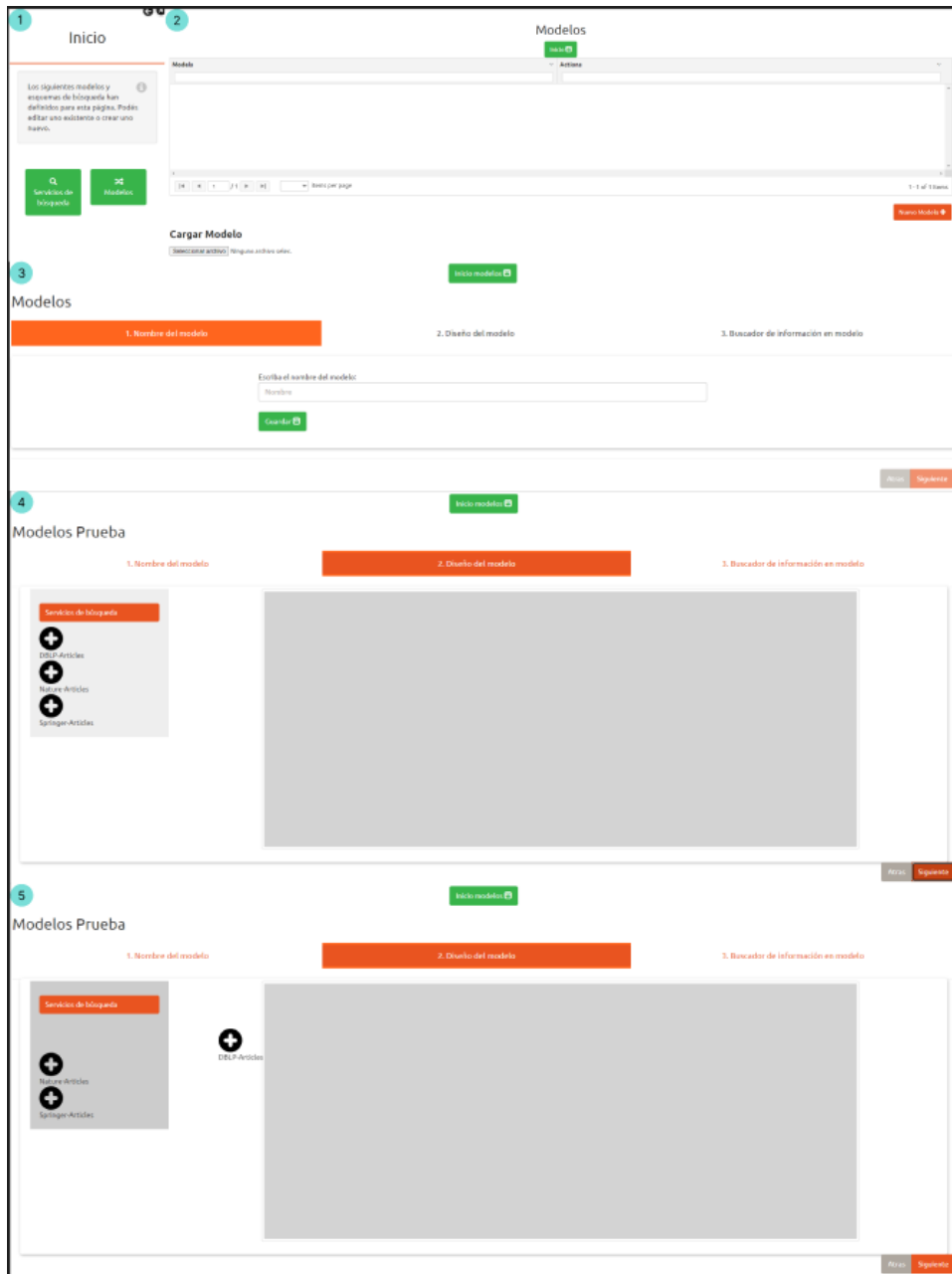


Figura 5.2: Ejemplo de creación de modelo en la herramienta de la evolución de ANDES

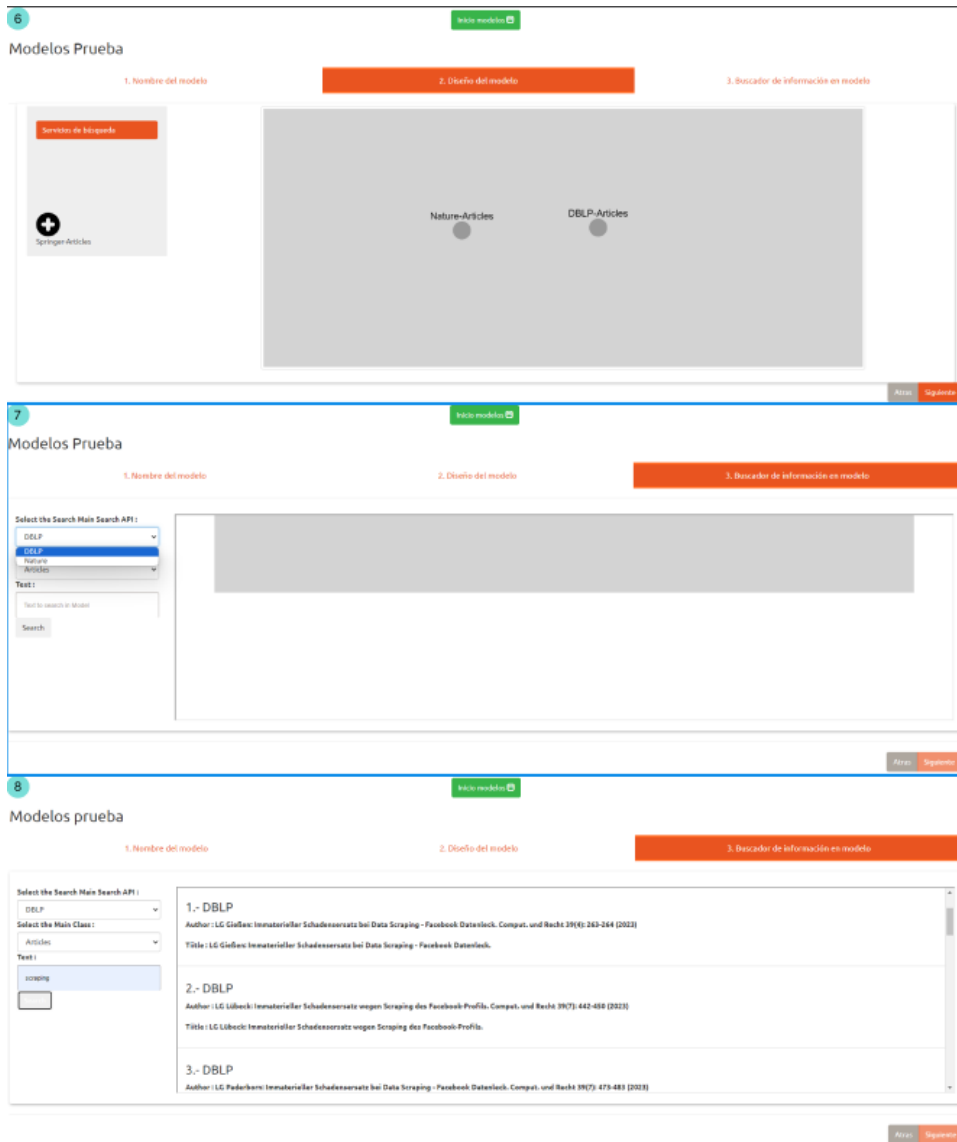


Figura 5.3: Ejemplo de creación de modelo y búsqueda utilizando el mismo

En la figura 5.3 se realiza la misma búsqueda que se hizo en la prueba al finalizar la creación del modelo, mostrando que los resultados que figuran tanto en el servicio de búsqueda utilizado como prueba así como en la prueba del modelo son los mismos.



