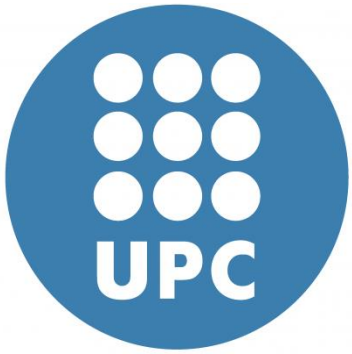

IDENTIFICACIÓN DE SERVICIOS WEB A PARTIR DE TRÁFICO CIFRADO

TREBALL DE FINAL DE GRAU
GRAU EN ENGINYERIA INFORMÀTICA
ESPECIALITAT EN TECNOLOGIES DE LA INFORMACIÓ

Carlos Jiménez Bailén

Facultad de Informática de Barcelona
Director: Pere Barlet-Ros (DAC)

Enero de 2021

Resumen

Este proyecto plantea la creación de una herramienta que identifique los servicios web a los que se conecta un usuario a partir de los third parties (servicios web de terceros a los que se conecta la web) a los que se conecta este servicio web y la experiencia que tiene el usuario con estos servicios. Se ha diseñado la primera herramienta contando con algoritmos de machine learning (aprendizaje automático). La segunda herramienta se basa en saber la experiencia del usuario visitando esos servicios, extrayendo y analizando métricas concretas.

Abstract

This project proposes the creation of a tool that identifies the web services, to which a user connects, from the third parties to which this web service connects and the experience that the user has with these services. The first tool has been designed with machine learning algorithms. The second tool is based on knowing the user's experience visiting those services, extracting and analyzing specific metrics.

Índice

Índice de tablas

Índice de figuras

1. Contextualización y justificación

1.1 Introducción

El Trabajo de Fin de Grado *"Identificación de servicios web a partir de tráfico cifrado"* pertenece a los estudios del Grado en Ingeniería Informática de la Facultad de Informática de Barcelona, Universidad politécnica de Cataluña, en la especialización de Tecnologías de la Información.

El proyecto pretende crear una herramienta de análisis de la red a partir del tráfico cifrado que genera esta.

1.2 Contexto

Hoy en día la actividad de las personas en internet ha aumentado mucho. Esto provoca que los proveedores de conectividad estén siempre investigando para intentar mejorar la experiencia que tienen sus usuarios en la red que ellos les proporcionan. Esto repercute considerablemente en los ingresos de las compañías. Para hacer estos estudios se necesitan personas especializadas en tráfico web y poder acceder a todo este tráfico.

Estas personas que se dedican a analizar la red necesitan que el contenido del tráfico web no esté encriptado para saber a qué servicios web se conectan los usuarios. Nos hemos dado cuenta que esto no debería ser así. La privacidad del usuario es vital para mantener una relación de confianza entre el proveedor y el mismo usuario.

Para ayudar a los proveedores con las dos necesidades previamente mencionadas se nos ocurrió si se podría hacer una herramienta que pudiera identificar el tráfico que genera un usuario manteniendo la confidencialidad de este. Uno de los elementos accesibles del tráfico cifrado son los servicios a los que se conecta una página web. Casi todas las páginas web se conectan con servicios web o con otras páginas para completar o complementar la suya propia. Por ejemplo, cuando Facebook nos muestra que tiempo hará, esta saca la información de algún servicio web de tiempo meteorológico. Estos servicios o páginas web se llaman third parties.

Nuestra premisa es que, estudiando los third parties a los que se conecta una página web, podemos identificar de cual se trata, como una especie de huella digital del servicio. Esto pasa por generar mucho tráfico y ponerlo a estudio. Posteriormente crear un modelo que prediga la web a la que se está conectando el usuario.

Para poder saber la experiencia que tienen los usuarios que navegan por estas redes nos conectaremos a las webs que previamente hayamos identificado. Una vez hecha la conexión haremos un seguido de pruebas para sacar las métricas que nos permitan saber la calidad de experiencia que tendrá una persona que visita esa web en esa red.

1.3 El problema

El principal problema que intenta resolver nuestro proyecto es la posibilidad de identificar los servicios web a partir del tráfico cifrado. Con identificar nos referimos a poder averiguar la url del servicio: si una persona se conecta a Facebook queremos saber la que la conexión es a www.facebook.com.

Después de un estudio previo nos hemos dado cuenta que la identificación de servicios web no es un ejercicio trivial si no que añade complejidad a cualquier aplicación que necesite esta información. Como hemos podido observar no hay ninguna herramienta que pueda hacer esta identificación así que lanzamos este proyecto para poder suplir esta necesidad.

1.4 Funcionamiento de la herramienta

La primera parte del proyecto consiste en crear una herramienta que detecte a que página web se está conectando el usuario. Nuestra solución es que, a partir de los third parties a los que se conecta una página web, podamos averiguar cuál es esta. Para ello capturaremos tráfico y guardaremos las páginas web y sus third parties. Esto lo haremos para hacer un modelo que observa la evolución de estos para determinar, a partir de los third parties, la página web original.

La segunda parte del proyecto es la creación de una herramienta que cuantifique la experiencia que tiene el usuario al navegar por la red. Esto se podrá saber a partir de métricas como puede ser el tiempo que tarda en cargar la página web, el tiempo que tarda en cargar el elemento más grande de la página web o el tiempo que tarda en ser usable esta. La forma de medir la experiencia de usuario no variará dependiendo del servicio al que se conecte ya que esta herramienta se basa en los parámetros comunes en todos las webs.

La parte final de proyecto es combinar ambas herramientas. La intención es que nuestra herramienta final haga un estudio exhaustivo de la red. Este estudio consistiría en identificar en tiempo real las páginas web a las que se conecta un usuario y, a partir de estas, saber que experiencia tiene.

Figura 1: Funcionamiento de la herramienta

El objetivo final de nuestra herramienta es ayudar a los administradores de red a poder monitorear la red, aunque esta esté cifrada.

1.5 Actores implicados

A continuación, se listan los distintos actores implicados, contando las personas interesadas, beneficiadas, afectadas y participantes del proyecto.

- **Desarrollador:** el proyecto constara de un único desarrollador, Carlos Jimenez Bailen. Este desarrollara el proyecto dentro del marco del trabajo de fin de grado de ingeniería informática. El será el encargado del desarrollo, testing y documentación de este proyecto
- **Director del Trabajo de Fin de Grado:** El director Pere Barlet Ros es miembro del departamento de Arquitectura de Computadores y será el supervisor del proyecto.
- **Administradores de red:** Los administradores de red son los principales beneficiarios de las herramientas desarrolladas. Este proyecto se inició para poder ayudar a estas personas a saber el estado de su red más en profundidad.
- **Empresas de telecomunicación:** A un nivel superior que los administradores de red, estas empresas también se podrán beneficiar mucho de nuestras herramientas. Estas empresas también necesitan tener un conocimiento exhaustivo de la experiencia que tienen sus usuarios con la red que le proveen.
- **Entusiastas:** La herramienta que identifica los servicios web a partir del tráfico cifrado será muy apreciada para incorporar en proyectos de entusiastas ya que se resuelve problemas reales.

1.6 Justificación

En los últimos tiempos ha aumentado el tráfico web, por eso creemos que esta herramienta será muy bien acogida para poder monitorear la red.

Después de una investigación exhaustiva hemos visto que puede haber mucho interés en esta herramienta. Estas personas que estarían interesadas son entusiastas y profesionales del sector. No solo en la herramienta final tiene interés, sino que también en las distintas partes de esta.

1.6.1 Estudio de la competencia

Tras hacer un estudio de lo que hay en el mercado, con respecto a la herramienta de identificación de páginas web, hemos concluido que no hay ninguna aplicación comercial que haga lo que la nuestra pretende hacer. Aunque hay estudios teóricos de como identificar este tráfico cifrado[21] estos se basan en otras métricas, como la longitud de los paquetes, y no tienen aplicaciones prácticas. Por lo que consideramos que estamos en un campo muy innovador.

Por el contrario, la herramienta de experiencia de usuario sí que tiene mucha competencia. Para nuestro proyecto nos decantamos por usar la API que ofrece Google. Esta API se llama Lighthouse[16] y hace un estudio de la web que queremos investigar y nos devuelve un informe del que nosotros podemos extraer los datos que consideremos más relevantes. Esta API esta en sintonía con la intención de nuestro proyecto ya que, para hacer este estudio, utiliza el navegador más usado -como explicamos en la sección 4.3- y nos da datos concretos del rendimiento de esta web.

La competencia más significativa extraída de *raygun*[1]:

- *Raygun Real User Monitoring[2]:* Real User Monitoring captura cada sesión de usuario y destaca automáticamente las mayores mejoras que puede realizar en su web.
- *SOASTA[3]:* SOASTA usa Data Science Workbench, que ofrece datos de monitoreo de usuarios en tiempo real sobre experiencias de usuarios.
- *Pingdom[4]:* El software de Pingdom se especializa en la supervisión del rendimiento, incluida la supervisión de transacciones y tiempo de actividad.
- *New Relic[5]:* La herramienta de monitoreo de usuarios reales de New Relic se enfoca en el monitoreo del navegador y funciona para mejorar el rendimiento del lado del navegador.

El problema que encontramos en todas estas herramientas es que están pensadas para el desarrollador de la web y no para el estudio de esta. Este no es nuestro objetivo, por lo que vemos que la más parecida a nuestros intereses es le previamente mencionada herramienta de Google.

1.6.2 Herramienta de identificación

La herramienta de identificación puede que sea la más atractiva para el sector de análisis de red. Después de investigar, buscando algún producto que se cumpliera nuestras necesidades, concluimos que no había nada como el nuestro en el mercado. Hemos podido observar la gran cantidad de peticiones en foros para poder identificar webs en tráfico cifrado. Ninguna de la solución que se proponían en estos foros pasaba por analizar los third parties de las páginas web si no que proponían investigar los certificados de estas. También descubrimos que ninguna de las soluciones tenía tanto éxito como las nuestra. Por lo tanto, vimos que hay un gran mercado de entusiastas para nuestra herramienta. Por la parte de las grandes empresas que proporcionan servicios de conectividad se hace evidente el atractivo de la herramienta.

1.6.3 Herramienta de experiencia de usuario

La herramienta de experiencia de usuario puede ser la que menos atractivo tenga, pero aun así no la hemos infravalorado. Si es verdad que hay mucha competencia para el análisis de experiencia de usuario. El problema que hemos encontrado en todas es el tiempo de ejecución y la gran cantidad de datos innecesarios que generan. Nuestra herramienta apunta a lo necesario y a la velocidad así que puede ser muy atractiva para el análisis inmediato de webs resultando unos datos muy detallados y concretos que tienen que ver solamente con la experiencia que tiene el usuario.

