



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Gandia

Automatización del proceso de compra de productos en  
comercios electrónicos

Trabajo Fin de Grado

Grado en Tecnologías Interactivas

AUTOR/A: Espínola Rodríguez, Marcelo Antonio

Tutor/a: Palanca Cámara, Javier

CURSO ACADÉMICO: 2022/2023

## INDICE

1. Agradecimientos .....	3
2. Introducción .....	7
1.1 Introducción .....	7
1.2 Motivación .....	9
1.3 Objetivos .....	9
1.4 Metodología .....	11
3. Estado del arte .....	12
2.1 Streetwear .....	12
2.2 Extensiones .....	13
2.3 Extensiones de compra existentes .....	15
2.4 Bots .....	16
2.5 Bots existentes .....	16
2.6 Sistemas de seguridad de las webs .....	16
2.7 Tecnologías WEB .....	20
3. Desarrollo .....	24
3.1 Metodología de trabajo .....	24
3.2 Requisitos .....	25
3.3 Diseño lógico .....	27
3.4 Diseño visual .....	29
3.5 Implementación .....	34
3.6 Casos de uso .....	38
4. Conclusiones y trabajo futuro .....	49
5. Objetivos de desarrollo sostenible .....	50
4. Relación con las asignaturas del grado .....	51
5. Anexos .....	52
Manual de usuario .....	52
6. Bibliografía .....	56

## INDICE DE ILUSTRACIONES

Ilustración 1. Lista de extensiones favoritas .....	13
Ilustración 2. Huella estándar de un navegador .....	17
Ilustración 3. Alerta de una ventana de navegador .....	19
Ilustración 4. Ejemplo de código Javascript .....	21
Ilustración 5. Ejemplo de código HTML .....	22
Ilustración 6. Ejemplo de código CSS .....	23
Ilustración 7. Fases del desarrollo.....	24
Ilustración 8. Ejemplo de añadir al carro en Snipes.es .....	25
Ilustración 9. Diseño lógico del servidor .....	27
Ilustración 10. Diseño lógico de la extensión .....	28
Ilustración 11. Diseño de la página de acceso .....	29
Ilustración 12. Diseño de la página de inicio .....	30
Ilustración 13. Diseño de la página de un módulo en específico.....	31
Ilustración 14. Diseño general de la pagina de módulos .....	31
Ilustración 15. Diseño de la página de perfiles de envío/pago .....	32
Ilustración 16. Diseño de la página de configuración.....	33
Ilustración 17. Ejemplo de implementación de reCaptcha .....	36
Ilustración 18. Ejemplo de manifest.json donde se ejecutan dos scripts en una web específica ..	36
Ilustración 19. Página de creación de perfil de envío .....	38
Ilustración 20. Resultado de la exportación .....	39
Ilustración 21. Exportación del perfil .....	39
Ilustración 22. Se añade un webhook en la sección correspondiente. ....	40
Ilustración 23. webhook.site enseñando el funcionamiento del webhook.....	40
Ilustración 24. Ejemplo de rellenado de perfil.....	42
Ilustración 25. El módulo de AW-LAB en acción.....	43
Ilustración 26. Configuración de AW-LAB.....	43
Ilustración 27. El módulo de Office/Offspring en funcionamiento .....	44
Ilustración 28. Configuración del módulo de Office .....	44
Ilustración 29. Hipervínculo de PayPal proporcionado por el módulo.....	45
Ilustración 30. Actualización del frontal debido a la compra .....	45
Ilustración 31. Webhook enviado por la compra realizada.....	46
Ilustración 32. El módulo de Shopify añadiendo al carrito .....	47
Ilustración 33. El módulo de Shopify nos proporciona un hipervínculo de PayPal .....	47

# 1. Agradecimientos

Quisiera empezar este apartado expresando mi más sincero agradecimiento a las personas e instituciones que han hecho posible la realización de este Trabajo de Fin de Grado.

En primer lugar, quiero agradecer a mi pareja, quien ha sido mi pilar fundamental en los momentos más críticos de este proceso. Tu apoyo incondicional y tu capacidad para entender y compartir mis angustias y alegrías han sido insustituibles. Sin ti, este camino habría sido mucho más arduo.

También quisiera expresar mi gratitud a mi tutor, quien no solo ha guiado este trabajo desde un punto de vista académico, sino que también me ha proporcionado los mejores consejos y perspectivas para enfrentar los retos que se me han presentado. Tu experiencia y sabiduría han sido de un valor incalculable para mí.

Mi reconocimiento también se extiende a la Universidad Politécnica de Valencia (UPV). Estudiar en esta institución ha sido una oportunidad invaluable que ha enriquecido mi vida tanto profesional como personalmente. Agradezco a la UPV por abrirme las puertas a un mundo de conocimiento y posibilidades.

Por último, pero no menos importante, quiero expresar mi agradecimiento a la serie de profesores que durante mi grado han sido más que excelentes. Cada uno de ustedes ha dejado una huella imborrable en mi formación y en mi vida. Vuestra pasión por la enseñanza y vuestro compromiso con la excelencia académica son verdaderamente inspiradores.

Gracias a todos por vuestro apoyo, orientación y por ser parte esencial en este importante capítulo de mi vida.

- Resumen

En este trabajo se pretende desarrollar una extensión de navegador en la cual un usuario pueda realizar compras de productos limitados en un amplio abanico de páginas web que se dedican al comercio online. Se le proporcionará la información necesaria y visitando un producto este se comprará automáticamente. Se podrá seleccionar el método de pago, la dirección de entrega y diferentes modos de 'checkout'.

Para realizar el trabajo se va a utilizar la infraestructura proporcionada por Google para la creación de extensiones con su Manifest V2 (JavaScript, CSS y HTML). Para poder proporcionar los módulos de las diferentes webs se utilizará un servidor HTTP escrito en Node.js y con el middleware Express.js siendo la comunicación mediante peticiones HTTP.

Palabras clave: JavaScript, CSS, HTML, extensión, Chrome, scrapping.

- Abstract

In this work we intend to develop a browser extension in which a user can make purchases of limited products in a wide range of web pages that are dedicated to online commerce. The user will be provided with the necessary information and by visiting a product it will be purchased automatically. It will be possible to select the payment method, the delivery address and different checkout modes.

To perform the work, we will use the infrastructure provided by Google for the creation of extensions with its Manifest V2 (JavaScript, CSS and HTML). In order to provide the modules of the different webs we will use an HTTP server written in Node.js and with the middleware Express.js being the communication through HTTP requests.

Keywords: JavaScript, CSS, HTML, extension, Chrome, web scrapping.

- Resum

En aquest treball es pretén desenvolupar una extensió de navegador en la qual un usuari pugui realitzar compres de productes limitats en un ampli ventall de pàgines web que es dediquen al comerç en línia. Se li proporcionarà la informació necessària i visitant un producte aquest es comprarà automàticament. Es podrà seleccionar el mètode de pagament, l'adreça de lliurament i diferents maneres de 'checkout'.

Per a fer el treball s'utilitzarà la infraestructura proporcionada per Google per a la creació d'extensions amb la seva Manifest V2 (JavaScript, CSS i HTML). Per a poder proporcionar els mòduls de les diferents webs s'utilitzarà un servidor HTTP escrit en Node.js i amb el middleware Express.js sent la comunicació mitjançant peticions HTTP.

Palabras clave: JavaScript, CSS, HTML, extensió, Chrome, scrapping.

## 2. Introducción

### 1.1 Introducción

Desde la popularización del comercio electrónico, las empresas que se dedican a la venta de ropa, calzado y accesorios han experimentado una transformación sin precedentes al aprovechar las oportunidades que brinda esta nueva plataforma digital. La transición de realizar compras de manera presencial a través de tiendas físicas a efectuarlas cómodamente en línea se ha convertido en una tendencia en constante crecimiento. Consecuentemente, estas compañías han reconocido la necesidad de expandir su presencia en el mundo virtual para llegar a una audiencia cada vez más conectada y ávida de adquirir productos exclusivos.[1]

Dentro de este mercado, han surgido productos *hypebeast* que se caracterizan por su escasez en el stock, siendo especialmente diseñados en colaboración con celebridades o prestigiosos diseñadores. Estos artículos, como el modelo New Balance 2002R Protection Pack Rain Cloud[2] o el reloj SO33R100 de Omega[3] en colaboración con Swatch llamado Mission to Mars, ejemplifican el fenómeno en el que se han convertido estos productos exclusivos. Pese a que su precio de venta inicial es de 150\$ y 260\$ respectivamente, sus unidades se agotan en cuestión de segundos tras su lanzamiento al mercado [4][5], dejando a muchos consumidores con la frustrante sensación de perder la oportunidad de adquirirlos. La consecuencia de esta demanda desbordante y la limitada disponibilidad es la aparición del mercado de reventa, donde los precios pueden alcanzar cifras exorbitantes.[6][7]

En este contexto surge el presente proyecto, que busca brindar una solución a esta problemática mediante el desarrollo de una aplicación innovadora. La propuesta consiste en proporcionar a los usuarios una herramienta que agilice y optimice el proceso de compra, permitiéndoles obtener de manera efectiva y rápida cualquier producto que deseen. Al proporcionar a la extensión una dirección de envío, un método de pago y el producto objetivo, los usuarios tendrán la posibilidad de competir en igualdad de condiciones en el lanzamiento de estos artículos codiciados, aumentando significativamente sus probabilidades de éxito. La confirmación de pedido obtenida tras completar el proceso satisfactoriamente, representa el objetivo clave de esta herramienta, que busca empoderar a los consumidores para que puedan acceder a productos exclusivos sin tener que recurrir a la reventa y a precios desorbitados.