The screenshot shows the DBLP (computer science bibliography) search interface. At the top, the logo 'дблп' and 'computer science bibliography' are visible, along with a search bar containing the term 'scraping'. Below the search bar, the search results are displayed, showing 138 matches found. The results are filtered for the year 2023 and list several publications related to data scraping and Facebook data leaks.

дблп  
computer science bibliography  
Stop the war!

Search dblp  
powered by CompleteSearch, courtesy of Hannah Bast, University of Freiburg

> Home

Publication search results

found 138 matches

2023

- LG Gießen: Immaterieller Schadensersatz bei Data Scraping - Facebook Datenleck. Comput. und Recht 39(4): 263-264 (2023)
- LG Lübeck: Immaterieller Schadensersatz wegen Scraping des Facebook-Profiles. Comput. und Recht 39(7): 442-450 (2023)
- LG Paderborn: Immaterieller Schadensersatz bei Data Scraping - Facebook Datenleck. Comput. und Recht 39(7): 473-483 (2023)
- K. Sukanya Varshini, R. Annie Uthra: Extraction of Meaningful Information from Unstructured Clinical Notes Using Web Scraping. J. Circuits Syst. Comput. 32(3): 2350041:1-2350041:19 (2023)
- York Yannikos, Julian Heeger, Martin Steinebach: Scraping and Analyzing Data of a Large Darknet Marketplace. J. Cyber Secur. Mobil. 12(2): 161-186 (2023)

Figura 5.4: Resultados de búsqueda en el sitio DBLP<sup>3</sup> para el concepto de “scraping”

Por último, en la figura 5.5 se puede ver a modo de ejemplo como funciona el proceso para realizar una búsqueda colaborativa en la red p2p utilizando la herramienta desarrollada.

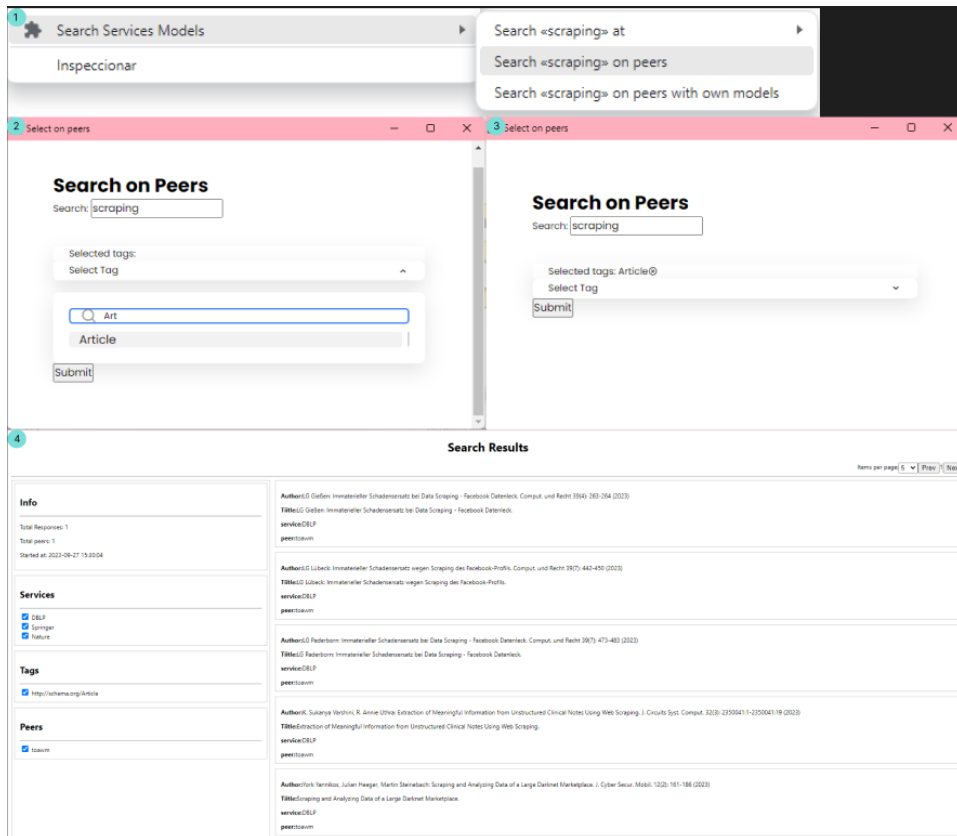


Figura 5.5: Ejemplo de búsqueda en la red p2p con la herramienta extendida en este trabajo

## 5.2. Middleware P2P

Como se mencionó anteriormente la comunicación P2P se consigue gracias a una extensión web que funciona como un middleware y se encarga de las conexiones entre los usuarios de la red p2p. La conexión entre el middleware y “Peerify”, esta conexión se da a través de un puerto que permite una conexión entre extensiones web por el cual se intercambian mensajes. Para la comunicación entre el middleware y “Peerify” se utiliza un framework que facilita el desarrollo y la integración del middleware a una extensión de navegador. En este se encuentra la lógica para el intercambio de mensajes de extensión a extensión, el inicio de la conexión de extensión a extensión, entre otras cosas.

Al middleware no se le realizaron grandes cambios que impacten en lo visual, sino que tuvo modificaciones en el funcionamiento interno del mismo tratando de adaptarlo al uso de la nueva herramienta desarrollada en este trabajo. Algunos de estos cambios son la adición de algunas funciones como por ejemplo para poder saber el nombre de los peers dentro de la red, adición de posibles mensajes que entren y salgan por el middleware, entre otros.

### 5.3. Configuración inicial

Es importante mencionar ciertos pasos, que suceden y también hay que hacer manualmente, al momento de instalar las extensiones web en el navegador y mostrar los pasos a seguir básicos para poder tener una conexión exitosa y tener la posibilidad de recibir y/o compartir servicios en la red peer-to-peer.

Ya mencionadas ambas extensiones y su instalación, lo que se hará ahora es detallar algunos pasos a realizar para dejar en evidencia como dar los posibles primeros pasos con las herramientas para dejarlas preparadas para su posterior uso, y además será útil conocer este proceso más adelante para los ejemplos de los casos de uso a los que fueron aplicados las herramientas en el desarrollo de este trabajo.

Algunas de las siguientes cosas que se mencionarán en esta configuración inicial serán: 1. La conexión a la red 2. La conexión entre las extensiones, 3. La comunicación entre los peers, 4. El pasaje de mensajes para compartir o recibir servicios. 5. La configuración para elegir entre tener una lista blanca (“whitelist”) o una lista negra (“blacklist”) de URLs.