1.6.4 Herramienta total

Finalmente tenemos la herramienta total que es una combinación de la dos mencionadas con anterioridad. Esta tiene un público más pequeño, pero con más interés. Las grandes empresas de telecomunicaciones se beneficiarían mucho con la combinación que nosotros ofrecemos. La capacidad de analizar lo contento que están todos y cada uno de los usuarios de la red es muy importante para estas empresas. A demás, nuestra herramienta, ofrece un estudio individual de cada usuario, esto significa que la empresa de telecomunicación, en caso de detectar algún problema, puede solucionarlo a diferentes niveles. Estos niveles pasan por solucionar el problema que tiene un usuario en concreto o un problema más grande que dependa de la propia teleoperadora.

1.6.5 Conclusión

Con las justificaciones anteriores vemos que hay muchos perfiles y muy distintos que podrían estar interesados en cada una de las partes de esta herramienta por lo que hemos concluido que es un proyecto con posibilidades de competir en un mercado que está en alza.

1.7 Alcance

En esta sección se hablará de los objetivos y requisitos principales que se tiene que conseguir el proyecto para proporcionar una solución valida.

1.7.1 Objetivos

Los siguientes objetivos son los que permitirán resolver el problema descrito con anterioridad.

- **Contestar la pregunta ¿se puede saber a qué página web se ha conectado un usuario sabiendo los third parties a los que se conecta la página web?**
    - Verificar la hipótesis principal del proyecto
- **Conseguir una herramienta que a partir de los third parties nos diga cuál es la página web con los que se conecta**
    - Crear un modelo suficientemente fiable que tenga un error inferior al 5%
    - Tomar suficientes muestras para tener una amplia base de datos de páginas web
- **Que el punto anterior sea lo suficientemente eficiente como para poder desplegarlo en una red real y que no sea un cuello de botella**
    - Una vez tengamos una herramienta funcional, optimizarla para que sea suficientemente rápida
- **Crear una nueva herramienta que nos diga lo contento o descontentos que están los usuarios visitando esas páginas web**
    - Creación de otra herramienta que nos diga la calidad de experiencia de los usuarios mirando métricas como el tiempo que tarda en cargarse esa web
- **Combinar las dos herramientas para que un administrador de red pueda averiguar la calidad de experiencia de sus usuarios de su red a partir de tráfico cifrado**
    - Desplegar esta herramienta en una red real para comprobar su eficiencia

1.7.2 Requisitos no funcionales

A parte de los objetivos principales del proyecto hay una serie de requisitos no funcionales de la herramienta que tiene que cumplir para garantizar el correcto funcionamiento:

- **Soporte técnico:** una vez alguna empresa o particular haya comprado nuestra herramienta se debe dar un mantenimiento adecuado
- **Disponibilidad:** la herramienta debe estar siempre disponible, que no haya caídas del sistema ya que se puede usar en cualquier momento
- **Escalabilidad:** la herramienta debe permitir monitorear tanto una red pequeña como grande
- **Seguridad y privacidad:** uno de los objetivos indispensables es que la privacidad de los usuarios se mantenga siempre

1.7.3 Obstáculos y riesgos

Existen posibles riesgos u obstáculos que pueden afectar al desarrollo de este proyecto:

- **Inexperiencia en las tecnologías usadas:** el proyecto se desarrollará en su mayoría en el lenguaje de programación *Python*. A esto se le tiene que sumar el hecho de la intención de usar algoritmos de machine learning, es posible que la inexperiencia sea un obstáculo añadido.
- **Fecha de entrega fijada:** el hecho de tener una fecha fijada hace que cualquier contratiempo en el desarrollo haga peligrar una parte o la totalidad del proyecto.

1.8 Metodología

El desarrollo del proyecto se hará usando la metodología A*gile*, en concreto la metodología *Scrum*. Esta es una metodología iterativa e incremental de manera que se definen una serie de iteraciones y en cada una se añade un valor al producto final. Esta divide el proyecto con diferentes roles que se adoptan en cada parte del proyecto por la misma persona.

Para facilitar el desarrollo con esta metodología se usará diversas herramientas.

1.8.1 Sistema de control de versiones

Para el control de versiones del código necesario para la herramienta se usará GitHub. Esta herramienta implementa un sistema de versiones llamado Git. Este implementa control de versiones del código que basado en repositorios y brancas

- *Git*[7]: es un sistema de control de versiones gratuito y de código abierto, se usa mucho en el desarrollo de software cooperativos ya que permite que diversos miembros del equipo trabajen en el mismo código.
- *GitHub[8]*: es un gestor de repositorios de Git que implementa una web para poder ver el repositorio en tiempo real.

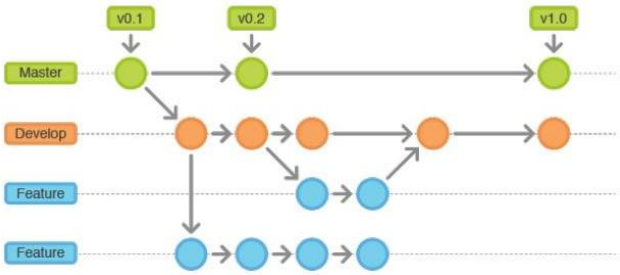

Figura 2: brancas de Git

1.8.2 Gestión de requisitos

Para controlar los requisitos impuestos para el desarrollo del proyecto se usará la herramienta trello. Esta herramienta es perfecta para nuestra metodología de trabajo ya que permite crear milestones, categorizarlos y saber en qué momento de desarrollo están.

1.8.3 Sistema de gestión de proyectos Agile

- *Taiga:* es un sistema de gestión de proyectos *Agile*. Este es de código abierto y gratuito lo que lo convierte en ideal para desarrollar este proyecto.

2 Estado del arte

2.1 Herramienta de identificación

La principal tecnología usada es el machine learning. Nosotros optamos por un modelo *Bag-Of-Words*[1] de machine learning para predecir las páginas webs que el usuario quiere encontrar.

En este campo se está haciendo muchos avances con algoritmos muy sofisticados. El machine learning o Inteligencia artificial esta, actualmente, en una gran variedad de aplicaciones de uso cotidiano. Como dice en el artículo [11] el machine learning está presente en tecnologías como Deep learning, Blockchain o internet de las cosas (IOT).

Toda esta investigación y avances nos hizo decantarnos por estas tecnologías. Al usar una api [12] para implementar nuestro machine learning nos aseguramos usar los últimos avances disponibles sobre la materia. Con esto nos aseguramos que nuestro proyecto está acorde con el estado del arte y que obtenemos el mejor rendimiento y resultados posibles.

2.2 Herramienta de experiencia de usuario

Para la herramienta de experiencia de usuario también optamos por usar una a api para poder obtener los mejores resultados. Esta api es de Google lo que nos supone una garantía de uso de las nuevas tecnologías disponibles.

Hemos tenido en cuenta los distintos parámetros del articulo [14] para poder comparar las diferentes alternativas. Finalmente nos decantamos por la de Google por ser la más completa y precisa.

2.3 Conclusión

Creemos que nuestro proyecto está en sintonía con el estado del arte ya que, en primer lugar, el machine learning de la herramienta de identificación, que usa las tecnologías de inteligencia artificial para poder predecir la web que estamos buscando. En segundo lugar, la API de Google que hace un exhaustivo estudio del rendimiento de dicha web teniendo en cuenta los parámetros más relevantes a dia de hoy. Finalmente creemos que, la combinación de estas dos herramientas está pensada para medir el comportamiento de los servicios web actualmente.

[1] El modelo de bolsa de palabras es una representación simplificadora utilizada en el procesamiento del lenguaje natural y la recuperación de información. En este modelo, un texto (como una oración o un documento) se representa como la bolsa de sus palabras, sin tener en cuenta la gramática e incluso el orden de las palabras, pero manteniendo la multiplicidad [20]

3. Planificación del proyecto

La duración aproximada del proyecto es de 4 meses y medio, empezando el día 21 de septiembre con la gestión de proyectos y acabando a mitades de diciembre (aproximadamente el día 18 de diciembre), ya que la lectura se hace la última semana de enero.

Del día 21 de septiembre al día 19 de octubre se imparte la asignatura de gestión de proyectos (GEP), analizando el contexto, definiendo el abasto y la planificación temporal, haciendo el análisis de riesgos y costes, y haciendo el informe sobre sostenibilidad. Siguiendo los requerimientos de GEP, esta franja constara de 80h de trabajo.

Del día 21 de septiembre hasta la fecha de entrega del proyecto (una semana antes de la lectura) se llevará a cabo la parte técnica del trabajo. Esta se llevará en paralelo a la asignatura de GEP (mientras esta dure). Esto nos deja un total de 390h de trabajo.

La memoria se cerrará los últimos días del proyecto, contando unas 30h (de las 390 totales) para acabarla.

El total del proyecto constara de 470h de trabajo.

3.1 Estimación temporal del proyecto

El proyecto se divide en cinco partes. La etapa inicial es la gestión del proyecto, las siguientes tres etapas serán la parte técnica y la final será la finalización la documentación de la memoria.

Las etapas técnicas constaran del desarrollo y de la documentación correspondiente a cada etapa.

Finalmente habrá una última etapa en la que se finalizará la documentación de la memoria. Se puede ver toda la información en la tabla siguiente, en ella se muestra la fecha de inicio y de finalización de cada etapa.

Etapa	**Fecha inicio**	**Fecha fin**
Gestión de Proyectos	21-9-2020	19-10-2020
Herramienta de identificación	21-9-2020	5-12-2020
Herramienta de experiencia de usuarios	14-10-2020	13-11-2020
Optimización total	10-12-2020	15-12-2020
Documentación final	15-12-2020	Entrega del proyecto

Tabla 1: Fechas planificadas para cada etapa

3.1.1 Definición de tareas

3.1.1.1 Gestión de proyectos

- **GP1 - Contextualización i abasto:** Definir el contexto y el alcance del proyecto

  Duración: 20h

  Dependencias: -

  Recursos humanos: Jefe de proyecto

  Recursos materiales: Ordenador con Internet

- **GP2 – Planificación temporal:** Realizar la definición de las tareas a hacer y su duración

  Duración: 20h

  Dependencias: GP1

  Recursos humanos: Jefe de proyecto

  Recursos materiales: Ordenador con Internet, Ganttproject

- **GP3 – Gestión Económica i sostenibilidad:** Realizar un plan económico y un informe de sostenibilidad

  Duración: 20h

  Dependencias: GP2

  Recursos humanos: Jefe de proyecto

  Recursos materiales: Ordenador con Internet

- **GP4 – Definición del Proyecto:** Agrupar los anteriores documentos, cambiando lo pertinente, para realizar un informe final

  Duración: 20h

  Dependencias: GP3

  Recursos humanos: Jefe de proyecto

  Recursos materiales: Ordenador con Internet

3.1.1.2 Herramienta de Identificación

- **HI1 – Estudio de data sets:** A partir de los data sets dados hacer un estudio de como trabajar con ellos para poder definir la dirección del proyecto

  Duración: 20h

  Dependencias: -

  Recursos humanos: Analista

  Recursos materiales: Ordenador con Internet, GIT

- **HI2 – Creación de un dataset único:** A partir de los data sets dados hacer un único dataset que combine el resto y haga que el tiempo de computación pueda ser más bajo

  Duración: 20h

  Dependencias: HI1

  Recursos humanos: Programador

  Recursos materiales: Ordenador con Internet, GIT, Visual Studio Code[2], Trello

- **HI3 – Búsqueda:** Programa que busque de que página web se trata. Este programa partirá del dataset único que se ha creado cona anterioridad.