A lo largo de este documento, se detallará el proceso de desarrollo de la extensión, abarcando desde la concepción de la idea hasta la implementación y puesta en marcha. Se describirán las metodologías y tecnologías utilizadas, así como los retos enfrentados y las soluciones propuestas. Asimismo, se analizarán los resultados obtenidos y se reflexionará sobre el impacto que esta aplicación podría tener en la experiencia de compra de los usuarios y en la industria del comercio electrónico.

Este proyecto busca contribuir a la evolución y mejora del comercio electrónico en el ámbito de los productos *hypebeast*, al brindar a los consumidores una herramienta para acceder a los artículos más exclusivos de manera eficiente y justa. A medida que continuamos adentrándonos en la era digital, la importancia del comercio electrónico no solo se consolida, sino que se

amplifica, y esta investigación busca ser un pequeño paso hacia una experiencia de compra más inclusiva y accesible para los amantes de la moda y el estilo de vida hypebeast [8].

## 1.2 Motivación

Mi principal motivación para embarcarme en este emocionante proyecto de Trabajo de Fin de Grado ha sido la posibilidad de obtener y disfrutar personalmente de esos codiciados productos hypebeast que tanto anhelo. Como bien mencioné anteriormente, la adquisición de estos artículos exclusivos sin el apoyo de herramientas especializadas se ha convertido en una tarea ardua y, en ocasiones, frustrante. Por lo tanto, la idea de desarrollar una aplicación que me permita competir en igualdad de condiciones en el mercado y aumentar mis posibilidades de obtenerlos ha sido una fuente inagotable de entusiasmo y dedicación.

Si bien mi motivación inicial fue centrada en la satisfacción personal, pronto se sumó otro aliciente que avivó aún más mi interés en este proyecto: la oportunidad de adentrarme en el fascinante mundo de JavaScript. Este lenguaje de programación, con su amplia trayectoria y prometedor futuro debido a runtimes innovadores como Bun[9] y Deno[10], me atrajo desde el principio. Aprender a utilizar JavaScript[11] en profundidad y dominar sus características avanzadas se convirtió en una meta adicional para mi desarrollo profesional. La perspectiva de poder aplicar este conocimiento en un proyecto real y en el contexto del comercio electrónico resultó especialmente emocionante y desafiante.

Por último, pero sin lugar a dudas igual de relevante, el deseo de consolidar mis habilidades en HTML[12] y CSS[13] fue otro de los pilares fundamentales que impulsaron mi decisión de desarrollar esta extensión. La plataforma proporcionada por Chrome para la creación de extensiones se caracteriza por su accesibilidad y simplicidad, lo cual me permitió concentrar mis esfuerzos en perfeccionar mis conocimientos en los lenguajes mencionados. El proceso de desarrollo de la extensión no solo me brindó una oportunidad para aplicar los conceptos teóricos aprendidos, sino que también me permitió adquirir nuevas habilidades y técnicas, lo cual enriqueció significativamente mi conjunto de habilidades como futuro profesional en el campo de la informática.

Mi motivación para llevar a cabo este Trabajo de Fin de Grado ha sido multidimensional y profundamente arraigada. Desde el deseo de obtener productos exclusivos para mi disfrute personal, pasando por el anhelo de aprender y dominar JavaScript en un contexto práctico y relevante, hasta la oportunidad de consolidar y expandir mis conocimientos en el desarrollo web, cada aspecto ha contribuido a nutrir mi pasión y dedicación en este proyecto. A través de esta investigación, espero no solo satisfacer mis propios objetivos y metas, sino también contribuir al crecimiento y mejora del comercio electrónico para el beneficio de otros entusiastas de los productos hypebeast.

## 1.3 Objetivos

El presente Trabajo de Fin de Grado tiene como objetivo principal **desarrollar una extensión que permita realizar compras de manera automática en diversas páginas web que ofrecen productos exclusivos**, con el fin de facilitar a los usuarios la **adquisición de artículos codiciados y de edición limitada**. Este objetivo central se enmarca en la idea de empoderar a los consumidores y aumentar sus posibilidades de éxito en la obtención de estos productos tan deseados.

Para lograr este propósito, es imprescindible abordar una serie de subobjetivos que aseguren el éxito y la eficiencia de la extensión:

- **Estudiar el mercado de páginas web que ofrecen productos exclusivos:** Este primer paso implica llevar a cabo una investigación exhaustiva del panorama de tiendas en línea que se especializan en la venta de artículos hypebeast y de edición limitada. Identificar las plataformas más relevantes, sus características distintivas y su funcionamiento es esencial para adaptar la extensión y garantizar su compatibilidad y efectividad en cada caso.
- **Comprobar la viabilidad del proceso de compra automático:** Una vez identificadas las páginas web objetivo, es necesario evaluar la viabilidad técnica y legal de automatizar el proceso de compra en cada una de ellas. Asegurar que la extensión funcione correctamente y cumpla con las políticas y términos de uso de cada sitio es fundamental para garantizar una experiencia de usuario óptima y evitar posibles problemas legales.
- **Analizar el estado del arte de las herramientas/extensiones existentes:** Investigar y analizar las herramientas o extensiones similares existentes en el mercado permitirá conocer las fortalezas y debilidades de las soluciones ya disponibles. Esto servirá como base para mejorar y potenciar la propuesta de la extensión desarrollada, buscando ofrecer características innovadoras y superar las limitaciones identificadas en otras soluciones.
- **Optimizar los procesos de compra:** El rendimiento y la eficiencia son aspectos clave en la automatización de compras. Se buscará optimizar los procesos de compra para que la extensión pueda llevar a cabo transacciones de manera rápida y precisa, minimizando los tiempos de respuesta y aumentando las probabilidades de éxito en la adquisición de los productos.
- **Verificar el funcionamiento de los procesos de compra:** Se llevarán a cabo rigurosas pruebas y simulaciones para asegurar que la extensión funcione de manera confiable y consistente en diversos escenarios y situaciones. La verificación del funcionamiento adecuado garantizará que los usuarios puedan confiar en la extensión para obtener los productos deseados de manera segura y efectiva.

A través de la consecución de estos objetivos principales y subobjetivos, este Trabajo de Fin de Grado aspira a brindar una solución integral y efectiva para la automatización de compras de productos exclusivos, proporcionando a los usuarios una herramienta confiable para acceder a artículos de manera más sencilla y justa. Al cumplir con estos propósitos, este proyecto busca no solo alcanzar el éxito académico, sino también contribuir al avance y mejora de la experiencia de compra en el ámbito del comercio electrónico.

## 1.4 Metodología

Para desarrollar este proyecto he decidido utilizar la metodología en cascada, en la cual se definen unas fases que deben ser completadas de manera secuencial. He decidido que fuera así debido a que puedo separar el trabajo en diferentes bloques que pueden ser completados independientemente del resto y aunque no pueden funcionar de manera independiente esto me permite dar por finalizadas secciones del trabajo y poder seguir con las siguientes. El primer hito a desarrollar será la interfaz de la extensión, una vez completada y funcional se realizará un servidor que se encargará de servir los diferentes módulos de las páginas web que se implementen y por último se realizarán los módulos que pertenecerán a una página web en específico.

### 3. Estado del arte

En este apartado se hace referencia a las modas y se profundiza en la situación de las diferentes extensiones disponibles en el mercado. También se analizan herramientas para su desarrollo.

#### 2.1 Streetwear

El streetwear, como estilo de ropa informal, ha experimentado un ascenso vertiginoso en popularidad desde sus orígenes en los años noventa, pero ha sido durante la década de 2010 cuando ha adquirido verdadera tracción. Uno de los factores que ha propiciado este crecimiento exponencial ha sido el auge de las redes sociales, en las cuales las personas, especialmente influencers, pueden compartir y exhibir su estilo personal, inspirando a otros y generando tendencias. Esto ha llevado a una mayor difusión y adopción del streetwear como una expresión de identidad y pertenencia a una comunidad con intereses afines.