El primer paso es instalar la extensión del Middleware P2P, ya que esta es la que nos permite conectarnos a la red y entablar una comunicación con los peers en la red. Lo próximo es instalar “Peerify”, al momento de instalar “Peerify” automáticamente en segundo plano se hace una conexión interna entre esta y el middleware. Con las conexiones iniciales y la conectividad a la red ya hechas, a continuación se podría crear un nuevo servicio de búsqueda y luego crear un modelo de búsqueda que utilice este servicio, o de igual manera importar servicios de búsqueda o modelos de búsqueda a partir de un archivo JSON<sup>4</sup> gracias a la posibilidad que nos brinda la herramienta desarrollada en [19]. Si bien la creación de servicios de búsqueda o modelos de búsqueda no es necesaria, al lector le será de utilidad tener esto en cuenta para los casos de uso que se explicarán más adelante. Lo próximo es comunicarse con los demás peers que puedan estar conectados en

---

<sup>4</sup><https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON>

la red peer-to-peer, de manera que se puedan compartir nuestros servicios de búsqueda, o realizar alguna búsqueda con los servicios de otros peers. Este paso se puede llevar a cabo simplemente compartiendo los servicios de búsqueda propios con los demás peers de la red, en la Figura 5.6 se puede observar esta opción, y en caso que se desee, se pueden ver los servicios que han sido compartidos por otros usuarios a nosotros haciendo click en la opción mostrada en la figura que dice “Show peer’s services”.

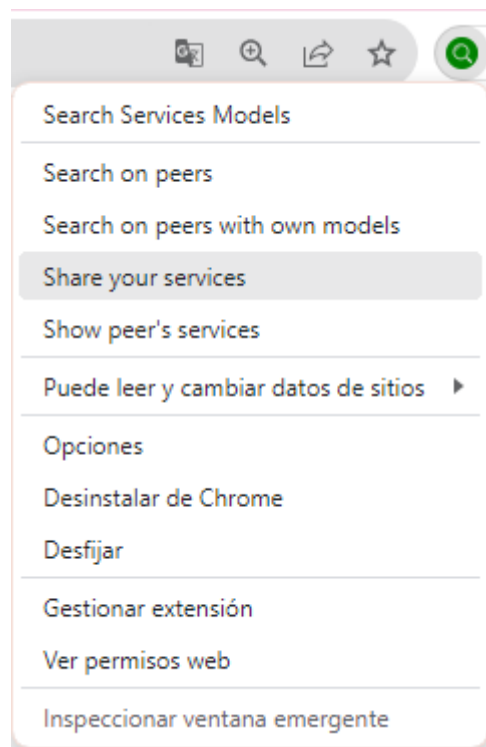


Figura 5.6: Compartir servicios a los demás peers en la red peer-to-peer.

Cabe mencionar que cuando se recibe un mensaje de un servicio compartido por otro peer en la red, internamente en “Peerify” se guarda este o estos servicios recibidos en una estructura de datos. Asimismo, con esta estructura de datos se tiene la posibilidad de hacer un “mapeo” (una relación o correspondencia de datos con los servicios) entre los servicios disponibles y sus tags semánticos con la finalidad de tener disponible esta información al momento de realizar los distintos tipos de búsquedas planteados en este trabajo.

Por último, en “Peerify” a modo de seguridad se pensó darle la opción al usuario para poder bloquear URLs de sitios de web a las que no desee permitir ser consultadas por otras peers en la red ó como contraparte darle la opción de solo permitir ciertas URLs de sitios web por las que puedan realizar consultas los peers de la red. Esto se pensó para evitar que un peer con intenciones negativas o maliciosas pueda enviar un pedido de búsqueda a una URL que, visitada por el peer pueda generarle algún mal al mismo, por ejemplo el sitio web de un banco donde el peer tenga una sesión iniciada.

Para configurar esto hay que dirigirse a las opciones de la extensión web, en Google Chrome se puede lograr esto haciendo click derecho en el ícono de la extensión web y luego haciendo click izquierdo en la opción que dice “Opciones”, esto se puede observar en la Figura 5.7. En Firefox este proceso se hace de manera muy similar, también haciendo click derecho en el ícono de la extensión web y luego seleccionando con click izquierdo la opción que dice “Administrar extensión”, como se puede ver en la Figura 5.8. Luego como se muestra en la Figura 5.9 habría que seleccionar si se desea utilizar una “whitelist” o una “blacklist”, y por último agregar las URLs de los sitios web a permitir o bloquear.

Por último al hacer click izquierdo en el botón de “Guardar” se guardará la configuración y se mostrará un mensaje exitoso de que los datos han sido guardados. Si se quisiera modificar esta configuración, ya sea para cambiar de opción, o agregar o quitar URLs se pueden repetir estos pasos para llegar a este modal de configuración nuevamente.

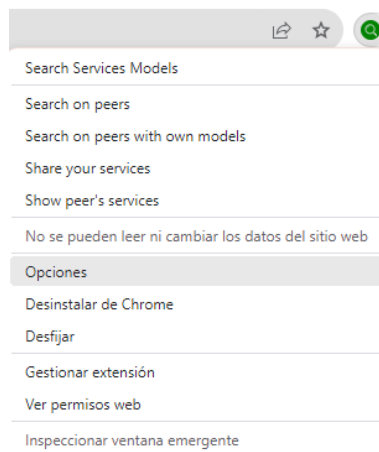


Figura 5.7: Proceso para dirigirse a la configuración de las opciones de la extensión web en Google Chrome.

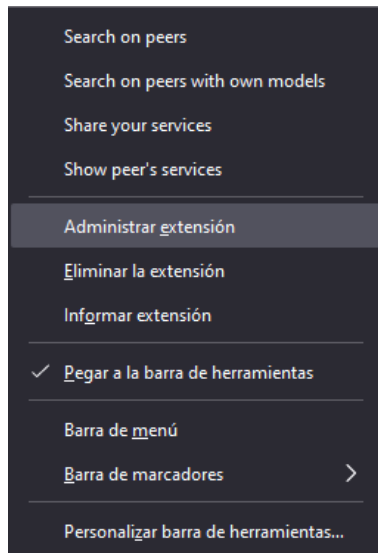


Figura 5.8: Proceso para dirigirse a la configuración de las opciones de la extensión web en Firefox.

## Opciones de la Extensión

- Usar una Blacklist
- Usar una Whitelist

Ingresar URLs (una por línea):

Guardar

Figura 5.9: Modal para configurar la whitelist o blacklist de la extensión web.

## Capítulo 6

# Casos de uso

En esta sección se explorarán los casos de estudio planteados para probar y analizar la herramienta desarrollada. Los casos que analizaremos son: 1) Realizar búsquedas en la red P2P, 2) Compartir servicios y comparar resultados de búsquedas.

El primer caso se centra en las búsquedas en la red P2P, que es una forma descentralizada de compartir información. Este caso sienta las bases para el desarrollo de este trabajo, ya que implica la integración de búsquedas en la red P2P y, además, sirve como punto de partida para el segundo caso. El segundo caso aborda cómo compartir servicios y analizar las diferencias en los resultados obtenidos. En otras palabras, busca realizar búsquedas a través de la red P2P mientras se comparten los modelos de búsqueda específicos de los usuarios que invocan la búsqueda en la red. Este caso de uso se concibe en el contexto del concepto de “burbuja de filtrado”, con el objetivo de identificar diferencias en los resultados entre pares, aunque utilizando el mismo modelo de búsqueda.

Antes de iniciar con el detalle puntual de cada caso de uso planteado se quiere dejar en claro algunas decisiones de diseño e implementación que se han tenido en cuenta y cuales se tomaron como decisión final a lo largo del desarrollo de este trabajo. Asimismo se detallará el proceso inicial que servirá para ambos casos de uso sobre el funcionamiento de la herramienta desarrollada.