  Duración: 70h

  Dependencias: HI2

  Recursos humanos: Programador

  Recursos materiales: Ordenador con Internet, GIT, Visual Studio Code, Trello

- **HI4 – Comprobación de premisa:** Comprobar si la premisa de nuestro trabajo se cumple haciendo tests. El error tendrá que ser menor al 20% para confirmar que la premisa se cumple.

  Duración: 25h

  Dependencias: HI3

  Recursos humanos: Programador

  Recursos materiales: Ordenador con Internet, GIT, Visual Studio Code, Trello

- **DOC1 – Documentación de *herramienta de identificación*:** Documentar lo hecho con anterioridad

  Duración: 20h

  Dependencias: HI4

  Recursos humanos: Jefe de proyecto

  Recursos materiales: Ordenador con Internet

[2] Editor de código: code.visualstudio.com

3.1.1.3 Herramienta de experiencia de usuarios

- **HEU1 – Investigación:** Investigar cómo se puede saber la experiencia de un usuario en una página web

  Duración: 20h

  Dependencias: -

  Recursos humanos: Analista

  Recursos materiales: Ordenador con Internet

- **HEU2 – Creación de herramienta:** A partir de la información obtenida crear la herramienta

  Duración: 45h

  Dependencias: HEU1

  Recursos humanos: Programador

  Recursos materiales: Ordenador con Internet, GIT, Visual Studio Code, Trello

- **HEU3 – Incorporación de las dos herramientas:** Creadas las dos herramientas combinar ambas para que una encuentre de que web se trata y la otra exprese cual es la experiencia del usuario

  Duración: 50h

  Dependencias: HEU3, HI4

  Recursos humanos: Programador

  Recursos materiales: Ordenador con Internet, GIT, Visual Studio Code, Trello

- **DOC2 – Documentación de *Herramienta de experiencia de usuarios*:** Documentar lo hecho con anterioridad

  Duración: 20h

  Dependencias: HEU3

  Recursos humanos: Jefe de proyecto

  Recursos materiales: Ordenador con Internet

3.1.1.4 Optimización total

- **OT1 – Optimización de la herramienta final:** Optimizar la herramienta para que pueda ser lo bastante rápida para ser desplegada en una red real

  Duración: 50h

  Dependencias: HEU3

  Recursos humanos: Programador

  Recursos materiales: Ordenador con Internet, GIT, Visual Studio Code, Trello

- **DOC3 – Documentación de *Herramienta de experiencia de usuarios*:** Documentar lo hecho con anterioridad

  Duración: 20h

  Dependencias: OT1

  Recursos humanos: Jefe de proyecto

  Recursos materiales: Ordenador con Internet

3.1.1.5 Documentación final

- **DOC4 – Documentación Final:** Completar la documentación con lo que falte

  Duración: 30h

  Dependencias: -

  Recursos humanos: Jefe de proyecto

  Recursos materiales: Ordenador con Internet

Etapa	Horas	Rol	Tipo de rol	Total (h)
GP1	20	Jefe de proyecto	Jefe de proyecto	170
GP2	20	Jefe de proyecto	Analista	65
GP3	20	Jefe de proyecto	Programador	235
GP4	20	Jefe de proyecto		
HI1	20	Analista		
HI2	20	Programador		
HI3	70	Programador		
HI4	25	Analista		
DOC1	20	Jefe de proyecto		
HEU1	20	Analista		
HEU2	45	Programador		
HEU3	50	Programador		
DOC2	20	Jefe de proyecto		
OT1	50	Programador		
DOC3	20	Jefe de proyecto		
DOC4	30	Jefe de proyecto		
Total	**470h**			

Tabla 2: Resumen etapas y horas por rol

3.1.2 Diagrama de Gantt

En la siguiente figura podemos ver como quedan las tareas repartidas en el tiempo mediante un diagrama de Gantt. Cada color diferencia las diferentes etapas (un color por cada etapa y un color para la documentación).

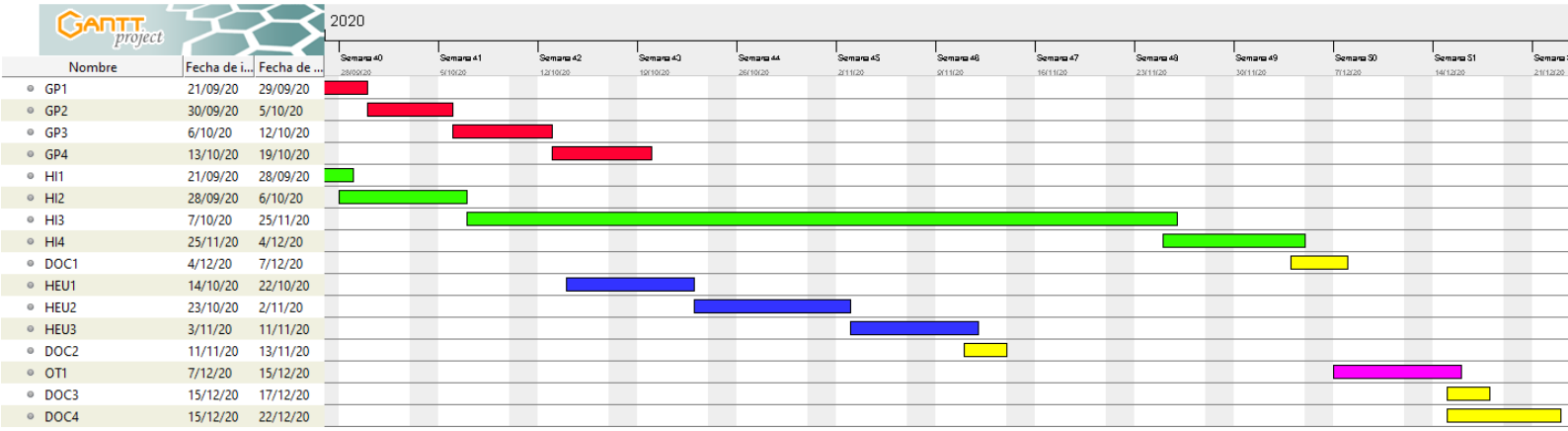

Figura 3: Diagrama de Gantt

3.2. Gestión de riesgos y planes alternativos

En la siguiente tabla podemos ver los riesgos y obstáculos que nos podremos encontrar en nuestro trabajo. Por cada riesgo u obstáculo se indica el impacto que tendrían y la probabilidad de que pasen. Finalmente se indica que se haría en el caso de que pasara.

Riesgo	**Impacto**	**Probabilidad**	**Plan alternativo**
Caída de GitHub	Medio	Baja	Tenemos una copia en local para poder seguir trabajando
Caída de Trello	Medio-alto	Baja	Tener apuntado los milestones para poder saber culés son los siguientes pasos
Planificación temporal errónea	Bajo	Medio	El plan temporal es dinámico, si alguna tarea dura más de lo esperado se quitara tiempo de otras que no necesiten tanto

Tabla 3: Riesgos y planes alternativos

3.3 Presupuesto

3.3.1 Identificación y estimación de los costes

Para el correcto desarrollo del proyecto son necesarios una serie de recursos humanos y materiales.

Los recursos materiales son el hardware y software con los que se desarrollara todo el proyecto, y los materiales humanos son los costes de los miembros del equipo. También hay recursos indirectos, como el coste de las instalaciones donde se llevará a cabo el desarrollo.

3.3.1.1 Recursos humanos

Para calcular el sueldo por hora para los recursos humanos se ha investigado el sueldo promedio de todos los roles adoptados en el proyecto en Barcelona[3].Como un año tiene 12 meses y por mes se trabaja unos 21 días por mes y se trabaja una jornada completa de 8h al día, esto nos da un precio por hora de 10,7 €/h (Programador junior), 19,6 €/h (jefe), 19,4 (Analista).

Los salarios expuestos ya cuentan con impuestos, estos son lo que pagara la empresa contratadora.

Cada etapa hace referencia al diagrama de GANTT que se ha mostrado con anterioridad.

Etapa	**Horas**	**Rol**	**Precio por hora**	**Total (€)**
GP1	20	Jefe de proyecto	19,6	392
GP2	20	Jefe de proyecto	19,6	392
GP3	20	Jefe de proyecto	19,6	392
GP4	20	Jefe de proyecto	19,6	392
HI1	20	Analista	19,4	388
HI2	20	Programador	10,7	214
HI3	70	Programador	10,7	749
HI4	25	Analista	19,4	485
DOC1	20	Jefe de proyecto	19,6	392
HEU1	20	Analista	19,4	388
HEU2	45	Programador	10,7	481,5
HEU3	50	Programador	10,7	535
DOC2	20	Jefe de proyecto	19,6	392
OT1	50	Programador	10,7	535
DOC3	20	Jefe de proyecto	19,6	392
DOC4	30	Jefe de proyecto	19,6	588
Total	**470h**			7.107,5€

Tabla 4: Costes humanos en horas y precio

[3] Salario medio Barcelona: https://es.indeed.com/salaries

3.3.1.2 Recursos materiales

Se usará un ordenador de torre montado por piezas (16gb RAM, AMD Ryzen 5 2600, Nvidia Gtx 1660, 2Tb) durante los 4 meses de desarrollo y documentación del proyecto. Los programas usados son públicos, por lo tanto, son gratuitos y se incluirán en los costes materiales.