Surgió inicialmente de la moda hip-hop y la cultura del surf californiana, fusionando elementos de la ropa deportiva, el punk, el skateboarding y la moda callejera japonesa. Sin embargo, con el tiempo, ha evolucionado y ha incorporado influencias de la alta costura, demostrando su capacidad para trascender las barreras tradicionales de la moda y adquirir un estatus cultural y artístico relevante.

El enfoque recae en prendas informales y cómodas, como vaqueros, camisetas, gorras de béisbol y zapatillas deportivas. Además, la escasez intencionada en la producción de ciertos productos ha contribuido a crear una sensación de exclusividad y deseo entre los entusiastas de este estilo. La lealtad a ciertas marcas es común en la comunidad del streetwear, ya que los seguidores buscan obtener productos exclusivos que reflejen sus gustos y personalidad.

Es importante destacar que las tendencias y productos demandados son altamente dinámicos y cambiantes. Los productos mencionados en la lista como tendencia en el momento actual pueden haber evolucionado o cedido su lugar a nuevos elementos en el mercado.

El impresionante crecimiento como industria también se refleja en su valor de mercado. Según las cifras proporcionadas, el valor de mercado alcanzó los 185 mil millones en 2021, aumentando a 187 mil millones en 2022, y se estima que seguirá creciendo para alcanzar los 230 mil millones para el año 2028. Este crecimiento sostenido subraya la importancia y relevancia cada vez mayor que está adquiriendo en el mundo de la moda y el estilo de vida.

El streetwear se ha convertido en mucho más que un simple estilo de ropa, ha evolucionado para convertirse en una poderosa forma de expresión cultural y un fenómeno global que influye en la moda contemporánea. Su capacidad para fusionar la moda de la calle con elementos de lujo y exclusividad ha generado una comunidad apasionada y dedicada, impulsando así su constante crecimiento económico y cultural en el panorama mundial.[14]

## 2.2 Extensiones

Las extensiones[15] son programas, generalmente pequeños, que ayudan a mejorar la experiencia de un usuario que utiliza un navegador. Usualmente las extensiones tienden a apuntar hacia un objetivo claro y conciso, algunos ejemplos serían los de la Ilustración 1:

The image shows a screenshot of the Chrome Web Store's 'Favorites of 2022' section. It features a grid of eight extension cards, each with a logo, a brief description, a star rating, and a download count. A 'View all' button is located in the top right corner.

Extension Name	Star Rating	Download Count
RoPro - Enhance Your Roblox ...	★★★★★	22,012
Workona Tab Manager	★★★★★	3,192
MyBib: Free Citation Generator	★★★★★	1,188
CrxMouse Chrome™ Gestures	★★★★★	11,208
eJOY AI Dictionary	★★★★★	4,357
LINER ChatGPT: AI Copilot for ...	★★★★★	5,719
Compose AI: AI-powered Writi...	★★★★★	159
SwiftRead - read faster, learn ...	★★★★★	1,080

*Ilustración 1. Lista de extensiones favoritas*

Como podemos observar en las más descargadas de 2022[16] cada una de ellas cumple con los objetivos recién citados: tienen un objetivo claro y conciso.

Desde que se lanzaron en 2010, las extensiones de Chrome han tenido un rol fundamental en la historia en constante evolución del navegador Google Chrome. Básicamente, estas extensiones son pequeños programas que los usuarios pueden agregar a su navegador para hacerlo más funcional y personalizado. El hito más importante ocurrió en diciembre de 2010, cuando se introdujeron oficialmente las extensiones. Esto abrió la puerta para que los usuarios exploraran una variedad cada vez mayor de herramientas y funciones que realmente mejoraron la forma en que navegaban por la web, haciéndola más productiva y cómoda.

Con el paso de los años, la Chrome Web Store, que es la plataforma donde se pueden encontrar y descargar estas extensiones de Chrome, se llenó de opciones para todos los gustos. Desde bloqueadores de anuncios hasta administradores de contraseñas y herramientas de productividad, estas extensiones permitieron que los usuarios adaptaran su experiencia de navegación según sus necesidades específicas.

Por supuesto, en este proceso surgieron retos en términos de seguridad y privacidad, ya que el uso de extensiones se volvió más popular. En respuesta, Google comenzó a poner más atención en proteger a los usuarios, implementando revisiones y validaciones más estrictas para las extensiones antes de permitir su inclusión en la Chrome Web Store. A pesar de estos esfuerzos, hubo ocasiones en las que extensiones maliciosas lograron evadir estos controles, lo que llevó a Google a reforzar sus medidas de seguridad y a eliminar estas extensiones problemáticas.

Conforme la preocupación por la privacidad en línea aumentó, Google ajustó sus políticas en relación a las extensiones. Se volvieron más rigurosos en términos de qué tipo de datos podían recopilar estas extensiones y cómo podían utilizarlos. Esto reflejó la evolución en las inquietudes de los usuarios, quienes deseaban tener más control sobre la información que las extensiones pueden acceder y manejar, a medida que se adentraban en un mundo digital cada vez más complejo y consciente de la privacidad.

Las tecnologías utilizadas son las mismas que se utilizarían para la creación de cualquier aplicación web:

1. HTML como lenguaje de marcado
2. CSS para proporcionar estilo
3. JavaScript para la lógica y scripting

Al desarrollar para Chrome tenemos acceso a las APIs de Chrome para realizar diferentes tipos de tareas relativamente útiles. Hay que comentar que, aunque se permita el acceso a una parte de las APIs de Google, no es posible acceder a todas hasta que la extensión está publicada y ha pasado por las reviews necesarias impuestas por Google. Podemos acceder a las API de mensajería entre content scripts, pero sería imposible acceder a las de los usuarios, ya que estas poseen información bastante más delicada.

En nuestro caso lo que realmente aporta valor es que podemos utilizar las APIs de JavaScript que ya vienen incluidas en casi todos los navegadores, especialmente Chrome.

### Estructura de una extensión

**Manifest.json:** Es el único archivo que obligatoriamente ha de tener un nombre fijado: manifest.json. Debe encontrarse en la raíz del proyecto y se encarga de guardar metadata importante, definir recursos, declarar permisos e identificar qué scripts han de ser ejecutados en segundo plano o en las páginas web.

**Service worker:** Es el encargado de atender los eventos del navegador. Se ejecuta en segundo plano y no puede acceder al contenido de ninguna página web cargada por el navegador. Se puede utilizar para, por ejemplo, comunicar datos de la extensión con un script que se esté ejecutando en una web.

**Content script:** Estos scripts se ejecutan en el contexto de una página web y por lo tanto pueden modificar la lógica y el DOM de estas.

**Páginas HTML:** Una extensión puede tener la cantidad de archivos HTML que desee, todas tienen el acceso a las APIs de JavaScript.

## 2.3 Extensiones de compra existentes

Las extensiones existentes[17][18] mantienen su funcionamiento en gran parte debido a que son altamente secretas y muy selectas a la hora de elegir usuarios.

Para poder entender cómo funcionan hay que explicar un par de cosas:

- Para poder acceder a una extensión debes de conseguir una invitación y acto seguido pagar un plan de uso, los precios varían entre 0€-50€ como pago inicial y 15€-30€ como pago mensual por la utilización, pudiéndose cancelar la mensualidad en cualquier momento. Normalmente utilizan pasarelas de pago externas como por ejemplo Stripe. Realizar el pago te permite acceder al grupo cerrado de usuarios pagadores que normalmente se mantiene en Discord.
- Solo las personas que hayan pasado por el proceso comentado con anterioridad pueden acceder al código fuente, normalmente es un archivo .zip, que, después de descomprimir, puedes añadir a tu navegador como extensión.

Esto provoca que sea la gente que realmente tiene un interés legítimo la que pueda conseguir las extensiones que desea. Esto imposibilita poder acceder al código fuente que, aunque esté ofuscado, podría dar alguna idea de cómo funciona a nivel lógico. Tampoco es conveniente compartir el código fuente públicamente. Los desarrolladores de las páginas webs podrían ver los métodos que se están utilizando en ese momento para realizar estas compras automáticas y bien desactivarlos, bien mermar su funcionamiento o actualizar la página web.

Lo que sí se puede saber es el feedback que proporcionan los usuarios mediante las redes sociales. Usualmente los usuarios comparten el éxito tenido usando las extensiones y nombran a los perfiles principales de las extensiones, es decir, el nivel de interacción que existe entre los usuarios y la extensión en las redes sociales te da a entender cómo está funcionando y en qué páginas web funciona mejor.

## 2.4 Bots

Un bot es un programa que intenta imitar el comportamiento humano para realizar ciertas tareas en internet. Los bots que existen para comprar los productos exclusivos intentan realizar la compras en el menor tiempo posible y utilizando la mayor cantidad de vulnerabilidades de las páginas web posible.