Un tema es el manejo de los servicios en la red p2p, es decir, como se llegan a conocer los servicios de los usuarios dentro de la red, quienes comparten sus servicios, cómo se comparten los servicios, y cómo son utilizados estos servicios. Esto en principio se había pensado y hecho para que al momento de que un usuario se conecte a la red sus servicios se envíen como

broadcast a todos los peers dentro de la red, para que cada usuario conozca los servicios que provee el usuario recientemente conectado, sin embargo, se modificó brindándole la opción a cada usuario de compartir sus servicios en el momento que el lo deseara haciendo click en el icono de la extensión (se puede observar esta opción en la figura 6.1) y que cada vez que un nuevo peer se conecta y comparte sus servicios los demás peers en la red pueden responderle con sus propios servicios al peer en particular.

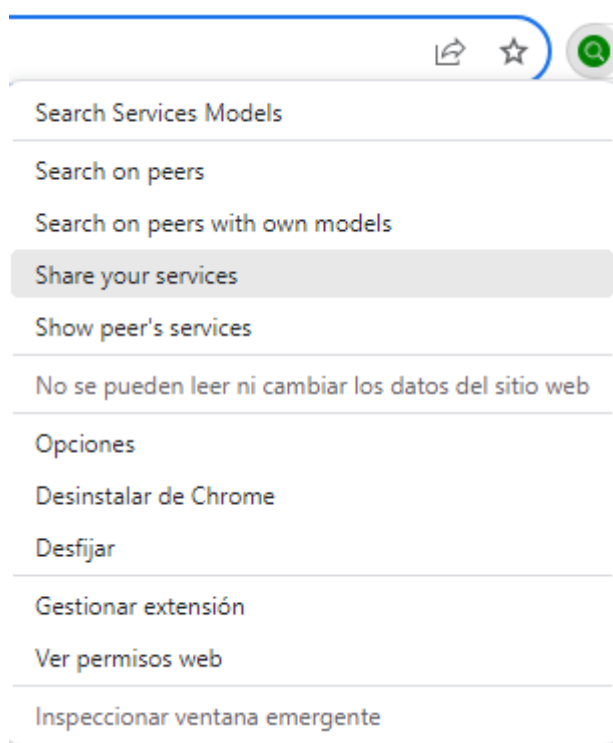


Figura 6.1: Opciones a través del ícono de la extensión (browser\_action)<sup>1</sup>

Por último, a continuación se explicarán ciertos pasos a seguir para conectarse a la red peer-to-peer y poder realizar las búsquedas peer-to-peer.

Lo primero es tener la extensión web “Peerify” desarrollada en este trabajo y la del Middleware P2P desarrollada en [2], previamente mencionada en el capítulo 4 de “Enfoque y Arquitectura”, instaladas. Luego, una vez conectados a la red (de este proceso se encarga la extensión del Middleware P2P) y teniendo ambas extensiones conectadas entre sí (que se implementó para que se conecten de manera automática) ya podemos compartir ser-



vicios, invocar búsquedas, y participar en búsquedas requeridas por otros peers en la red. Por último, para poder realizar búsquedas tendrías que desde el menú desde el ícono de la extensión invocar una búsqueda con alguna de las dos opciones mostradas en la figura 6.2.

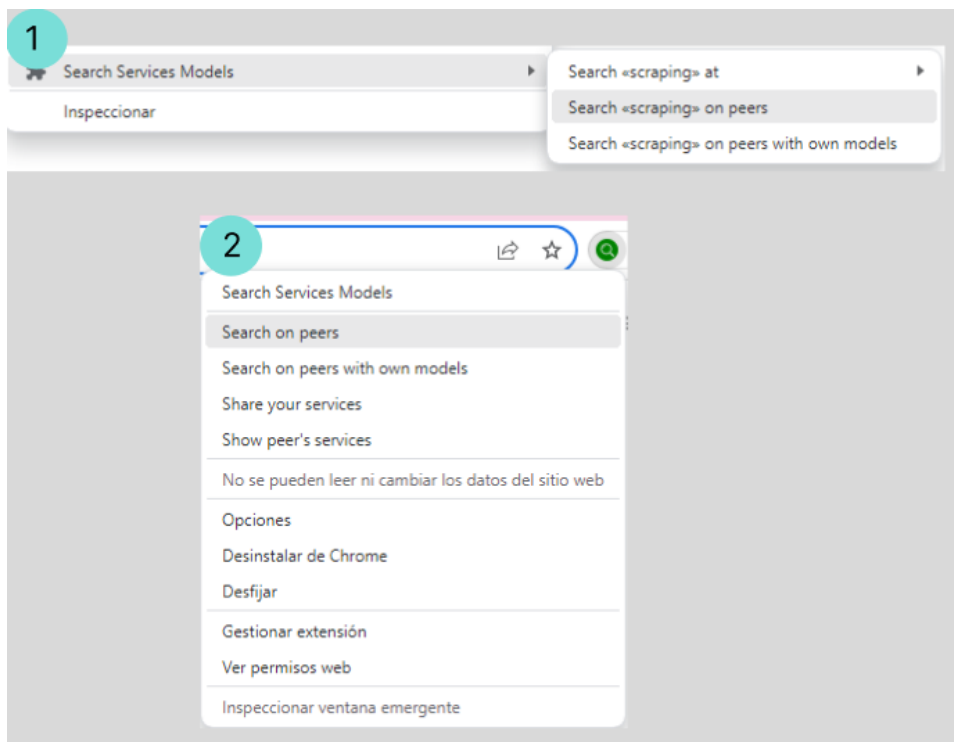


Figura 6.2: Opciones de búsqueda en el menú contextual y en el menú del ícono de la extensión (browser\_action)

## 6.1. Integración de búsquedas P2P

La integración de búsquedas en la red peer-to-peer representa tanto el primer caso como el fundamento principal de este trabajo. Este caso es la adaptación de las búsquedas planteadas en [19] junto con la colaboración de peers en un red peer-to-peer para poder realizar búsquedas de manera colaborativa.

A continuación se ilustrarán los pasos a seguir para poder invocar una búsqueda en la red p2p, demostrando el primer caso de uso de “Integración de búsquedas P2P”.

Se pondrá de ejemplo la búsqueda con la palabra “mamífero” y se invocará una búsqueda en la red donde están conectados 3 peers en total, incluyendo al que la invoca. Los dos peers adicionales al que invoca la búsqueda ambos poseen un modelo de búsqueda configurado junto con servicios de búsqueda para sitios web de “Springer”<sup>2</sup> y “Nature”<sup>3</sup>. El peer que invoca la búsqueda ahora en adelante será referido como “toawm”.

“toawm” quiere invocar una búsqueda con el concepto “mamífero.”<sup>en</sup> la red, el primer paso que deberá realizar “toawm” es abrir el pop-up donde podrá ingresar el texto a buscar. Esto puede lograrlo de dos maneras, la primera manera es haciendo click derecho sobre el ícono de la extensión “Peerify” y después seleccionar la opción de “Search on peers” que se puede observar en la Figura 6.3. La segunda manera es a través de la selección de un texto, en este caso “mamífero”, y haciendo click derecho sobre el texto seleccionado, esta acción abrirá un menú contextual donde se mostrará una opción para poder hacer una búsqueda en la red peer-to-peer. Los pasos de la segunda manera se pueden observar en la Figura 6.4 y en la Figura 6.5.