Estos elementos tienen una vida limitada, todo este hardware se estropea con el paso del tiempo. En este caso se calculará su amortización a partir de la siguiente formula:

Coste del equipo / Vida útil

Equipo	**Coste**	**Vida útil**	**Amortización**
Ordenador Torre	800€	4 años	200€

Tabla 5: Recursos materiales

3.3.1.3 Recursos indirectos

En la tabla que está a continuación se detalla el precio de los costes indirectos generados para la realización del proyecto. Estos constan de la conexión de internet [9] necesaria para recoger los dataset y para poder trabajar durante todo el proyecto, y de la electricidad que se necesita para conectar los equipos necesarios [10] (contando que mi ordenador consume unos 2,2kWh cada 10h).

Producto	**Precio**	**Consumo**	**Tiempo**	**Total**
100 Mb fibra	28,95€/mes	-	4 meses	115,8€
Electricidad	0,09884 €/kWh	2,2kWh/10h	390 h	8,48€

Tabla 6: Recursos indirectos

3.3.1.4 Contingencias

Para las contingencias, se establecerá un porcentaje del 15%. En la tabla siguiente se especifica el precio resultando de aplicar la contingencia a los precios previamente mencionados.

Tipo de coste	**Precio**	**Precio final**
Recursos humanos	7.107,5€	8173,62€
Recursos materiales	800€	920€
Recursos indirectos	124,28€	142,92€

Tabla 7: coste de contingencias

3.3.1.5 Imprevistos

Los imprevistos vistos en la planificación temporal no afectan a los costes del proyecto. Por lo tanto, no suponen más planificación de los costes. Los únicos imprevistos que podrían añadir costes al presupuesto son los fallos con el equipo con el que trabaja. Estos se especifican en la tabla siguiente.

Elemento	**Reparación**
Ordenador torre	50€

Tabla 8: Coste de imprevistos

3.3.1.6 Presupuesto final

Haciendo un balance final, en la siguiente tabla está el presupuesto final del coste del desarrollo del proyecto.

Tipo de coste	**Precio**
Recursos humanos	8173,62€
Recursos materiales	920€
Recursos indirectos	142,92€
Total	**9.236,54€**

Tabla 9: presupuesto final

3.3.2 Control de desviación

Para controlar las posibles desviaciones del presupuesto usaremos las fórmulas siguientes para recalcular el nuevo presupuesto. Esto se hará una vez finalizada cada etapa y usaremos las horas reales y no las presupuestadas.

- **Desviación de horas por cada etapa**
  - *(H. presupuestadas – H. reales) * Coste estimado*
- **Desviación de coste**:
  - *(H. presupuestadas – H. reales) * Coste real*
- **Desviación de coste en recursos humanos**:
  - *(Coste estimado – Coste real) * H. reales*
- **Desviación de coste de materiales:**
  - *Coste material estimado – Coste material real*
- **Deviación de costes indirectos**:
  - *Coste indirecto estimado – Coste indirecto real*
- **Desviación de costes imprevistos**:
  - *Costes imprevistos estimado – Coste imprevistos real*
- **Desviación de costes personal**:
  - *Coste personal estimado – Coste personal real*
- **Desviación de horas totales**:
  - *Horas estimadas – Horas reales*
- **Desviación total de costes**:
  - *Costes estimados – Costes reales*

4 Tecnologías

A continuación, explicaremos las tecnologías usadas para realizar el proyecto. Estas son el Machine learning para la herramienta de identificación y Lighthouse para la de experiencia de usuario.

4.1 Word2Vec

Nuestro Machine Learning se basa en un modelo llamado *Word-To-Vec* o *Word2vec* [13]. *Word2vec* es una técnica para el procesamiento del lenguaje natural . El algoritmo utiliza un modelo de red neuronal para aprender asociaciones de palabras de una gran cantidad de texto . Una vez entrenado, el modelo puede detectar palabras sinónimas o sugerir palabras adicionales para una oración parcial. Como su nombre lo indica, *word2vec* representa cada palabra distinta con una lista particular de elementos, vectores . Los vectores se eligen de modo que una función matemática simple (la similitud del coseno entre los vectores) indique la similitud semántica entre las palabras representadas por esos vectores.

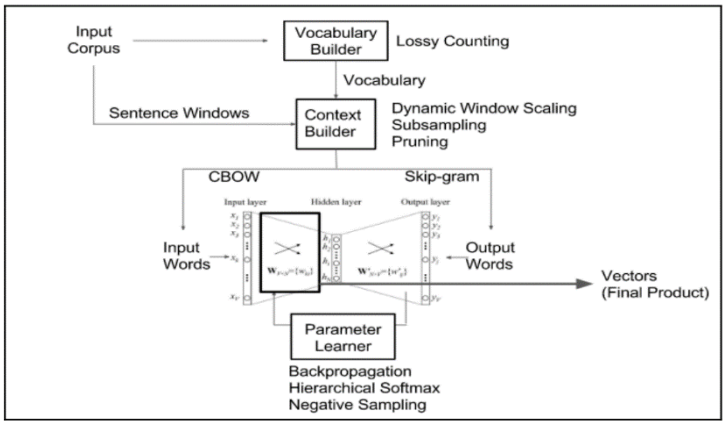

Figura 4: Arquitectura Word2Vec [18]

Para poder identificar cual será el servicio que estamos buscando, inferiremos un nuevo vector. Una vez lo tenemos, buscaremos en el espacio n-dimensional el vector que este más cercano. Para determinar cuál es el más cercano usaremos la distancia euclidiana. Una vez hayamos seleccionado este vector miraremos su tag (nombre del servicio web) y consideraremos que ese es el servicio que estamos buscando

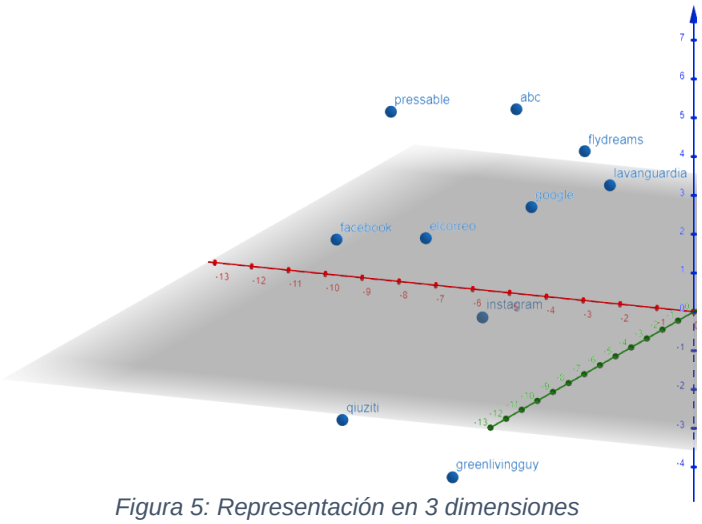

Figura 5: Representación en 3 dimensiones

Tras ver que esta era la solución que más se adecuaba a nuestro proyecto, descubrimos que no daba buenos resultados. Este método está pensado para predecir una palabra dando un input con otras.

Ej.

Word2Vec:

Input: [‘el’, ‘niño’, ‘es’] → Output → ‘alto’

Lo que necesitamos:

Input [‘niño’, ‘chico’, ‘hombre’] →Output→ ‘masculino

Lo que nosotros necesitábamos es predecir un “documento” a partir del contenido de este por lo que optamos por la variante de este algoritmo *Doc-To-Vec*.

4.2 Doc2Vec

Este algoritmo parte del anterior (*Word2Vec*). Dado un conjunto de palabras (documento) las parametriza en vectores. Con todos estos vectores crea uno único y le asigna un tag (nombre del documento). Esto lo guarda en el modelo. Este modelo nos permite seleccionar el tamaño del vector, esto determinara las dimensiones de estos.

Para nosotros el conjunto de palabras serán la lista de third parties de cada web y el tag será el nombre de esta web. Gracias a esto podremos usar este algoritmo de machine learning para nuestro propósito.

Para poder hacer una búsqueda se infiere un nuevo documento. Esto hace lo mismo que en el entreno del modelo. Crea un vector a partir de los inputs introducidos. Una vez tenemos ese vector podemos buscar el más próximo a este, esto nos dará cual es el que buscamos.

Esta solución satisfacía perfectamente las necesidades de nuestro proyecto. Con este método la búsqueda es casi instantánea por lo que es perfecto.

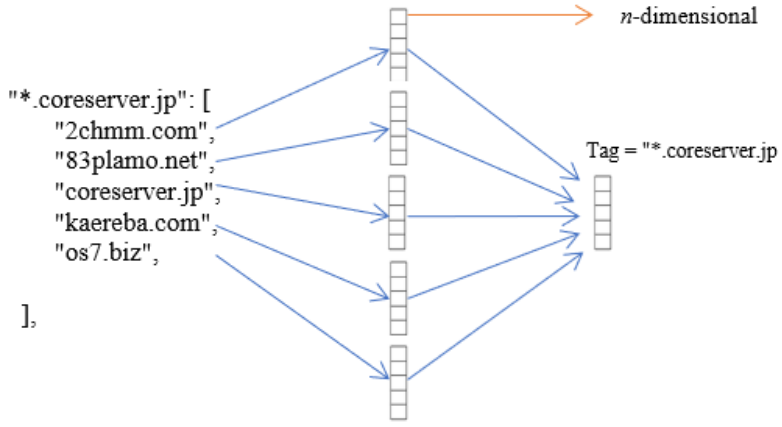

Figura 6: Arquitectura Doc2Vec [12]

4.3 Lighthouse

Lighthouse [16] es una herramienta automatizada open-source, que busca auditar la calidad de las páginas web. Se puede correr contra cualquier página web, sea pública o requiera autenticación. Cuenta con auditorías de rendimiento, accesibilidad, SEO y más.

4.3.1 Parámetros principales

Esta API hace un exhaustivo estudio de la página web a la que se conecta. Tras este estudio nos retorna una serie de parámetros con sus resultados individuales. Los principales son los siguientes:

4.3.1.1 First Contentful Paint:

FCP (First Contentful Paint) mide cuánto tiempo necesita el navegador para procesar la primera parte de la página web después de que un usuario entre a su página. Las imágenes, los <canvas> elementos que no son blancos y los SVG de la página se consideran contenido del dominio; no se incluye nada dentro de un iframe .

Puntuación

La puntuación FCP es una comparación del tiempo de la página a buscar y los tiempos para sitios web reales. Por ejemplo, el 99% de las webs cargan este contenido en 1.5s. Si el FCP del sitio web es de 1,5 segundos, su puntuación es 99.

Esta tabla muestra cómo interpretar la puntuación FCP:

Tiempo (seg)	Color	Puntuación (porcentaje)
0-2	Verde	100-75
2-4	Naranja	74-50
+4	Rojo	49-0

Tabla 10: Puntuación First Contentful Paint

4.3.1.2 Largest Contentful Paint:

LCP mide el tiempo que tarda en presentarse en la pantalla el elemento de contenido más grande de la ventana. Esto se aproxima a cuando el contenido principal de la página es visible para los usuarios.