La diferencia principal entre una extensión y un bot, es que la primera se ejecuta en el navegador y por lo tanto está limitado por toda la seguridad que tenga implementada el mismo. Un bot sin embargo se ejecuta, normalmente en un navegador programado por los propios desarrolladores de los bots, como, por ejemplo, versiones ampliamente modificadas de Chromium [19]

Como ejemplo de seguridad vulnerada podemos hablar de CORS, como sabemos, este mecanismo permite que se puedan solicitar recursos de origen cruzado, es decir, de dominios diferentes. Este sistema de seguridad es desactivado en los navegadores de los bots, de manera que se permiten las peticiones con orígenes cruzados.

## 2.5 Bots existentes

Al igual que las extensiones, los bots pueden seguir funcionando debido a que son altamente selectivos, caros y secretos (por lo menos en cuanto a funcionamiento).

En cuanto a cómo se accede a ellos, podríamos decir que son iguales que las extensiones. La única diferencia, bastante importante, es que debido a que el bot es un ejecutable, este tiene muchísimo más acceso a la información que presenta tu ordenador y si es requerido el permiso de administración el control es casi absoluto. Esto se hace para poder controlar que hace el usuario con el programa que se le ha concedido, desde controlar qué programas tiene instalados, pasando por cuales está ejecutando, hasta saber qué hardware tiene el ordenador.

## 2.6 Sistemas de seguridad de las webs

Debería de ser una prioridad de las páginas web que la misma no se llene de bots a la hora de realizar un lanzamiento de un producto nuevo. La mayoría de las páginas web poseen sistemas para evitar que esto ocurra. Es lo que se suele llamar antibot.

Un antibot es un script que es capaz de reconocer a un bot y mermar su funcionamiento. La gran mayoría de las páginas web implementan algún tipo de antibot. Tanto los bots como las extensiones han de lidiar con este tipo de mecanismos, pero son los bots los que realmente se enfrentan a ellos, ya que, usualmente, la intervención humana requerida para el funcionamiento de una extensión es suficiente para saltar estos mecanismos.

Estos scripts recogen toda la información posible de los usuarios y con esta información intentan diferenciar entre un bot y un usuario legítimo.

En la web de SannySoft[20] se recoge casi en su totalidad la información disponible que ofrece un navegador cuando accede a una página web (Ilustración 2).

## Intoli.com tests + additions

Test Name	Result
User Agent (Old)	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
WebDriver (New)	missing (passed)
WebDriver Advanced	passed
Chrome (New)	present (passed)
Permissions (New)	prompt
Plugins Length (Old)	5
Plugins is of type PluginArray	passed
Languages (Old)	en-ES,es-ES,es,ca-ES,ca,en-GB,en-US,en
WebGL Vendor	Google Inc. (AMD)
WebGL Renderer	ANGLE (AMD, AMD Radeon RX 7900 XTX Direct3D11 vs_5_0 ps_5_0, D3D11)
Hairline Feature	missing
Broken Image Dimensions	16x16

*Ilustración 2. Huella estándar de un navegador*

En la imagen podemos observar parte de la información que ofrece el navegador de nuestro ordenador. Cada uno de estos parámetros sirve para al final poder generar una huella, esta huella debe de ser valida.

Si nosotros modificáramos el ‘User Agent’ del navegador a uno personalizado como por ejemplo “El navegador de la UPV2.0” una página web el día del lanzamiento de un producto no nos dejaría realizar una compra. Esto se debe a que, para los sistemas antibots, la huella no tiene validez.

Esto se puede dar con cualquier parámetro que ofrezca nuestro navegador. El objetivo principal es poder ofrecer esta información sin realmente tener un navegador abierto.

## Akamai

**Akamai**[21] es una empresa que se dedica a proporcionar servicios en la nube. Uno de ellos es el sistema antibot. Este era uno de los mejores sistemas que existían hasta que se hizo ingeniería inversa de su funcionamiento y se conoció como funcionaba.

Akamai recoge toda la información que consideran útil de tu navegador y la empaquetan en una petición /POST hacia sus servidores, ellos lo llaman 'sensor data', se puede esperar que sea información parecida a la Ilustración 3. Esta información va ofuscada y es la que utilizan para discernir una persona de un bot.

Se descubrió que esta información era enviada cada ciertos tics pero que una vez una petición contenía información valida las siguientes 10 peticiones que se hicieran contra la página web se daban por validas.

Por lo tanto, lo único que se debe hacer para salvar el antibot es generar un 'sensor data' valido y ofuscado que enviar cada 10 peticiones.

Esto es obviamente más complicado de lo que parece, pero es interesante saber cómo funcionan estos sistemas, ya que, aunque no se lidie con ellos de manera directa, se puede utilizar esta información para que sea más fácil la compra.

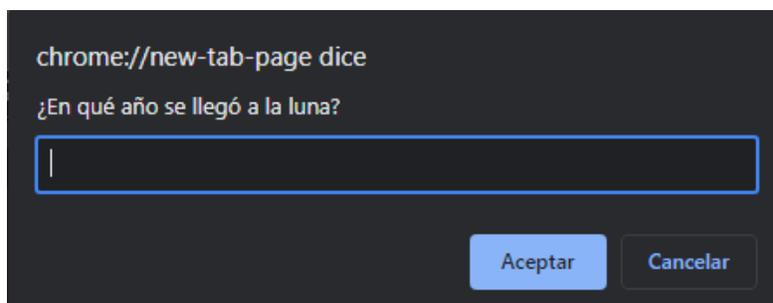
Un ejemplo de cómo podríamos utilizar esto sería: en Zalando, que utiliza Akamai, habría que navegar por la web de manera activa durante un instante antes de ejecutar la extensión en el lanzamiento del producto. Debido a que el 'sensor data' es siempre valido, podemos asegurarnos de que todo funcionará perfectamente y que, efectivamente, a la hora de ejecutar la extensión quedan las 10 peticiones validas por hacer.

## PerimeterX

**PerimeterX**[22] es una empresa estadounidense que se dedica a proporcionar soluciones de ciberseguridad. Es aclamada en el mundo de la detección de bots por su innovadora y efectiva forma de lidiar con los mismos, lo que le ha llevado a ganar varios premios como ‘Cybersecurity Excellence Awards’ o ‘SC Awards’ entre otros.

Como todos los antibots, PerimeterX recoge toda la información que es proporcionada por el navegador, sin embargo, donde realmente resalta su trabajo es en el análisis de la conducta de los visitantes de la web en cuestión. A continuación, voy a exponer algunos de los mecanismos que utiliza:

- Análisis del recorrido del usuario: un usuario no puede saltarse páginas en ningún proceso de compra. Por ejemplo: no puedes acceder a la página de pago sin haber pasado por rellenar los datos. Esto que puede parecer bastante simple hace que tengas que realizar todos los pasos para poder optar a la compra, en el caso de que quisieras hacer los pasos mediante peticiones, te obliga a hacerlas en orden.
- Métodos propios del objeto window del navegador: obliga a que renderices las páginas por las que pasas debido a que en las mismas hay llamadas a funciones como el window.prompt(), window.alert() o window.confirm(). Algunos de estos métodos llevan consigo la necesidad de que contestes alguna pregunta, o simplemente la confirmación ejecuta un callback que bloquea el proceso en caso de no ser ejecutado. En muchas ocasiones se utiliza para realizar preguntas, que han de responderse correctamente (Ilustración 3).



*Ilustración 3. Alerta de una ventana de navegador*

- Machine Learning: utilizan datos de como navegan las personas por la web para realizar un aprendizaje automático y poder bloquear todos aquellos comportamientos que discrepan con dicho aprendizaje. Un ejemplo sencillo sería comprobar que el ratón, cuando el usuario compra un producto, está relativamente cerca del botón de comprar o comprobar que el ratón esté en algún lugar que tenga sentido, en cuanto a coordenadas. Normalmente cuando un usuario utiliza una web no tiene el ratón en las esquinas (coordenadas 0,0).

Solo con los dos primeros puntos ya obtienes un sistema bastante robusto que, por lo menos, obliga a todos los bots a pasar por todas las páginas y con el tercero, cada vez que un usuario pasa por alguna de las páginas ha de ceñirse a un comportamiento general obtenido del groso de los visitantes. Es decir, aunque utilices algún sistema de automatización o bot, acabas compitiendo de una manera bastante igualitaria con el resto de usuarios.

## 2.7 Tecnologías WEB

### JavaScript

**JavaScript** es un lenguaje de programación esencial en el mundo actual, y su presencia es omnipresente en el entorno web. Como parte fundamental de las tecnologías de la World Wide Web (WWW), es el responsable de dotar de interactividad y dinamismo a la gran mayoría de páginas web existentes. Gracias a su capacidad para procesar el comportamiento de los elementos en el navegador, JavaScript ha revolucionado la forma en que interactuamos con las aplicaciones web.