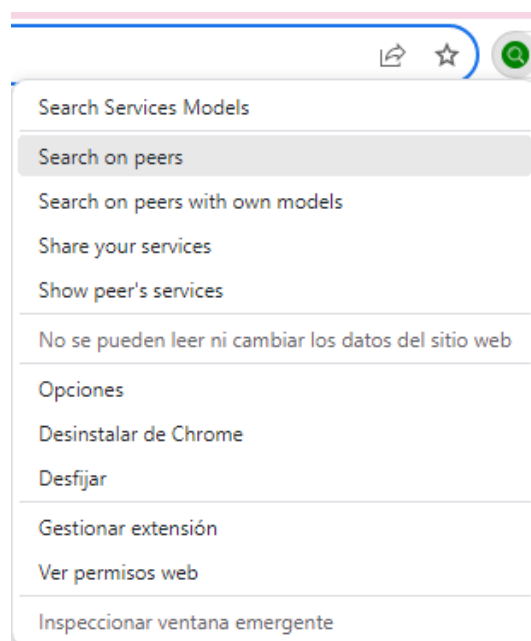


Figura 6.3: Opciones de búsqueda desde el ícono de la extensión (browser\_action<sup>5</sup>)

<sup>2</sup><https://link.springer.com/>

<sup>3</sup><https://www.nature.com/>

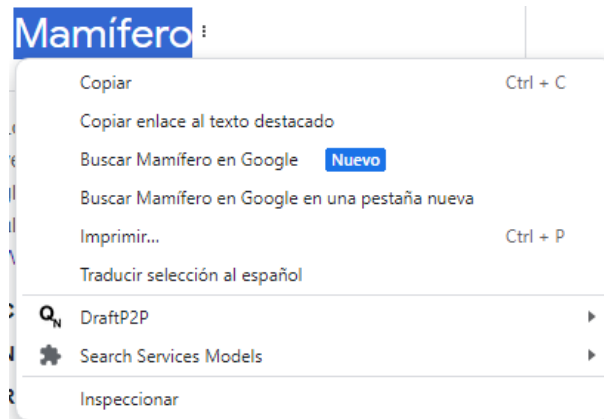


Figura 6.4: Menú contextual al querer iniciar una búsqueda

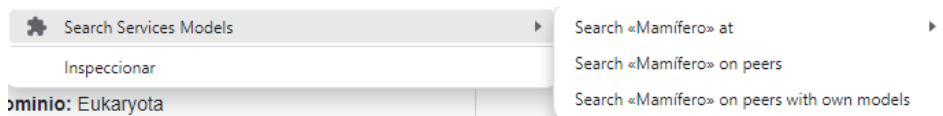


Figura 6.5: Opciones de búsqueda en el menú contextual

Los próximos pasos para concluir con la búsqueda serán seleccionar los tags semánticos para la búsqueda, confirmar la búsqueda, y por último ver los resultados recibidos por los demás peers.

En la Figura 6.6 se muestra la pantalla pop-up para la pre-visualización de la búsqueda, en esta pantalla se puede ver que hay un campo de entrada de texto editable donde se encuentra el texto que se quiere buscar. Además, hay un campo selector de tags semánticos que serán utilizados para filtrar los peers que participarán de la búsqueda en la red p2p. En este campo, debido a la cantidad de tags que pueden llegar a existir, se puede escribir para poder filtrar por el tág semántico deseado sin tener que “scrollar” en busca de los mismos.

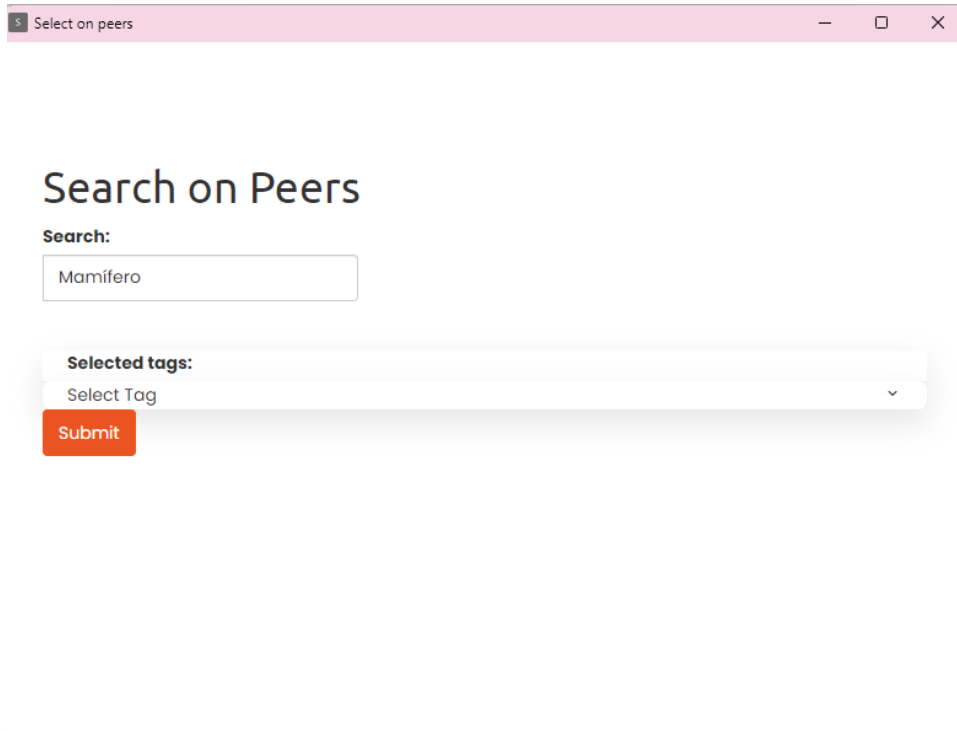


Figura 6.6: Pop-up para invocar la búsqueda en la red P2P.

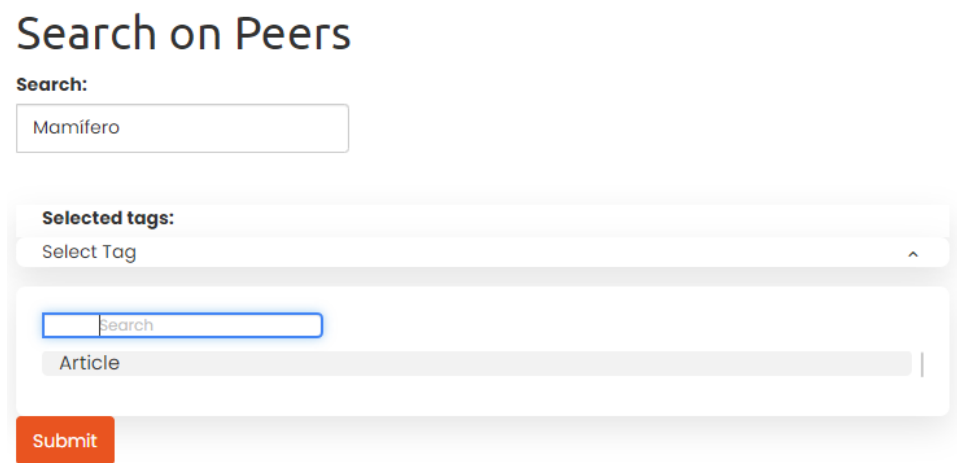


Figura 6.7: Selección de posibles tags semánticos para la búsqueda a realizar



no todos van a responder a la misma velocidad o tiempo, algunos pueden tener menor o mayor cantidad de servicios de búsqueda definidos cuyo tag semántico coincida con el deseado por el peer que ha invocado la búsqueda, así como también puede influir la velocidad de red que tenga cada peer en resolver las búsquedas.