Puntuación

La puntuación del Largest Contentful Paint sigue la siguiente tabla:

Tiempo (seg)	Color	Puntuación (porcentaje)
0-1.2	Verde	100-75
1.3-1.41	Naranja	74-50
+1.41	Rojo	49-0

Tabla 11: Puntuación First Meaningful Paint

4.3.1.3 Speed Index:

El índice de velocidad mide lo rápido se muestra visualmente el contenido durante la carga de la página. Lighthouse primero captura un video de la carga de la página en el navegador y calcula la progresión visual entre fotogramas. Luego, Lighthouse usa el módulo Speedline Node.js para generar la puntuación del índice de velocidad.

Puntuación

La Puntuación de índice de velocidad es una comparación de este de la página a buscar y los índices de velocidad de sitios web reales.

Tiempo (seg)	Color	Puntuación (porcentaje)
0–4,3	Verde	100-75
4.4–5.8	Naranja	74-50
+5.8	Rojo	49-0

Tabla 12: Puntuación Speed index

4.3.1.4 Time to Interactive:

TTI mide el tiempo que tarda una página en volverse completamente interactiva. Una página se considera completamente interactiva cuando:

- La página muestra contenido útil, que se mide con la primera pintura con contenido ,
- Los controladores de eventos están registrados para la mayoría de los elementos visibles de la página, y
- La página responde a las interacciones del usuario en 50 milisegundos.

Puntuación

La puntuación TTI es una comparación del TTI de la página y el TTI de sitios web reales, según los datos del HTTP . Por ejemplo, los sitios que se desempeñan en el percentil 99 representan TTI en aproximadamente 2.2 segundos. Si el TTI de su sitio web es de 2,2 segundos, su puntuación TTI es 99.

Tiempo (seg)	Color
0-3,8	Verde
3,9–7,3	Naranja
+7.3	Rojo

Tabla 13: Puntuación Time to Interactive

4.3.1.5 Total Blocking Time:

TBT mide la cantidad total de tiempo que una página está bloqueada y que no responde a la entrada del usuario, como los clics del mouse, los toques de pantalla o las pulsaciones del teclado. El resultado se calcula sumando la parte de bloqueo de todas las tareas largas entre First Contentful Paint y Time to Interactive . Cualquier tarea que se ejecute durante más de 50 ms es una tarea larga. La cantidad de tiempo después de 50 ms es la parte de bloqueo. Por ejemplo, si Lighthouse detecta una tarea de 70 ms de duración, la porción de bloqueo sería de 20 ms.

Puntuación

La puntuación TBT es una comparación del tiempo TBT de su página y los tiempos TBT para los 10,000 sitios principales cuando se carga.

Tiempo (milisegundos)	Color
0–300	Verde
300-600	Naranja
+600	Rojo

Tabla 14: Puntuación Total Blocking Time

4.3.1.6 Cumulative Layout Shift:

CLS mide la suma total de todas las puntuaciones de cambio de diseño individuales para cada cambio de diseño inesperado que se produce durante toda la vida útil de la página.

Se produce un cambio de diseño cada vez que un elemento visible cambia su posición de un fotograma renderizado al siguiente.

Puntuación

La puntuación de este parámetro se calcula siguiendo la siguiente formula:

*layout shift score = impact fraction * distance fraction*

Tiempo (milisegundos)	Color
0–0.10	Verde
0.10-0.25	Naranja
+0.25	Rojo

Tabla 15: Puntuación Cumulative Layout Shift

4.3.1.7 Cálculo final:

Para calcular la puntuación final la api le da los siguientes pesos a cada uno de los parámetros [17]:

Parámetro	Peso
First Contentful Paint	15%
Speed Index	15%
Largest Contentful Paint	25%
Time to Interactive	15%
Total Blocking Time	25%
Cumulative Layout Shift	5%

Tabla 16: Peso de los parámetros

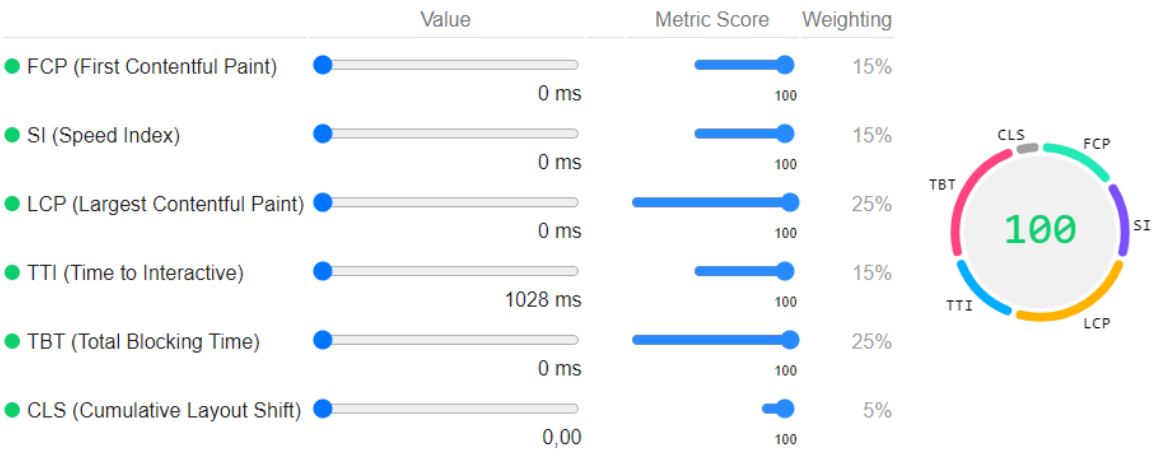

Figura 7: Cálculo de Experiencia de usuario [17]

5 Proyecto

5.1 Herramienta de identificación

Este es el primer apartado que se empezó a desarrollar y es la columna vertebral del proyecto. Se trata de una herramienta escrita en Python que utiliza los third parties a los que se conectan las diferentes webs para poder identificarlas en un entorno encriptado.

5.1.1 Solución original

La solución simple, consistía en extraer estos third parties de más de 50.000 webs diferentes, hacerlo durante el tiempo y configurar un dataset único que fuese la combinación de estos. Seguidamente se usaría un algoritmo muy básico de búsqueda basado en las coincidencias para determinar, a partir de un input de los third parties, cual es la página web que estamos buscando.

Algoritmo

El algoritmo usado consistía en, una vez tuviéramos todos los diferentes third parties de una web, calcular el porcentaje en el que cada uno de estos aparecía en las diferentes instancias de los datasets. A esto se le añadiría una importancia superior a las apariciones más recientes por lo que una web que tiene las mismas apariciones que otra, tendrá un número más alto la que más recientemente haya aparecido Ej.:

Dado los third parties de una sola página web, crearemos una puntuación para cada uno.

```
Plus = 0;
For (data in datasets){
        For (third_partie in data){
                Frec_dataset_total[third_partie] += (1 + plus)
        }
        Plus += 0.1
}
```

Porcentaje final = (Frec_dataset_total / total * 100)
Fecha de Dataset1< Dataset2< Dataset3< Dataset4

Dataset 1	Dataset 2 (+0.1)	Dataset 3 (+0.2)	Dataset 4 (+0.3)	Porcentaje	Puntuacion
Google.com	-	-	Google.com	50	57.5
Facebook.com	Facebook.com	Facebook.com	-	75	82.5
Youtube.com	-	Youtube.com	-	50	55
Fulstory.com	Fulstory.com	Fulstory.com	Fulstory.com	100	115
Nr-data.es	-	-	-	25	25

Tabla 17 Ejemplo Algoritmo Básico

Una vez creado el dataset con el total de las webs siguiendo el modelo anterior se procedía a hacer la búsqueda.

Esta búsqueda consistía en coger los inputs y calcular cual sería los mayos suma que se podía generar y a que web principal pertenecía.

Este método simple solo tenía en cuenta ocurrencia de los third parties y la temporalidad de ellos. Después de esta implementación nos dimos cuenta que, aunque los resultados eran buenos, esta herramienta era demasiado lenta y sencilla para poder desplegarla en una red real, como veremos a continuación en los resultados.

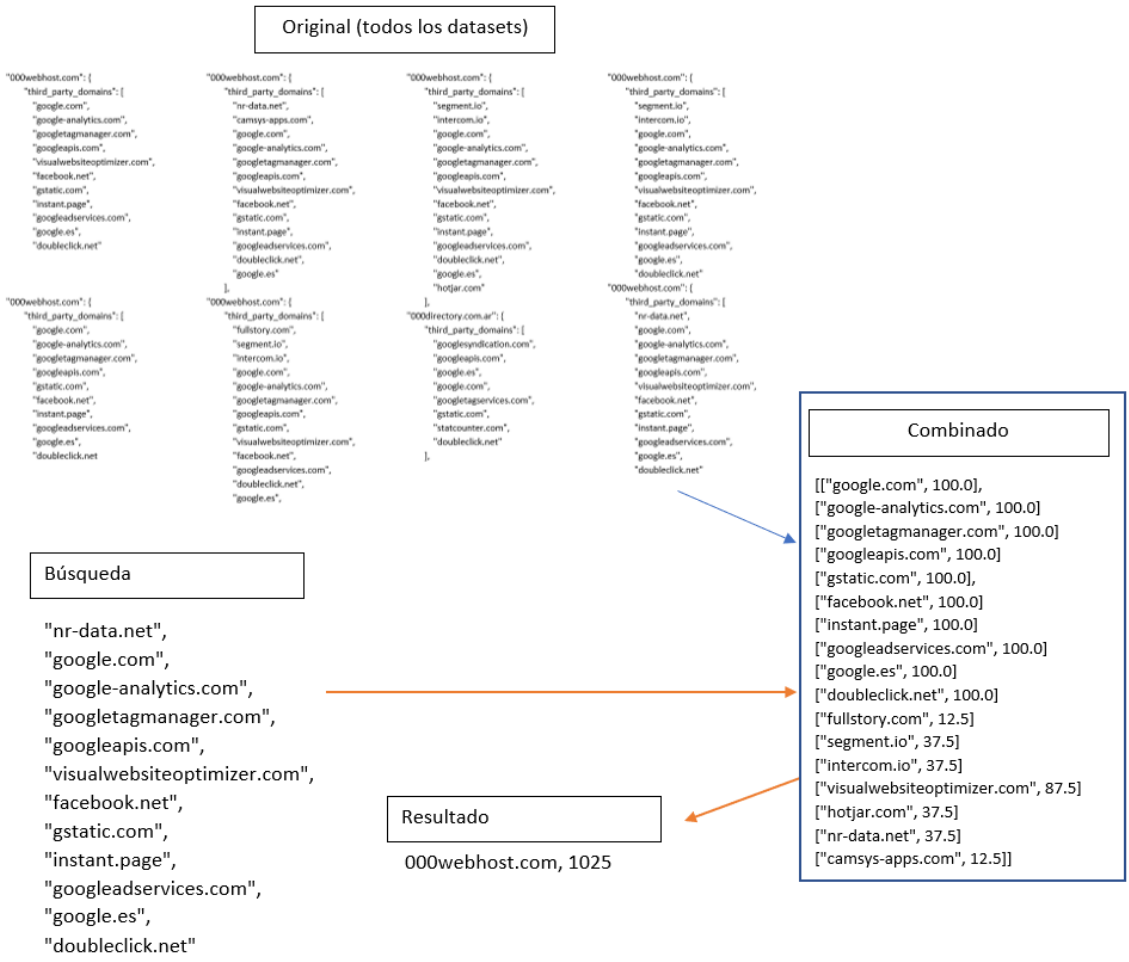

Figura 8: Ejemplo Real Algoritmo Básico

Descripción del dataset

Dataset	Numero de webs	Fecha
Dataset 1	25781	13/03/2020
Dataset 2	21620	20/03/2020
Dataset 3	21430	23/03/2020
Dataset 4	42929	25/03/2020
Dataset 5	21722	27/03/2020
Dataset 6	21937	29/03/2020
Dataset 7	25713	04/06/2020
Dataset 8	25404	15/10/2020
Total	46129	-