Siendo un lenguaje de programación de alto nivel, JavaScript se adhiere al estándar **ECMAScript**[22], lo que garantiza su compatibilidad y portabilidad en distintos entornos y plataformas. La tipificación dinámica de JavaScript permite la flexibilidad en la definición de variables y la capacidad de adaptarse a diferentes tipos de datos durante la ejecución del programa. Además, JavaScript es un lenguaje multiparadigma, lo que significa que ofrece soporte tanto para la programación orientada a objetos como para la programación funcional e imperativa, lo que amplía su versatilidad y aplicabilidad en diversos contextos de desarrollo.

Un aspecto distintivo de JavaScript es su enfoque orientado a eventos, lo que le permite responder a interacciones del usuario y eventos del sistema, como clics del mouse o cambios en la página. Esto facilita la creación de aplicaciones web interactivas y dinámicas que responden en tiempo real a las acciones del usuario.

Otra característica poderosa de JavaScript es su capacidad para interactuar con el DOM (Document Object Model), lo que le permite manipular y modificar la estructura y el contenido de una página web de manera dinámica. Además, JavaScript cuenta con APIs (Application Programming Interfaces) que proporcionan funcionalidades específicas para el manejo de texto, fechas, expresiones regulares y ciertos tipos de estructuras de datos.

Aunque JavaScript es una herramienta poderosa para el desarrollo web, no está diseñado para acceder directamente al hardware del computador del usuario, como lo hacen los lenguajes de programación de bajo nivel. En cambio, JavaScript delega estas tareas al navegador web, que proporciona cierta información sobre el hardware, pero de manera restringida para preservar la seguridad y privacidad del usuario.

Más allá de su aplicación en entornos web, JavaScript ha experimentado un crecimiento exponencial gracias a la introducción de runtimes como Node.js. Esto ha permitido que JavaScript se utilice también en el desarrollo de servidores y aplicaciones de lado del servidor, lo que amplía significativamente su alcance y versatilidad.

A screenshot of a code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The code is written in a light-colored font and is as follows:

```
1  /* Recibe un numero que mutiplica por 2 y devuelve */
2  // Number -> multiplyBy2() -> Number
3  function multiplyBy2(number){
4      return number*2;
5  }
6  let resultado = multiplyBy2(4);
7  console.log(resultado);
```

*Ilustración 4. Ejemplo de código Javascript*

## HTML

**HTML (HyperText Markup Language)** es el lenguaje esencial que impulsa la World Wide Web al definir la estructura y el significado del contenido web. Gracias a HTML, los navegadores pueden interpretar y presentar de manera coherente el contenido de las páginas web, permitiendo una experiencia de navegación uniforme y accesible para los usuarios en todo el mundo.

El lenguaje de marcado de HTML se basa en el uso de etiquetas o "tags", que son elementos especiales rodeados por los símbolos "<" y ">". Estas etiquetas se utilizan para delimitar diferentes partes del contenido y proporcionan información esencial sobre la función y presentación del texto, imágenes y otros elementos multimedia en la página. Algunas etiquetas esenciales en HTML incluyen:

- `<head>`: Esta etiqueta se utiliza para contener elementos de metadatos, como títulos de página, enlaces a hojas de estilo (CSS) y scripts.
- `<body>`: Define el cuerpo principal de la página web y contiene todo el contenido visible, como texto, imágenes, enlaces y otros elementos.
- `<html>`: Esta etiqueta encierra todo el contenido HTML de la página y marca el inicio y el final del documento.
- `<img>`: Utilizado para insertar imágenes en la página web, permitiendo que los navegadores muestren gráficos y fotografías.

Además de estas etiquetas mencionadas, HTML ofrece una amplia variedad de otras etiquetas que permiten una presentación más estructurada y rica del contenido. Las etiquetas de HTML se organizan jerárquicamente para formar una estructura coherente de elementos anidados, lo que facilita la creación de páginas web bien organizadas y fáciles de mantener.

La evolución de HTML a lo largo de los años ha permitido la adopción de nuevas características y elementos avanzados, lo que ha llevado a la creación de HTML5, la versión más reciente y poderosa de este lenguaje de marcado. HTML5 ha introducido una serie de características modernas, como etiquetas de vídeo y audio, formularios mejorados, API para geolocalización y canvas, lo que ha permitido un desarrollo web más dinámico y atractivo.



```
1 <html>
2   <head>
3 </head>
4   <header> Ejemplo </header>
5   <body>
6     <p> Esto es un ejemplo </p>
7   </body>
8   <script>
9     alert("El JS se ejecuta!")
10  </script>
11 </html>
```

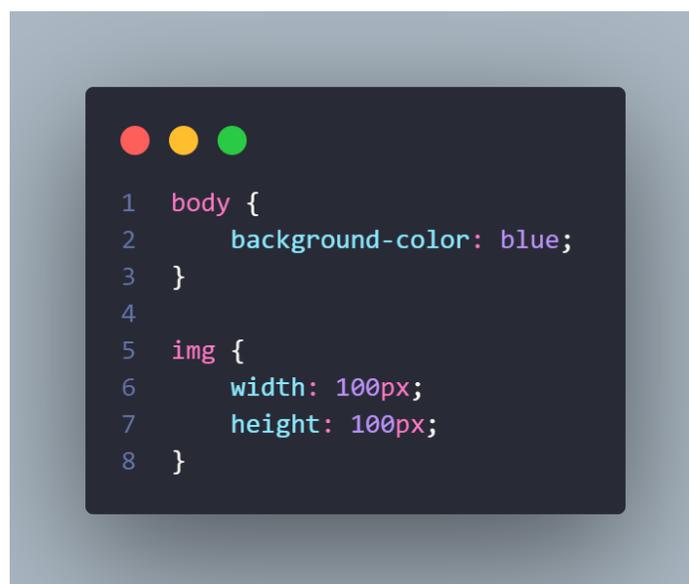
*Ilustración 5. Ejemplo de código HTML*

## CSS

CSS (Cascading Style Sheets) desempeña un papel fundamental en la apariencia y el diseño de las páginas web, al permitir la definición de aspectos visuales como colores, tipografías y la disposición de los elementos. Formando parte esencial del tríptico web junto con HTML y JavaScript, este lenguaje se erige como un pilar central en la construcción de experiencias digitales cautivadoras.

Al abrazar los rigurosos estándares establecidos por el **World Wide Web Consortium (W3C)**[23], CSS garantiza la uniformidad y la consistencia en la forma en que los elementos de las páginas se presentan en diversos navegadores y plataformas. Si bien en épocas pasadas se observó una distinción entre sus distintas versiones, como CSS1 y CSS2, el panorama actual revela una convergencia en un único paradigma conocido simplemente como CSS.

Este lenguaje trasciende su rol meramente estilístico, ya que también se erige como un instrumento versátil para adaptar las interfaces a una diversidad de dispositivos. Ofrece una amplia gama de herramientas que posibilitan discernir entre distintos tipos de dispositivos que acceden a la web, lo cual se traduce en la facultad de ajustar la presentación de manera personalizada para cada plataforma. Desde pantallas de ordenadores hasta dispositivos móviles o tabletas, CSS concede la capacidad de optimizar la visualización y la usabilidad de una página web, asegurando una experiencia de usuario óptima e intuitiva sin importar el medio de acceso.

A code editor window with a dark background and light text. At the top left, there are three colored circles: red, yellow, and green. The code is as follows:

```
1  body {  
2      background-color: blue;  
3  }  
4  
5  img {  
6      width: 100px;  
7      height: 100px;  
8  }
```