## 6.2. Burbuja de filtrado

El segundo escenario se centra en la compartición de los modelos de búsqueda definidos por los usuarios y en el análisis de las disparidades en los resultados obtenidos. En otras palabras, se trata de llevar a cabo búsquedas a través de la red P2P haciendo uso de los modelos de búsqueda específicos de los usuarios que lanzan las consultas en la red. Este caso de uso se enmarca en el contexto del fenómeno conocido como “burbuja de filtrado” y tiene como objetivo detectar discrepancias en los resultados entre los pares, a pesar de utilizar el mismo modelo de búsqueda. A través de este escenario, se pretende destacar las similitudes y diferencias en los resultados que pueden surgir al navegar por la web y realizar búsquedas con un propósito específico.

A continuación se ilustrarán los pasos a seguir para poder invocar una búsqueda en la red p2p, demostrando el segundo caso de uso de “Burbuja de filtrado”.

Los pasos son muy similares a los del primer caso. Se pondrá de ejemplo la búsqueda con la palabra “Argentina” y se invocará una búsqueda en la red donde están conectados 3 peers en total, incluyendo al que la invoca. Para este caso no hace falta que los demás peers en la red tengan definidos servicios o modelos de búsqueda, ya que el peer que invoca la búsqueda enviará por la red su/s modelo/s de búsqueda. El peer que invoca la búsqueda ahora en adelante será referido como “toawm”.

“toawm” quiere invocar una búsqueda con el concepto “Argentina” en la red, el primer paso que deberá realizar “toawm” es abrir el pop-up donde podrá ingresar el texto a buscar. Esto puede lograrlo de dos maneras, la primera manera es haciendo click derecho sobre el ícono de la extensión desarrollada en este trabajo y después seleccionar la opción de “Search on peers with own models” que se puede observar en la Figura 6.9. La segunda manera es a través de la selección de un texto, en este caso “Argentina”, y haciendo click derecho sobre el texto seleccionado, esta acción abrirá un menú contextual donde se mostrará una opción para poder hacer una búsqueda en la red peer-to-peer. Los pasos de la segunda manera se pueden observar en la Figura 6.10 y en la Figura 6.11.

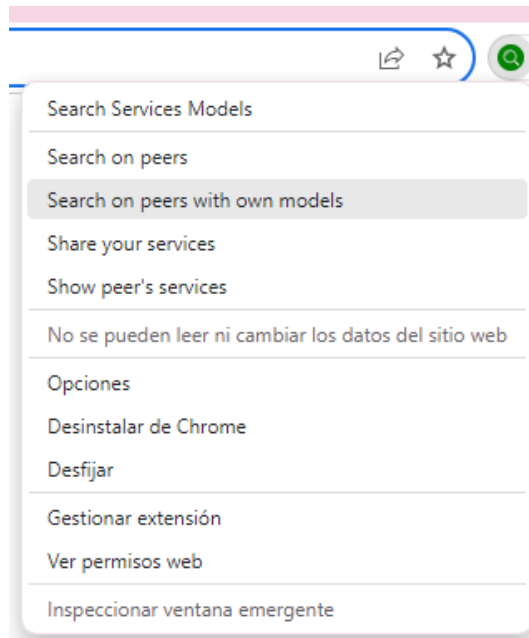


Figura 6.9: Opciones de búsqueda desde el ícono de la extensión (browser\_action <sup>7</sup>)

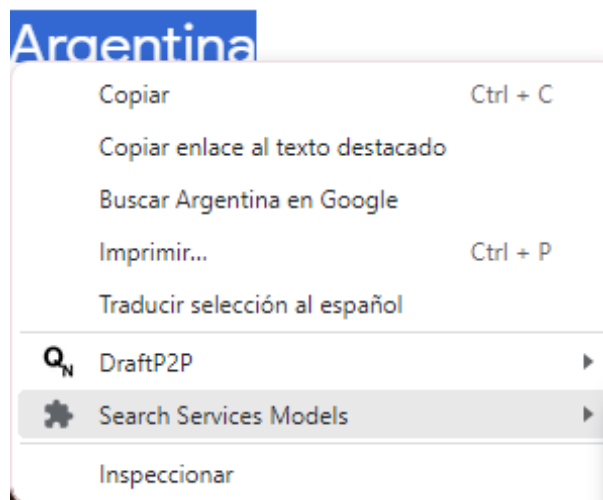


Figura 6.10: Menú contextual al querer iniciar una búsqueda

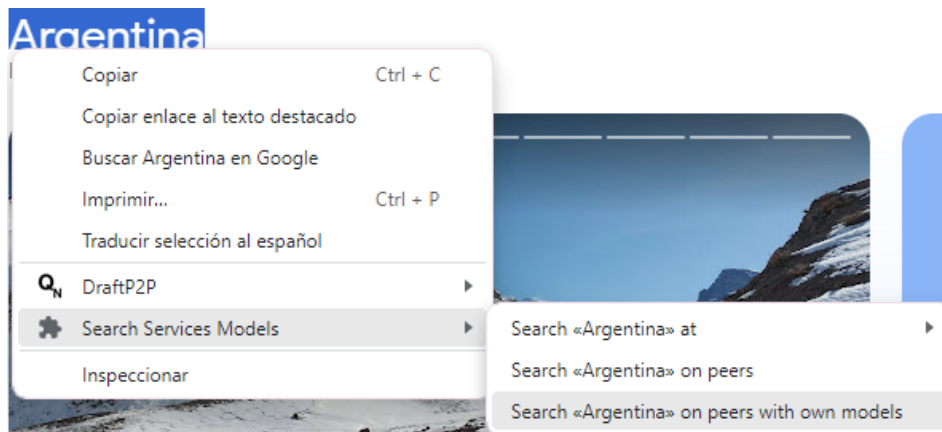


Figura 6.11: Opciones de búsqueda en el menú contextual

El siguiente paso, una vez abierto el pop-up, será seleccionar los modelos de búsqueda e ingresar el texto que se desea buscar. Si se optó por la segunda forma mencionada para abrir el pop-up entonces el campo del texto a buscar ya estará completado con el texto seleccionado en el primer paso, sin embargo, si se desea modificar el texto se puede hacerlo editando el texto dentro de este campo.

En la Figura 6.12 se muestra la pantalla pop-up para la pre-visualización de la búsqueda, en esta pantalla se puede ver el campo de entrada de texto editable donde se encuentra el texto que se quiere buscar. Además, se encuentra un campo selector de modelos de búsqueda que serán enviados a los peers que participarán de la búsqueda en la red p2p. En la Figura 6.13 se filtra el modelo de búsqueda "Buscadores" para mostrar que es posible hacer un filtro en el selector en caso de que haya una gran cantidad de modelos de búsqueda poder elegirlos con mayor facilidad.