Tabla 17 Descripción de datasets

Resultados

Los resultados que obtuvimos fueron los siguientes:

Acierto: 60%

Tiempo medio de búsqueda: 2s

El acierto se refiere a las páginas web que averiguo correctamente, si la conexión es con Facebook.com y el resultado del algoritmos es Facebook.com.

Aunque el tiempo medio era relativamente bajo se vio como dependía directamente del tamaño de la entrada, era lineal. Para poder desplegarlo en una red real era necesario bajar el tiempo de ejecución.

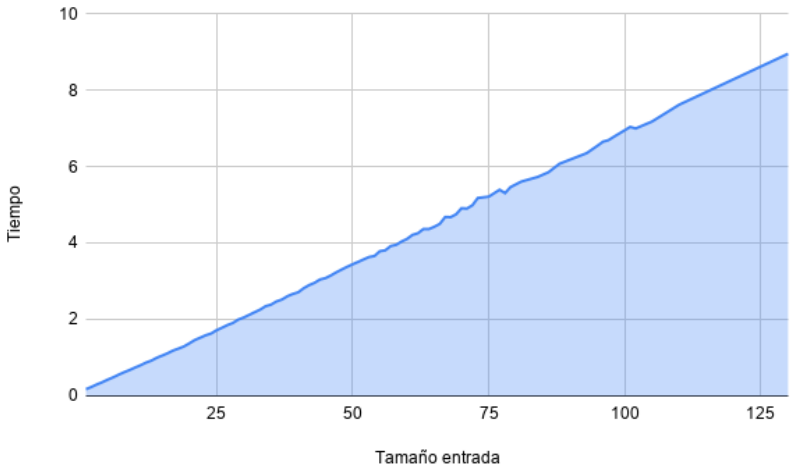

Figura 9: Grafico Rendimiento Algoritmo Básico

5.1.2 Solución final

Finalmente, se decidió apostar por una herramienta basada en machine learning. Aprovechamos este dataset único para entrenar a un machine learning que se basa en el Doc2Vec (como hemos explicado anteriormente). Este modelo convierte un documento (nuestros diferentes third parties por cada web) en un vector de *n* dimensiones y le asigna un Tag (en este caso el nombre de la web).

Para poder buscar un input en nuestro modelo este se inferirá con el modelo, lo que nos dará un vector del mismo tamaño y dimensiones que el resto. Finalmente se buscará el vector que esté más cerca de este.

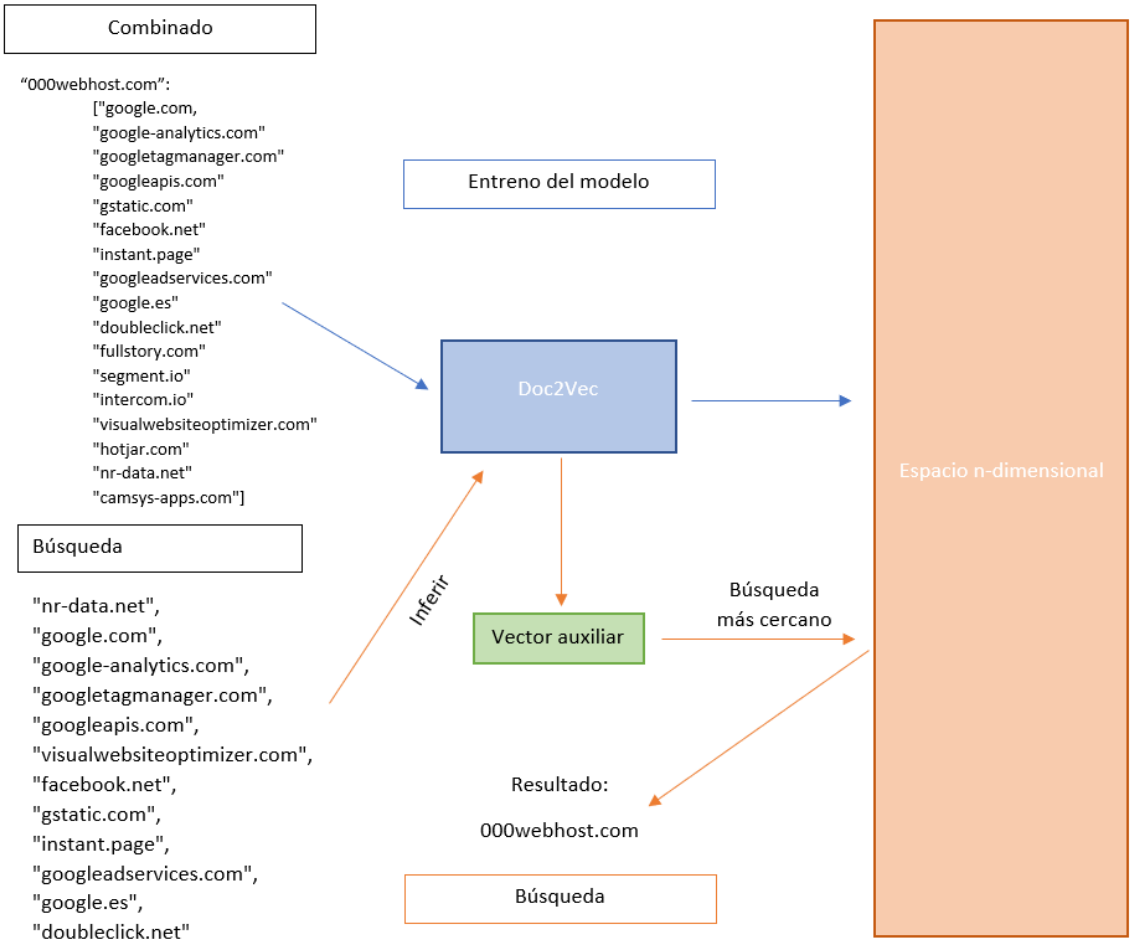

Figura 10: Ejemplo de funcionamiento Doc2vec

Resultados

Los resultados que obtuvimos con los siguientes parámetros fueron los siguientes:

Parámetros:

Max_epochs (Iteraciones) = 280

Vec_size (Tamaño del vector) =200

Alpha = 0.065

Min_count = 0

Window = 2

Resultados**:**

Acierto: 75%

Tiempo medio: 0.082s

Estos resultados, con las mismas pruebas y los mismos datasets que en la solución inicial, demostraron que esta opción era la mejor ya que aumentaba la tasa de acierto y disminuía el tiempo a uno muy usable en una red real.

Cambiando el parámetro *Vec_size* cambiaríamos el tiempo, pero también cambiaríamos el acierto por eso convenimos en que los resultados obtenidos eran muy acertados.

5.1.3 Implicaciones del cifrado de los third parties

Hacia la finalización del proyecto nos encontramos con un problema con la seguridad de los third parties. Después de tener listas las dos herramientas, hicimos un estudio del tipo de conexión que tienen las webs con sus third parties. Este estudio concluyo que el 96.4 % de estos tienen una conexión https y por lo tanto está cifrada.

Este contratiempo nos obligó a buscar en los certificados de estas conexiones, más específicamente en el campo SNI para ver si este puede ser identificativo para el third party y así poder seguir con la misma hipótesis del proyecto.

El SNI o identificador del nombre del servidor, es un parámetro que envían los servidores en las conexiones https cuando responden con la página web. Este parámetro, como su nombre indica, sirve para identificar el nombre del servidor. El inconveniente que tiene es que no tiene por qué coincidir con el nombre del servicio.

Con esta solución nos dimos cuenta que si el SNI de los third parties sirven para usarlos en nuestro algoritmo porque no iba a servir este campo para identificar la página que estamos buscando. Después de un estudio observamos que el 60% de los SNI no son identificativos para las páginas web, mientras que para los third parties sí que son útiles. La razón por la que en los third parties sí que nos sirve este parámetro es porque no necesitamos que el nombre de este coincida con el SNI mientras que para identificar la web sí.

Los criterios que consideramos para determinar si un SNI es identificativo de una página web son los siguientes: si un SNI pertenece a más de una página web consideramos que no puede ser identificativo y si un SNI no contiene el nombre de la web a buscar también consideramos que no es identificativo. Por lo tanto, para que sí lo sea, necesita ser único y contener el nombre. Ej.:

Web: **www.Facebook.com**; Sni: **Facebook**; Apariciones: **1** -> Identificativo

Web: **www.Google.com**; Sni: **Google**; Apariciones: **10** -> No Identificativo

Web: **www.Instagram.com**; Sni: **Akamai**; Apariciones: **1** -> No Identificativo

Aunque los SNI no los consideremos identificativos los usaremos para acotar la búsqueda ya que gracias a estos tendremos un subconjunto menor en el que buscar y esto aumentara la precisión.

Organizaremos las diferentes webs que tenemos por su SNI. En caso que un SNI tenga más de una página web buscaremos en este subconjunto la web que estamos buscando.