*Ilustración 6. Ejemplo de código CSS*

## 3. Desarrollo

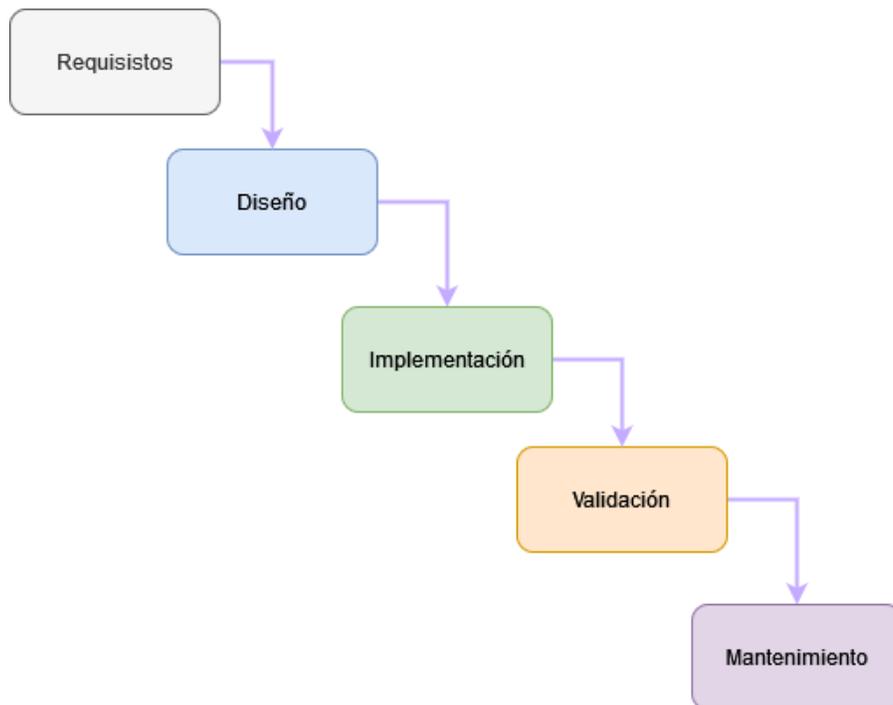
### 3.1 Metodología de trabajo

Para realizar el proyecto se ha seguido la metodología en cascada[24][25]. Esta metodología consiste en separar claramente en distintas fases secuenciales donde no se puede avanzar a la siguiente fase hasta que la anterior no haya sido completada en su totalidad. Es decir, es un procedimiento lineal. Cada fase es repetida solo una vez y los resultados de cada fase son utilizados como punto de partida de la siguiente. Si cambia alguna especificación a lo largo de la realización del proyecto, se ha de empezar de nuevo la metodología por la primera fase.

Esta metodología se ha seguido solamente hasta tener toda la extensión acabada, a partir de ese momento, a la hora de seguir implementando más páginas web o alguna mejora en la misma, no se ha utilizado ninguna metodología en específico, simplemente se investigaba la viabilidad y se implementaba el cambio.

#### Fases del proyecto

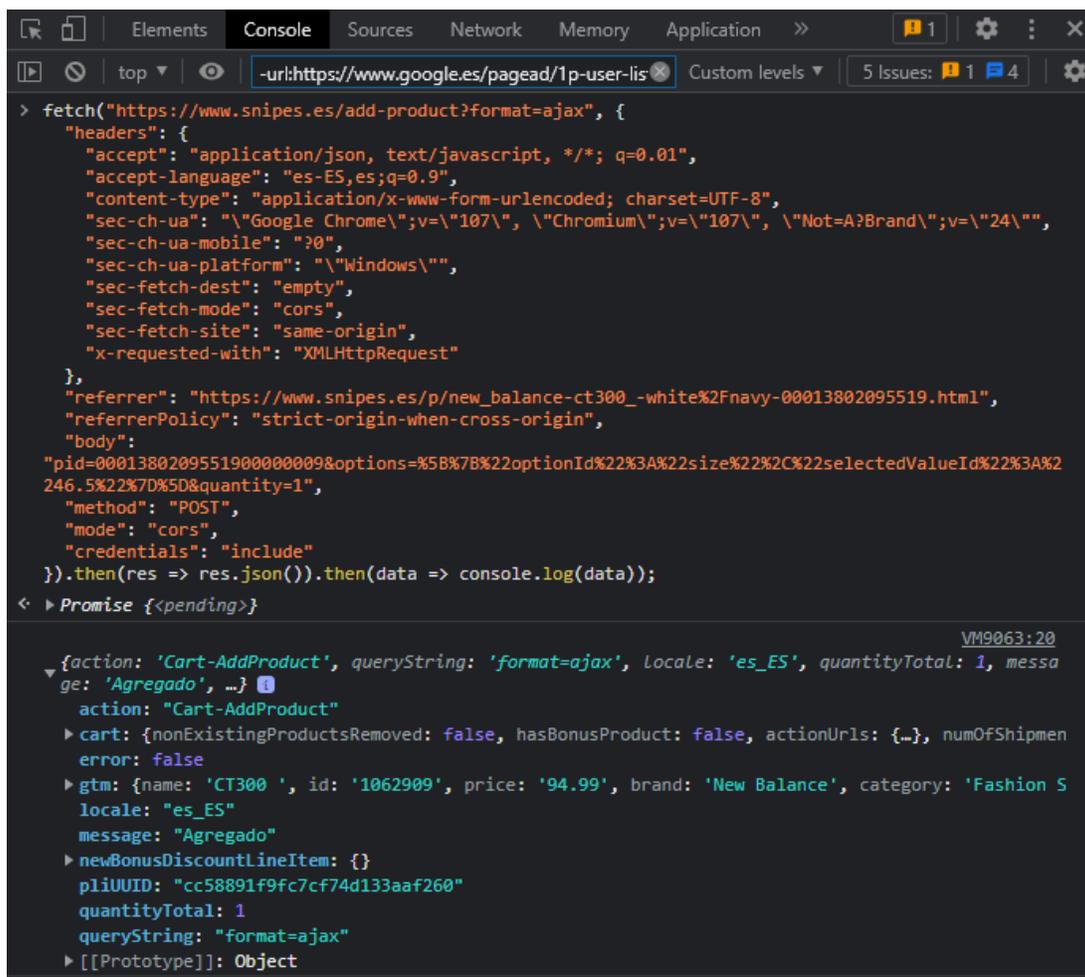
El proyecto se divide en las siguientes fases para su desarrollo: requisitos, diseño, implementación, validación y mantenimiento.



*Ilustración 7. Fases del desarrollo*

## 3.2 Requisitos

En la fase de requisitos, como su nombre indica, se han estudiado los requisitos necesarios para llevar a cabo el proyecto. La idea primaria era conseguir que cierto código JavaScript se ejecutara en el contexto de la página web para poder realizar los pasos necesarios en el proceso de compra. La primera opción a tratar fue la de utilizar en la consola del inspector del navegador, fue descartada con presteza debido a que era inviable poder inyectar el código en la consola sin vulnerar la seguridad de los navegadores. Como ejemplo voy a proporcionar unas imágenes de cómo se añadiría al carrito un producto en la página web de Snipes con el producto de New Balance CT300 (Ilustración 8).



```
> fetch("https://www.snipes.es/add-product?format=ajax", {
  "headers": {
    "accept": "application/json, text/javascript, */*; q=0.01",
    "accept-language": "es-ES,es;q=0.9",
    "content-type": "application/x-www-form-urlencoded; charset=UTF-8",
    "sec-ch-ua": "\"Google Chrome\";v=\"107\", \"Chromium\";v=\"107\", \"Not=A?Brand\";v=\"24\"",
    "sec-ch-ua-mobile": "?0",
    "sec-ch-ua-platform": "\"Windows\"",
    "sec-fetch-dest": "empty",
    "sec-fetch-mode": "cors",
    "sec-fetch-site": "same-origin",
    "x-requested-with": "XMLHttpRequest"
  },
  "referrer": "https://www.snipes.es/p/new_balance-ct300-white%2Fnavy-00013802095519.html",
  "referrerPolicy": "strict-origin-when-cross-origin",
  "body":
    "pid=0001380209551900000009&options=%5B%7B%22optionId%22%3A%22size%22%2C%22selectedValueId%22%3A%2246.5%22%7D%5D&quantity=1",
  "method": "POST",
  "mode": "cors",
  "credentials": "include"
}).then(res => res.json()).then(data => console.log(data));
< ▶ Promise {<pending>}

VM9063:20
{action: 'Cart-AddProduct', queryString: 'format=ajax', locale: 'es_ES', quantityTotal: 1, message: 'Agregado', ...}
  action: "Cart-AddProduct"
  ▶ cart: {nonExistingProductsRemoved: false, hasBonusProduct: false, actionUrls: {...}, numOfShipmen error: false
  ▶ gtm: {name: 'CT300 ', id: '1062909', price: '94.99', brand: 'New Balance', category: 'Fashion S locale: "es_ES"
  message: "Agregado"
  ▶ newBonusDiscountLineItem: {}
  pliUUID: "cc58891f9fc7cf74d133aaf260"
  quantityTotal: 1
  queryString: "format=ajax"
  ▶ [[Prototype]]: Object
```