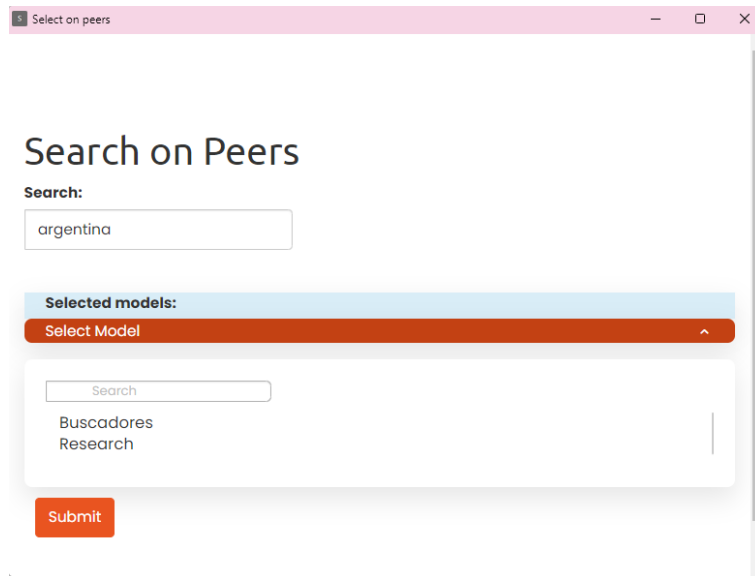


Figura 6.12: Pop-up para invocar la búsqueda en la red P2P.

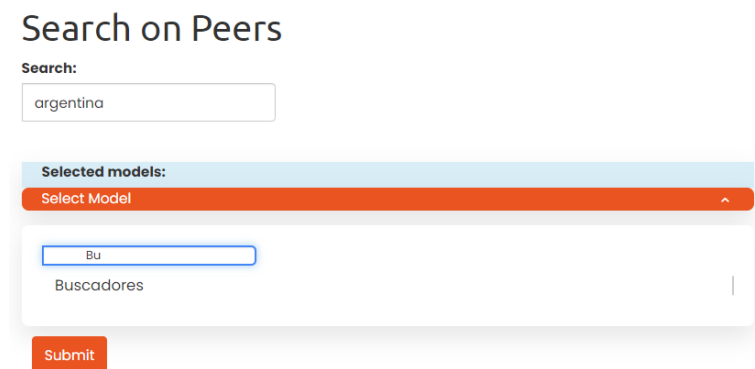


Figura 6.13: Selección de posibles modelos de búsqueda para la búsqueda a realizar

Una vez que “toawn” haya confirmado que quiere buscar “Argentina” y haya seleccionado los modelos de búsqueda deberá hacer click izquierdo en el botón “Submit” de la pantalla pop-up. Esta acción invocará la búsqueda, la herramienta enviará un mensaje a la extensión del middleware p2p la cual distribuirá el mensaje a los peers correspondientes para que realicen

la búsqueda en su navegador. A su vez, se abrirá una pantalla donde se mostrarán los resultados de los peers involucrados en la búsqueda.

Los modelos de búsqueda seleccionados por “toawn” serán enviados como objetos a través de la red peer-to-peer, cada peer que reciba el mensaje de búsqueda por modelos se guardará temporalmente estos modelos de búsqueda recibidos y los utilizará para hacer la búsqueda por el termino recibido en el mensaje. Luego, al finalizar el proceso de búsqueda enviará los resultados y descartará estos modelos recibidos.

En la pantalla de resultados se muestran opciones para poder paginar reduciendo o aumentando la cantidad de resultados a mostrar por página, botones para poder ocultar o mostrar resultados que provengan de servicios de búsqueda en particular, y, para este segundo escenario, se agregaron botones para poder ordenar los resultados por mayor o menor número de ocurrencias. También se muestran datos como la cantidad de peers que participaron en la búsqueda y cuántos respondieron. A continuación se mostrará en las figuras como se ven los resultados, y cómo se pueden ordenar por mayor o menor número de ocurrencias. Las ocurrencias son las veces que se repiten los mismos resultados para los distintos peers en la búsqueda.

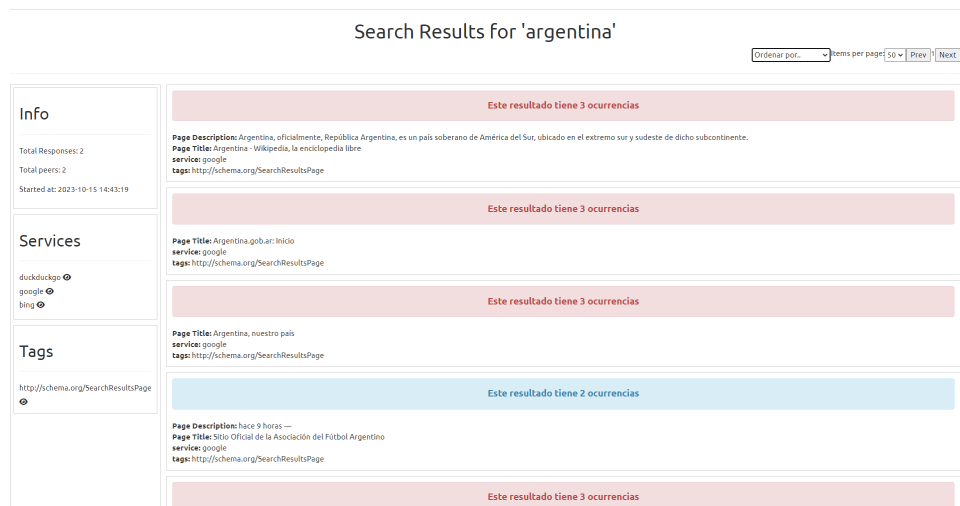


Figura 6.14: Pantalla de resultados de la búsqueda.

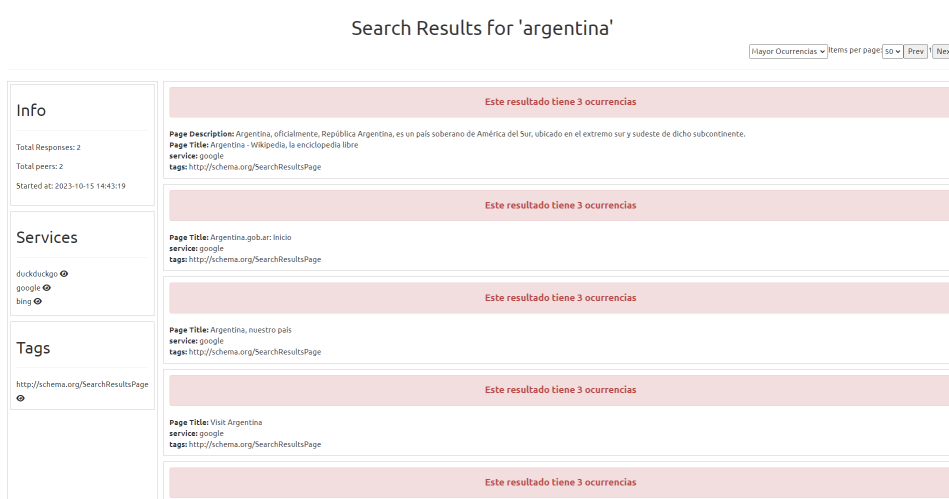


Figura 6.15: Pantalla de resultados de la búsqueda ordenando por mayor número de ocurrencias.