Ej.

```
"*.coreserver.jp": [//SNI
   "2chmm.com",
   "83plamo.net",
   "coreserver.jp",
   "kaereba.com",
   "os7.biz",
   "269g.net",
   "car-license.co.jp",
   "dptheme.net",
   "jdb-tantei.com",
   "numbers34.jp",
   "puzzle-ch.com"
 ],
```

```
"www.baishancloud.com": [//SNI
   "81.cn",
   "china.org.cn",
   "it168.com",
   "kongzhong.com",
   "tibet.cn",
   "china.com.cn",
   "cnkang.com",
   "ce.cn",
   "sousuo.gov.cn"
 ],
```

5.1.4 Uso del SNI como identificador de los third parties

Tras los obstáculos descritos en el apartado anterior y el resultado del estudio se decidió que en vez de usar el nombre del third party (ej. nr-data.com) se usaría la información del SNI de este third party.

Después de cambiar todo el dataset con el que alimentamos el algoritmo de machine learning y volver a ejecutar la herramienta, resultó que pasamos del **75%** de acierto al **60%.** Esto es debido a la variedad de third parties comparado a la variedad de los SNI de estos.

Muchos de estos third parties comparten propietario, por ejemplo, Google tiene más de 500 third parties distintos y por lo tanto suelen tener el mismo SNI. Esto pasa con una gran mayoría de servicios. Esto hace que si una web tenia, por ejemplo, 15 third parties distintos, al pasar a los SNI de estos, acabe con unos 10 distintos. Esto hace que las diferentes webs se parezcan más, lo que complica la identificación de estas.

5.1.5 Solución final

Finalmente, tras tener en consideración la problemática del SNI y los estudios realizados nos decantamos para una solución combinada entre la búsqueda por este parámetro y los third parties.

Primero hacemos una criba por el SNI de la página a buscar, si este identifica la página web, esta será el resultado. En caso de que este parámetro no identificara la página web nos quedaríamos con el subconjunto de webs en las que puede estar, después haríamos la búsqueda por los third parties.

Esta solución nos da una precisión final para nuestra herramienta de identificación del 88%. Si solo hubiéramos usado los SNI para identificarlos el resultado sería del 40%.

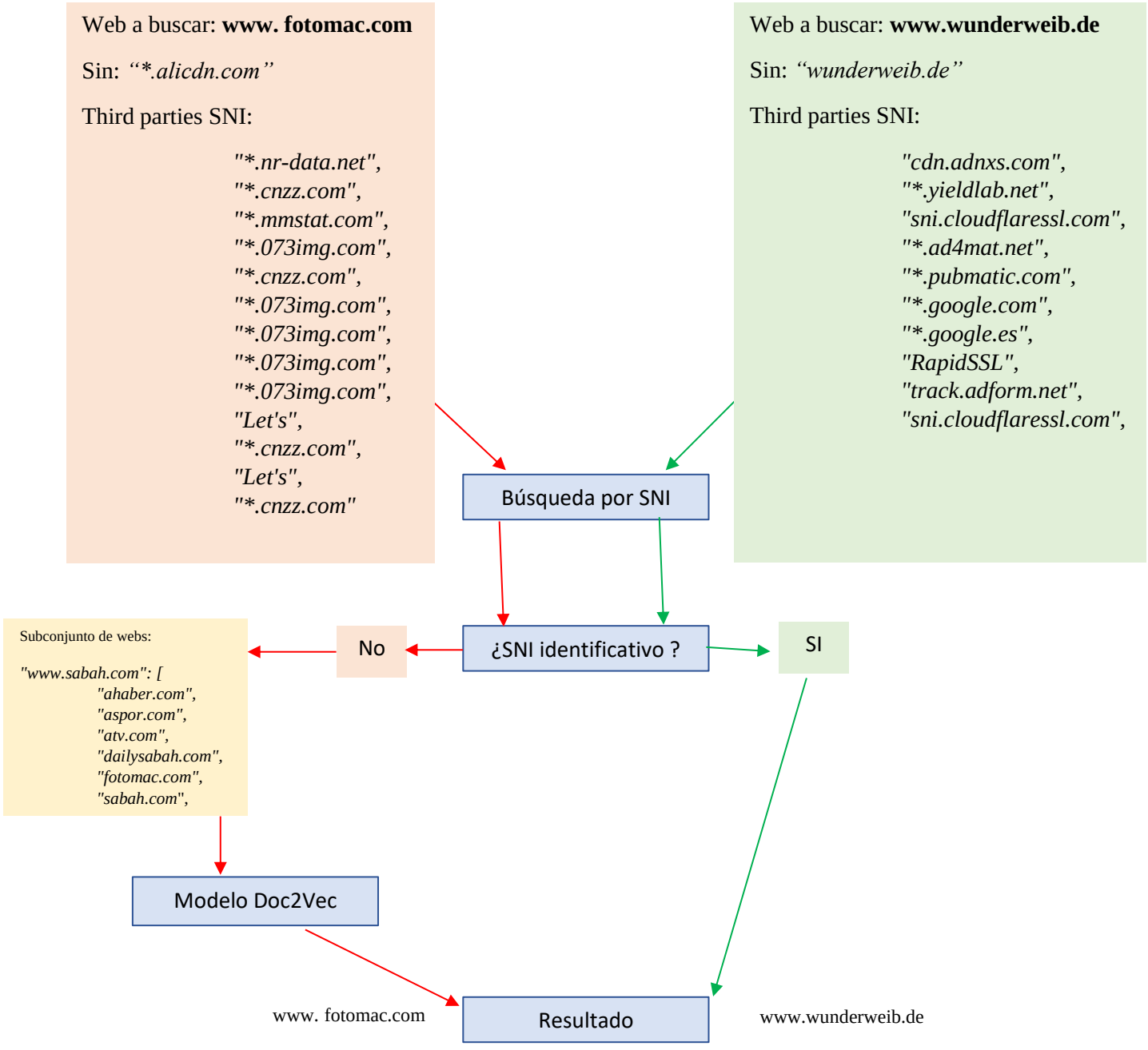

Figura 11: Ejemplo Herramienta de identificación

5.2 Herramienta de experiencia de usuario

Para desarrollar esta herramienta se hizo un estudio de cuál es el navegador más usado por la población[15]. Concluimos que Google Chrome es el navegador más usado. Una vez que descubrimos esto empezamos la investigación de algúna Api que pudiera comunicarse con esta aplicación para poder saber la experiencia de usuario.

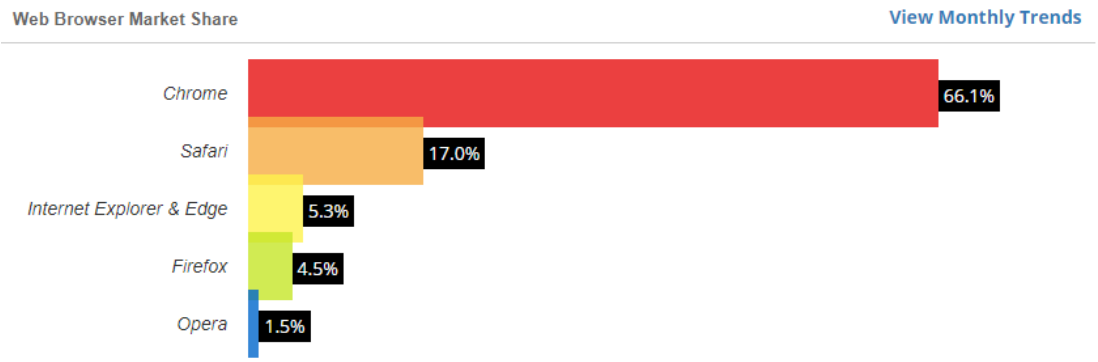

Figura 12: Gráfico Uso De Browsers

Descubrimos la Api de Lighthouse[16] de Google. Esta api lo que hace es abrir una instancia de Google Chrome y hace la búsqueda de la web que nosotros le indicamos. Después de las distintas pruebas que realiza la Api nos devuelve un informe de cómo es el rendimiento de esta página web en estas condiciones. Con este informe extraemos la información más relevante y la usamos para crear un número del 0 al 100 que muestra la experiencia de usuario. Esta Api es muy potente y nos permite acceder a más métricas que en nuestra aplicación no nos interesan pero que almacenamos por si quisiéramos hacer un estudio más adelante.

5.2.1 Resultados:

Después de la ejecución, Lighthouse, nos entrega un informe en formato Json del cual podemos obtener todos estos parámetros individualmente e interpretarlos como nosotros veamos necesario.

A parte de este informe en Json también puede crear un HTML para poder ver los resultados gráficamente.

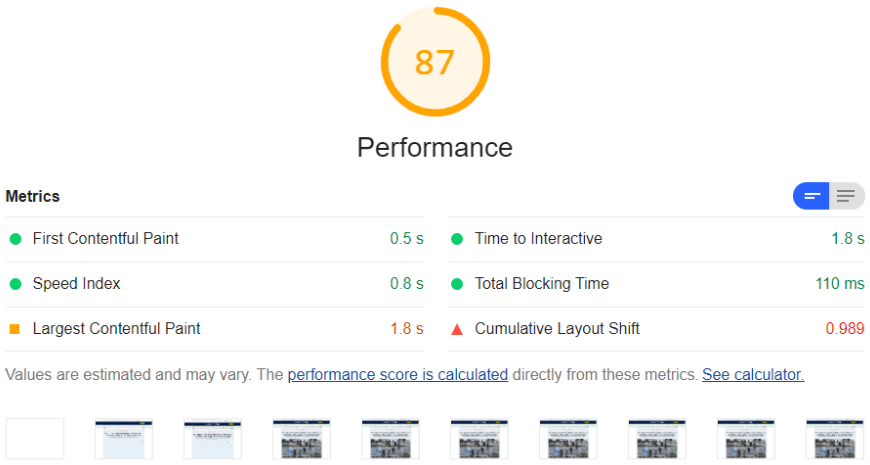

Figura 13: Ejemplo informe Experiencia de usuario

5.3 Herramienta final

La herramienta final es una combinación de ambas herramientas previamente mencionadas.