Ilustración 8. Ejemplo de añadir al carro en Snipes.es

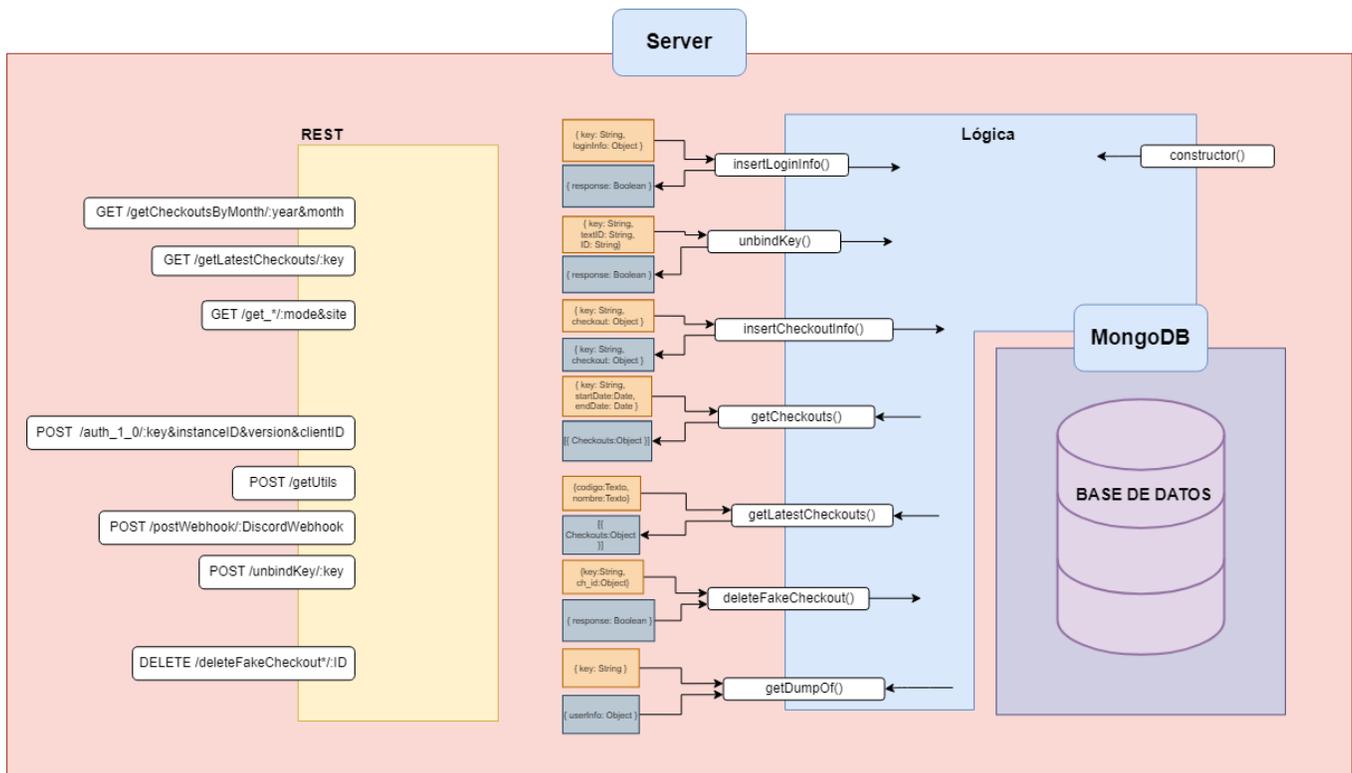
Se puede observar en la imagen, que el código JavaScript es una sencilla petición POST a “/add-product”. En la cabecera hay información relevante, como el idioma, el protocolo de petición o la plataforma de la cual se hace la misma. En el cuerpo(body) de la petición, hay información que hace referencia al producto que se va a añadir al carrito, en este caso, el PID (ID del producto), generalmente es un número, y unas opciones que describen el color o el tamaño del producto.

Como podemos observar en la respuesta, impresa mediante el `console.log()`, la respuesta, con un código 200, contiene un mensaje de éxito y nos devuelve el estado actual del carrito, en el que se puede observar el producto recién añadido.

Otra opción a tratar fue la de interactuar desde el propio PC hacia las páginas web, haciendo las peticiones pertinentes desde un script o servidor, esta opción fue descartada también debido a que convertía el proyecto en un bot y con esto venían una serie de complicaciones insalvables en la duración del proyecto.

Por lo tanto, la opción que permitía acceder al contexto de la web con mucha facilidad y además pensado con esa finalidad, fue la extensión.

### 3.3 Diseño lógico



*Ilustración 9. Diseño lógico del servidor*

En la Ilustración 9 podemos observar el diseño del servidor. Consiste de una base de datos en MongoDB, es una base de datos no relacional, es decir, los datos guardados entre ellos no están relacionados. Esta base de datos posee dos colecciones: 'users' y 'checkouts', la primera guarda todos los datos de los usuarios como los accesos y el uso de la clave, la colección 'checkouts' guarda todos los datos relacionados con las compras.

Para lidiar con el acceso a la base de datos se define un modelo, que en la ilustración se llama lógica. Para poder acceder desde la interfaz se han definido una serie de reglas REST, estas reglas proporcionan toda la información necesaria para el correcto funcionamiento de la misma.

A continuación, se presenta el diseño de la extensión, hay que tener en cuenta que la mayoría de las extensiones siguen una estructura parecida.

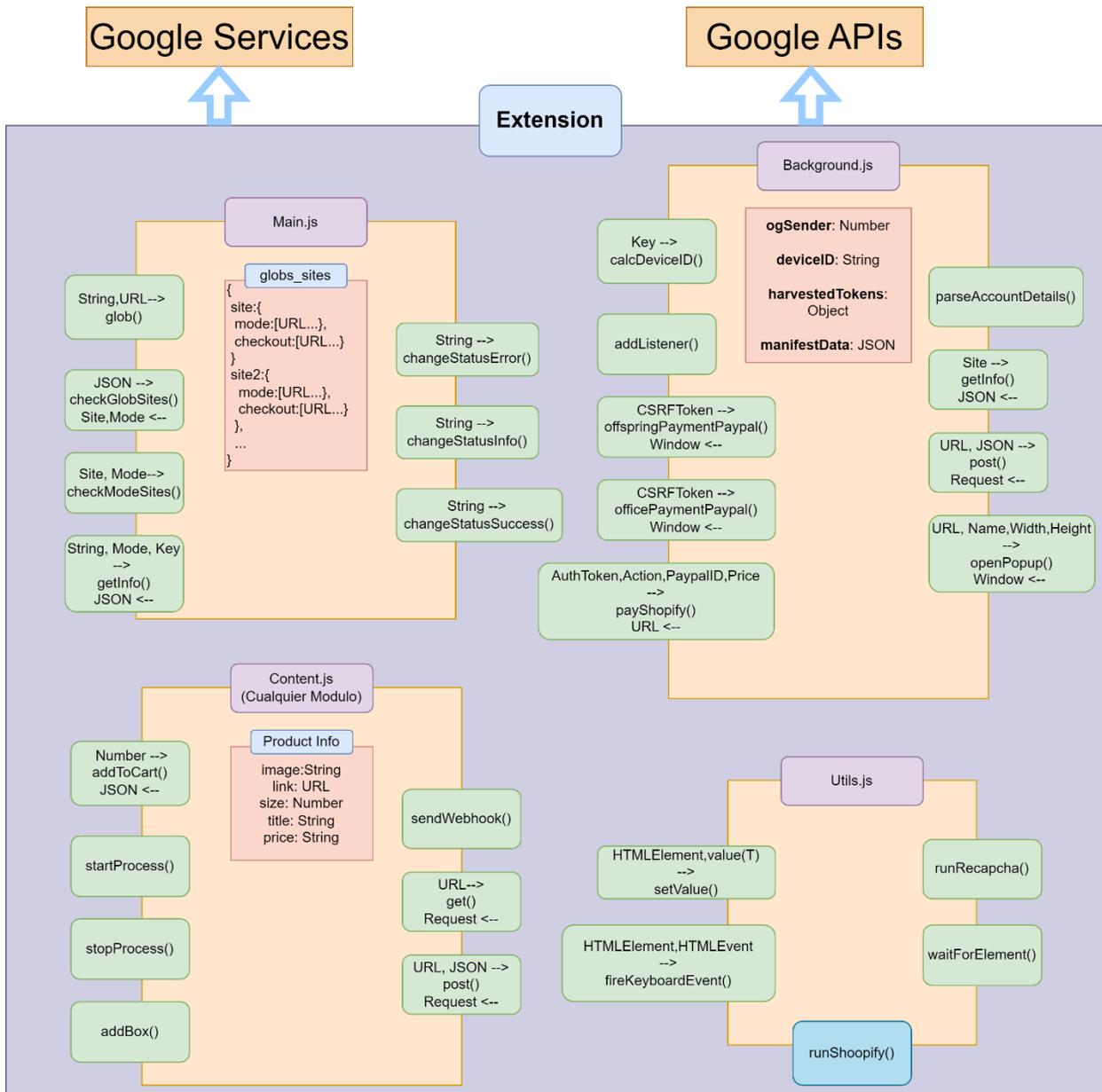


Ilustración 10. Diseño lógico de la extensión

En este caso se observa que la extensión posee una serie de archivos que realizan tareas específicas, de manera general, el background.js se ejecuta en segundo plano, main.js/tfg.js es el archivo que se encarga de lidiar con la lógica entre el segundo plano y el primero. Content.js se encarga de la lógica de la interfaz de la extensión y utils.js tiene funciones de apoyo.