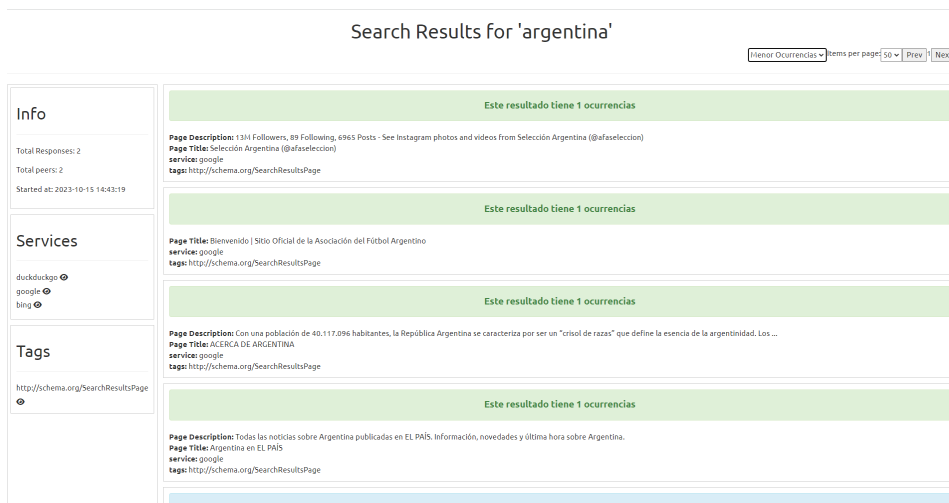


Figura 6.16: Pantalla de resultados de la búsqueda ordenando por menor número de ocurrencias.

Como en el primer escenario, los resultados se van añadiendo uno abajo de otro a medida que llegan respuestas de los peers, ya que no todos van a responder a la misma velocidad o tiempo, puede influir la velocidad de red que tenga cada peer en resolver las búsquedas.

## Capítulo 7

# Conclusiones y trabajo futuro

Este trabajo propone desarrollar una extensión web que mediante el uso de una arquitectura peer-to-peer provea la posibilidad de definir servicios de búsqueda para cualquier sitio web y realizar con estos servicios una búsqueda utilizando los peers conectados a la red peer-to-peer como voluntarios para estas búsquedas.

Como conclusión al trabajo realizado, se logró desarrollar una potente herramienta de definición de APIs de búsqueda (search APIs) basadas en web scraping que permita brindar servicios de datos basándose en una arquitectura peer-to-peer, además, sumado a esta herramienta se logró plantear casos o escenarios de uso como pruebas de concepto para evidenciar el potencial de la misma y algunos posibles escenarios donde podría ser de utilidad.

A lo largo de este trabajo, surgieron algunos desafíos a los que hubo que enfrentarse, empezando por la comunicación peer-to-peer así como lograr la integración de las herramientas utilizadas en este trabajo, también la implementación de nuevas adiciones para la comunicación entre las extensiones de navegador. Para esto se realizó un exhaustivo análisis y estudio tanto al middleware, al framework, y a la herramienta desarrollada en [19], este análisis y estudio tuvo lugar previo y durante el desarrollo de este trabajo. Con respecto al framework, si bien su integración fue sencilla ya que funciona como un conector entre dos extensiones web, analizarlo para entender su funcionamiento y su estructura tomó su tiempo y esfuerzo.

En cuanto a limitaciones en este trabajo, tratando la seguridad en la herramienta desarrollada si bien se planteó una medida de seguridad tal

como lo es la de tener una lista blanca (“whitelist”) de sitios permitidos u optar por tener una lista negra (“blacklist”) para prohibir sitios al momento de recibir mensajes para realizar búsquedas, a futuro se podrían mejorar y/o agregar más medidas para la seguridad de cada peer. Otra limitación a la que no se llegó a resolver en este trabajo es sobre almacenar o dejar un registro sobre las búsquedas que se han realizado, podrían tomarse las pedidas por uno mismo y las solicitadas por los demás peers en la red, de este modo se podría consultar este registro para que cada peer sepa donde se han utilizado sus servicios como voluntario en la red.

En resumen, se lograron cumplir satisfactoriamente todos los objetivos y desarrollos propuestos en este proyecto. Tanto el middleware como el framework desempeñaron sus roles con éxito: el primero proporcionó la necesaria capa de comunicación peer-to-peer, mientras que el segundo facilitó la conexión entre las extensiones web, dotándolas de funcionalidad P2P.

Además, se llevó a cabo un análisis y estudio del desarrollo presentado en [1]. Se destaca la creación de la herramienta “Peerify”, y se espera que esta herramienta no solo sea útil en el ámbito cotidiano, sino que también tenga un potencial valor académico, pudiendo ser empleada en investigaciones y estudios académicos.

Para finalizar y dejar una línea al lector que quiera continuar con este trabajo, a continuación se mencionan algunos de los posibles trabajos para realizar a futuro:

1. Agregar a “Peerify” la posibilidad de guardar un registro o historial donde se almacenen las búsquedas donde participó cada peer y los servicios de búsqueda involucrados, entre otras cosas.
2. Agregar una capa de seguridad a la interacción con la red peer-to-peer.
3. Consideraciones con sitios web en los que algún país donde se encuentren peers no tengan acceso al mismo.
4. Realizar una evaluación de “Peerify” con personas en un entorno preparado para distintos tipos de prueba.

# Bibliografía

- [1] G. Bosetti, A. Tacuri, I. Gambo, S. Firmenich, G. Rossi, M. Winckler, and A. Fernandez, “Andes: an approach to embed search services on the web browser,”
- [2] R. A. Gonzalez, *Colaboracion P2P para ambientes de desarrollo de usuario final basados en navegadores web*. PhD thesis, Universidad Nacional de La Plata, Facultad de Informática, 2020.
- [3] K. Nath, S. Dhar, and S. Basishtha, “Web 1.0 to web 3.0 - evolution of the web and its various challenges,” 2014.
- [4] K. Jannes, B. Lagaisse, and W. Joosen, “The web browser as distributed application server: Towards decentralized web applications in the edge,”
- [5] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge UP, 2009.
- [6] A. R. Arguedas, C. T. Robertson, R. Fletcher, and R. K. Nielsen, “Echo chambers, filter bubbles, and polarisation: a literature review,”
- [7] S. de S Sirisuriya, “A comparative study on web scraping,”
- [8] M. A. Khder, “Web scraping or web crawling: State of art, techniques, approaches and application,”
- [9] T. BERNERS-LEE, J. HENDLER, and O. LASSILA, “The semantic web,” 2001.
- [10] Y. L., *A Developer’s Guide to the Semantic Web*. 2014.
- [11] S. Firmenich, G. Bosetti, G. Rossi, and M. Winckler, “Web objects ambient: an integrated platform supporting new kinds of personal web experiences,”

- [12] L. F. G. Sarmenta, “Volunteer computing,” 2001.
- [13] T. Wang, C. Zhao, Q. Yang, , S. Zhang, and S. C. Liew, “Ethna: Analyzing the underlying peer-to-peer network of ethereum blockchain,”
- [14] R. Schollmeier, “A definition of peer-to-peer networking for the classification of peer-to peer architectures and applications,”
- [15] P. Backx, T. Wauters, B. Dhoedt, and P. Demeester, “A comparison of peer-to-peer architectures,”
- [16] B. Yang and H. Garcia-Molina, “Designing a super-peer network,”
- [17] M. Bender, T. Crecelius, S. Michel, and J. X. Parreira, “P2p web search: Make it light, make it fly,” 2007.
- [18] R. Gonzalez, S. Firmenich, A. Fernandez, and G. Rossi, “Collaborative web extensions: a p2p approach,” 2021.
- [19] A. Tacuri, S. Firmenich, G. Rossi, and A. Fernandez, “A data service layer for web browser extensions,”