Primero usaremos la primera para identificar la web a partir de unos third parties. El resultado de esta búsqueda la pasaremos la segunda herramienta que nos devolverá un número del 0 al 100. Ejemplo:

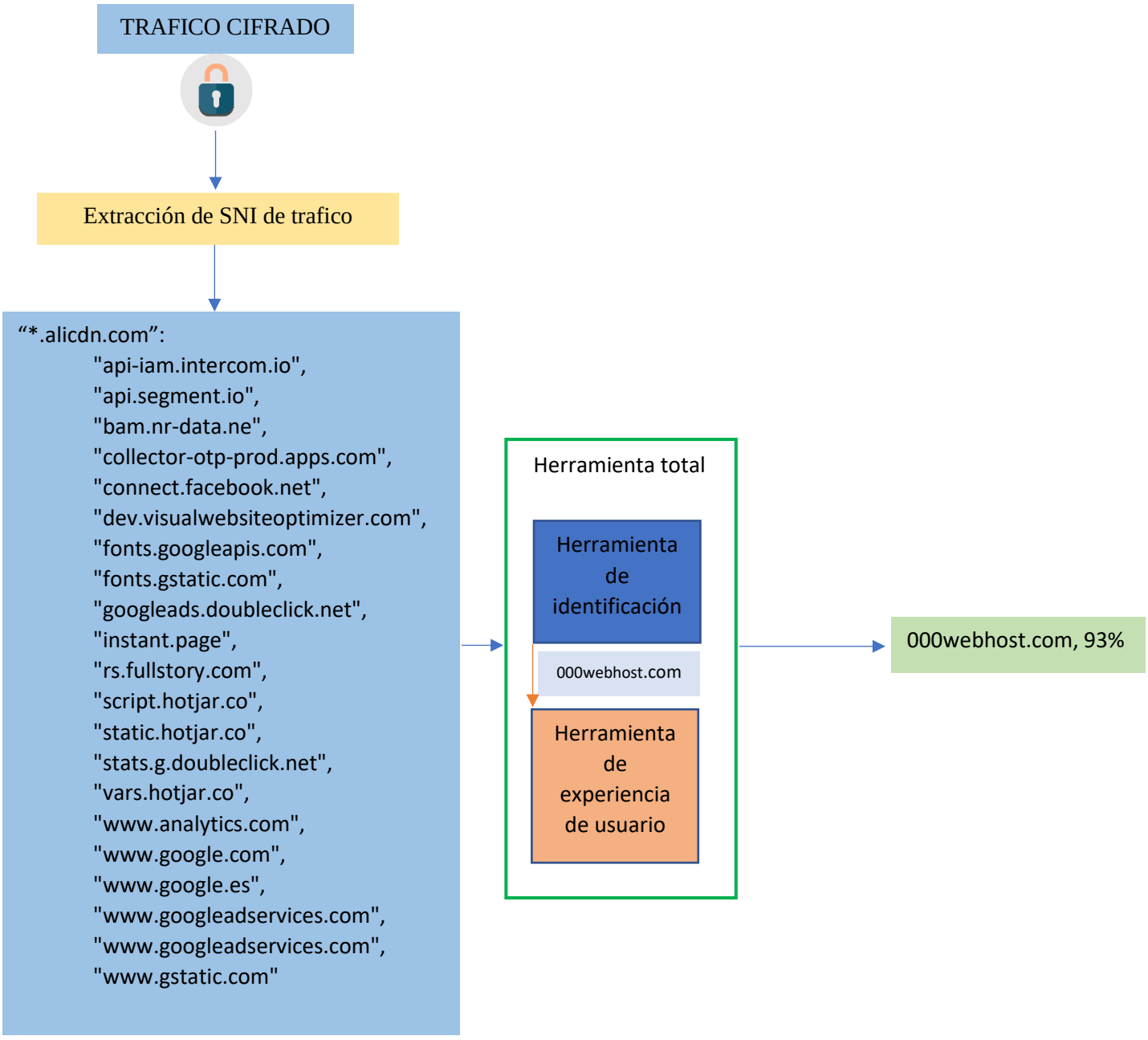

Figura 14: Ejemplo Herramienta Total

6. Informe de sostenibilidad

Hay muchos factores a tener en cuenta cuando se trata de la sostenibilidad en la especialidad de Tecnologías de la Información (TI). La principal suele ser la de la sostenibilidad medioambiental. Una de las asignaturas pendientes es reciclar bien la energía y recursos que consume en *hardware* necesario para programar y ejecutar los programas.

Con respecto al aspecto económico se analizará los costes de implementación y de mantenimiento de un producto de *software* como el que se está desarrollando, se ha analizado el mercado para que el producto se innovador y diferencial respecto a otras soluciones. Finalmente, también se analizará la viabilidad del proyecto con fines lucrativos.

Finalmente, con respecto al aspecto social, se usa herramientas colaborativas. También se diseña el *software* para que sea transparente, accesible y ético ya que este tiene la intención de ayudar.

6.1 Dimensión Social

Este proyecto presenta una oportunidad muy buena para aprender sobre las relaciones sociales y la situación actual de las redes por las que todos navegamos. También es una oportunidad para aprender sobre la gestión de proyectos de *software* y de tener experiencia con nuevos lenguajes de programación.

Cada vez más las personas usan la red para multitud de actividades, desde ver qué tiempo va hacer mañana como transacciones bancarias. Los estudios relacionados con la red son necesarios para detectar cuando el usuario está contento o está teniendo un problema con esta. Esto permite que los proveedores de conectividad puedan mejorar la experiencia que tiene el usuario navegando por su red. También se ha puesto de manifiesto la importancia de la privacidad del usuario al navegar por la red. Esto hace que la herramienta creada sea una muy importante herramienta que incorporar a los proveedores de red ya que permite hacer los estudios que son vitales para el avance de la red sin sobrepasar la línea de la privacidad del usuario.

6.2 Dimensión económica

El apartado de presupuesto del proyecto incluye los recursos necesarios para el desarrollo y mantenimiento la herramienta, por lo tanto, no hay ningún malgasto innecesario de dinero o recursos.

Los proveedores de conectividad gastan una gran cantidad de dinero contratando analistas para que analicen la red para poder mejorarla y poder tener a sus clientes más satisfechos con los servicios contratados. La herramienta que se ha desarrollado puede ahorrarles una buena cantidad de dinero al no tener que tener esas personas trabajando en ese aspecto y tenerlas en mejorar las infraestructuras que derivan de aplicar el estudio de la herramienta.

6.3 Dimensión medioambiental

El único aspecto negativo que puede tener el desarrollo y ejecución de la herramienta es el coste energético. Este coste solo se tiene en cuenta en las ejecuciones locales, mientras se desarrolla el proyecto, ya que una vez desplegado en los servidores que ya poseen los clientes no aumentaras el consumo energético que ya están generando.

7. Conclusión

En este proyecto se han creado dos herramientas para usar con el tráfico de red. Una identifica la web que un usuario está visitando. Esta identificación se hace sin violar la confidencialidad del cifrado de estas webs. La segunda nos proporciona información sobre la experiencia que tienen los usuarios después de visitar estas webs. Con todo eso se puede tener una muy buena visión de cómo está funcionando una red.

Con los estudios y pruebas realizados a lo largo del desarrollo de este proyecto se ha demostrado la viabilidad de la identificación de webs a partir del tráfico cifrado generado por esta. Gracias a los third parties a los que se conecta esta web y el SNI de esta, es posible y fiable. Se ha conseguido que un 88% de webs visitadas se haya podido averiguar su dirección con solo el tráfico cifrado que genera.

A pesar de que el proyecto se ha desviado de su premisa inicial, creo que ha resultado en un buen método para poder descubrir el tráfico de una red real. Los resultados obtenidos nos indican que aparte de ser un buen ejercicio de estudio es posible su despliegue en condiciones normales de cualquier red.

La herramienta de identificación es la que más importancia se ha dado en este proyecto ya que es la más innovadora de ambas e interesante.

7.1 Futura investigación

Para una futura investigación, sería muy interesante seguir investigando más sobre los certificados de estas webs para buscar más parámetros que identifiquen la web o que puedan ser usados para la identificación.

Otra vía de estudio, que se desvía más del propósito de saber a qué webs se conecta un usuario en el tráfico cifrado, sería qué más información de una web podemos obtener sabiendo los third parties de esta. Algo que hemos descubierto durante este proyecto es que hay mucha información de las webs que se pasa por alto. En este caso se podría usar la información con la que hemos trabajado a lo largo del desarrollo, como el número, frecuencia y cambio con el tiempo de sus third parties, para obtener más información de cómo se comporta una web.

8. Referencias (bibliografía)

[1] Raygun. https://raygun.com/blog/best-real-user-monitoring-tools Ultima visita: 12/10/2020

[2] Raygun Real User Monitoring. https://raygun.com/platform/real-user-monitoring Ultima visita: 12/10/2020

[3] SOASTA. https://www.akamai.com/es/es/products/performance/mpulse-real-user-monitoring.jsp Ultima visita: 12/10/2020

[4] Pingdom. https://www.pingdom.com Ultima visit: 12/10/2020

[5] New Relic. https://newrelic.com Ultima visita: 12/10/2020

[6] Scrum. https://www.scrum.org/resources/what-is-scrum Ultima visita: 12/10/2020

[7] Git. https://hackmd.io/@ETrs8IH-TXGRgF0lUhdYMg/S1J3KABIH Ultima visita: 29/09/2020

[8] GitHub. https://github.com/about Ultima visita: 29/09/2020

[9] Precio internet: https://ofertas.jazztel.com/fibra-optica Ultima visita: 12/10/2020

[10] Precio electricidad: https://tarifasgasluz.com/comparador/precio-kwh Ultima visita: 12/10/2020

[11] Inteligencia artificial (AI): Estado del arte…: https://planetachatbot.com/inteligencia-artificial-ai-estado-del-arte-67b325c7bbfc Ultima visita 07/01/2021 Ultima visita: 07/01/2021

[12] Doc2Vec: https://radimrehurek.com/gensim/auto_examples/tutorials/run_doc2vec_lee.html Ultima visita: 07/01/2021

[13] Word2Vec: https://radimrehurek.com/gensim/models/word2vec.html Ultima visita: 07/01/2021

[14] Measuring the Quality of Experience of Web user: https://c3lab.poliba.it/images/8/83/Webqoe-ccr-16.pdf Ultima visita: 07/01/2021

[15] Browser & Platform Market Share: https://www.w3counter.com/globalstats.php Ultima visita: 07/01/2021

[16] Lighthouse: https://developers.google.com/web/tools/lighthouse?hl=es Ultima visita: 07/01/2021

[17] Lighthouse Scoring Calculator: https://googlechrome.github.io/lighthouse/scorecalc/ Ultima visita: 07/01/2021

[18] The Architecture of Word2Vec: https://medium.com/@vishwasbhanawat/the-architecture-of-word2vec-78659ceb6638 Ultima visita: 07/01/2021

[19] Representación 3 dimensiones Word2Vec: https://images2.programmersought.com/347/b5/b52e1c2245cb830ca4602b5469730a33.png Ultima visita: 07/01/2021

[20] Bolsa de palabras: https://en.wikipedia.org/wiki/Bag-of-words_model#:~:text=The%20bag%2Dof%2Dwords%20model,word%20order%20but%20keeping%20multiplicity. Ultima Visita 16/01/2021

[21] Browser identification base don encrypted traffic: https://www.atlantis-press.com/article/25862501.pdf Ultima Visita 16/01/2021