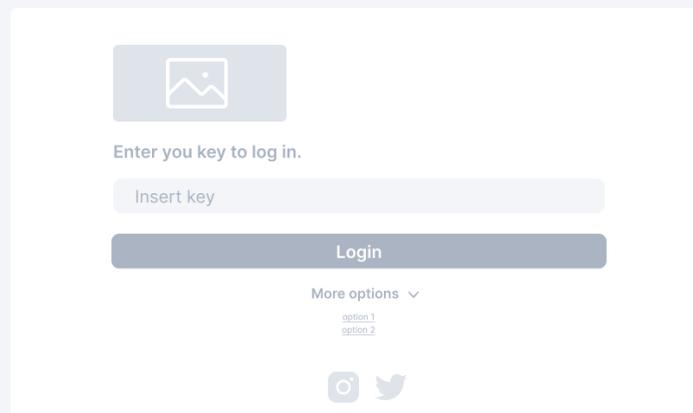
Por último, podemos observar que los servicios que nos proporciona Google no están contenidos en la extensión sino que son exteriores. Hay que hacer una mención especial para aclarar que los servicios de Google no están disponibles en su totalidad hasta que se publique la extensión.

## 3.4 Diseño visual

### Diseño del log-in

La página del log-in debe tener un campo para poder introducir la llave que te permita entrar en la extensión, así como varios botones extras para poder borrar el proxy y para borrar la información del perfil almacenada en el localStorage del navegador (Ilustración 11).

Por cómo funcionan las extensiones, es necesario añadir un botón para poder borrar la configuración del proxy antes de realizar ninguna conexión a internet. Esto es debido a que si inyectas a Chrome un proxy caducado o no funcional este no te permitirá realizar ninguna conexión ya que espera que la conexión con el proxy sea exitosa y nunca lo es.



*Ilustración 11. Diseño de la página de acceso*

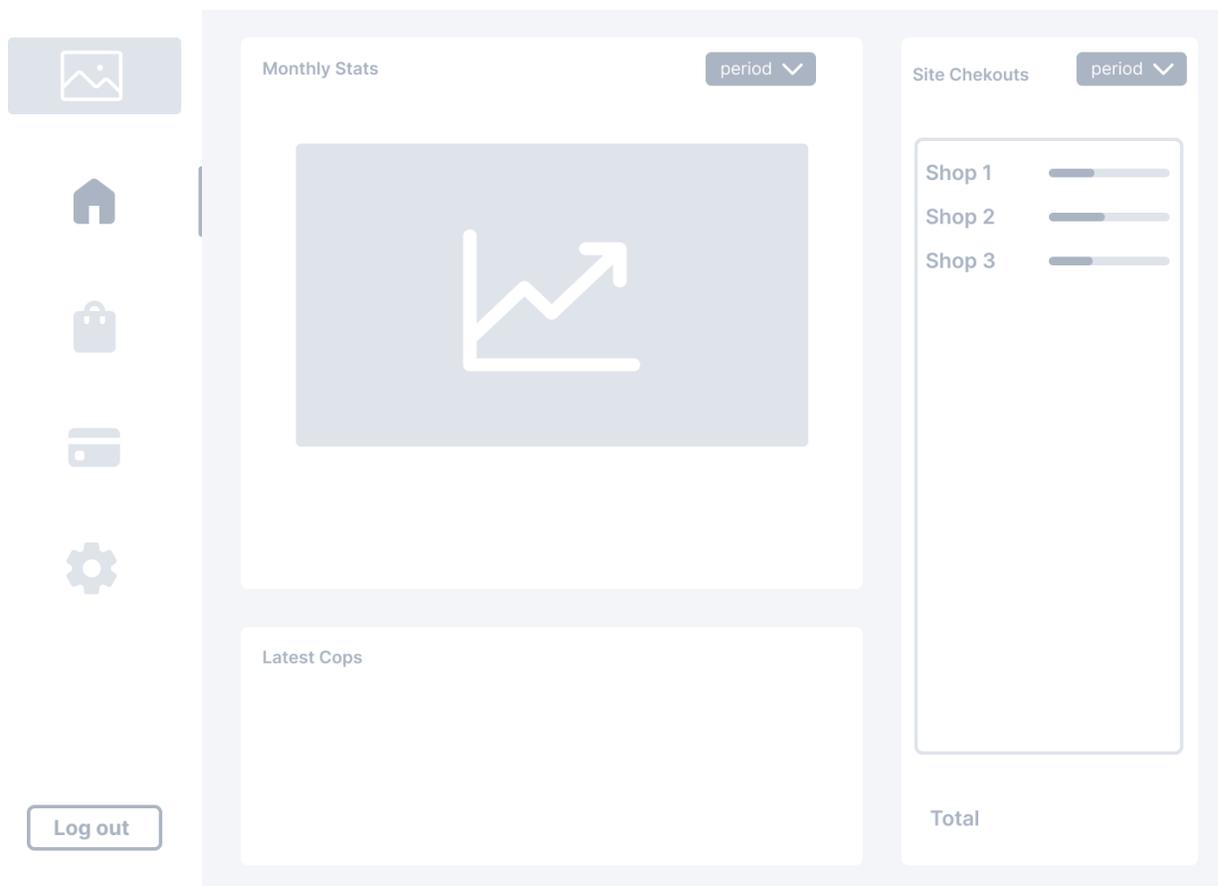
## Diseño home

La intención para la página principal es que muestre un pequeño resumen de cómo está funcionando la extensión actualmente (Ilustración 12).

A la izquierda, en un menú hamburgués permanentemente desplegado, encontramos el logo y una lista de iconos que nos permitirán visitar el resto de las páginas de la extensión. En la parte inferior hay un botón para cerrar la sesión.

En la parte central nos encontramos con un gráfico de líneas en el cual se puede observar cuantas compras estás realizando cada día pudiendo ajustar el periodo de tiempo desde el desplegable de arriba a la derecha. En la parte inferior hay una lista de tarjetas las cuales contienen una foto, nombre, precio, tienda y fecha, en la misma podemos borrar aquellos productos que queramos o simplemente observar cuales y cuando han sido.

A la derecha hay una comparación de las compras realizadas en las diferentes tiendas, de esta manera podemos ver cuáles son las más exitosas. También podemos cambiar el periodo desde el desplegable superior.



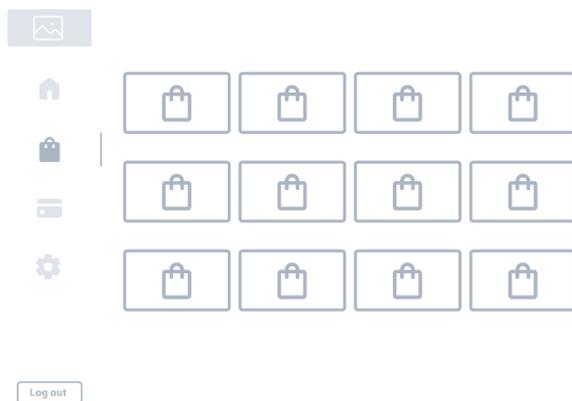
*Ilustración 12. Diseño de la página de inicio*

## Diseño de módulos

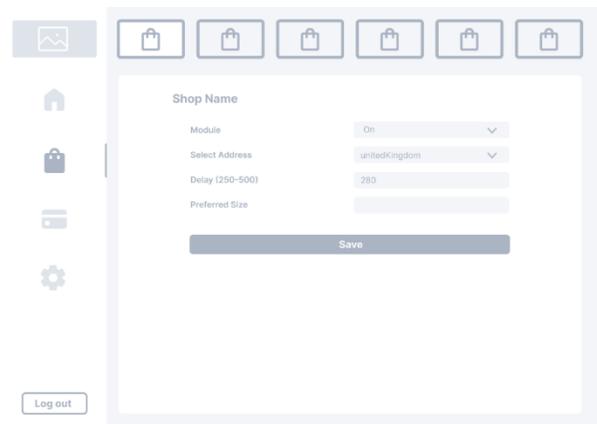
Cuando se acceda a esta pestaña (Ilustración 14) se despliega una lista de sitios web, llamados módulos, que están implementados, cada uno de ellos lleva un logo que la diferencia del resto, siendo este normalmente el de la web de compras en cuestión.

Una vez se haga clic en alguno cambiará la vista y se podrá observar en la parte superior un carrusel donde poder cambiar de módulo de manera rápida, los que están activos tienen un color característico, de manera que se pueden identificar con facilidad (Ilustración 13).

En la parte inferior una serie de campos con varios parámetros ajustables para el correcto funcionamiento de la extensión en dicha página. Estos campos en algunas ocasiones serán los mismos, pero en algunos casos habrá páginas altamente configurables.



*Ilustración 14. Diseño general de la página de módulos*



*Ilustración 13. Diseño de la página de un módulo en específico*

## Diseño de perfiles

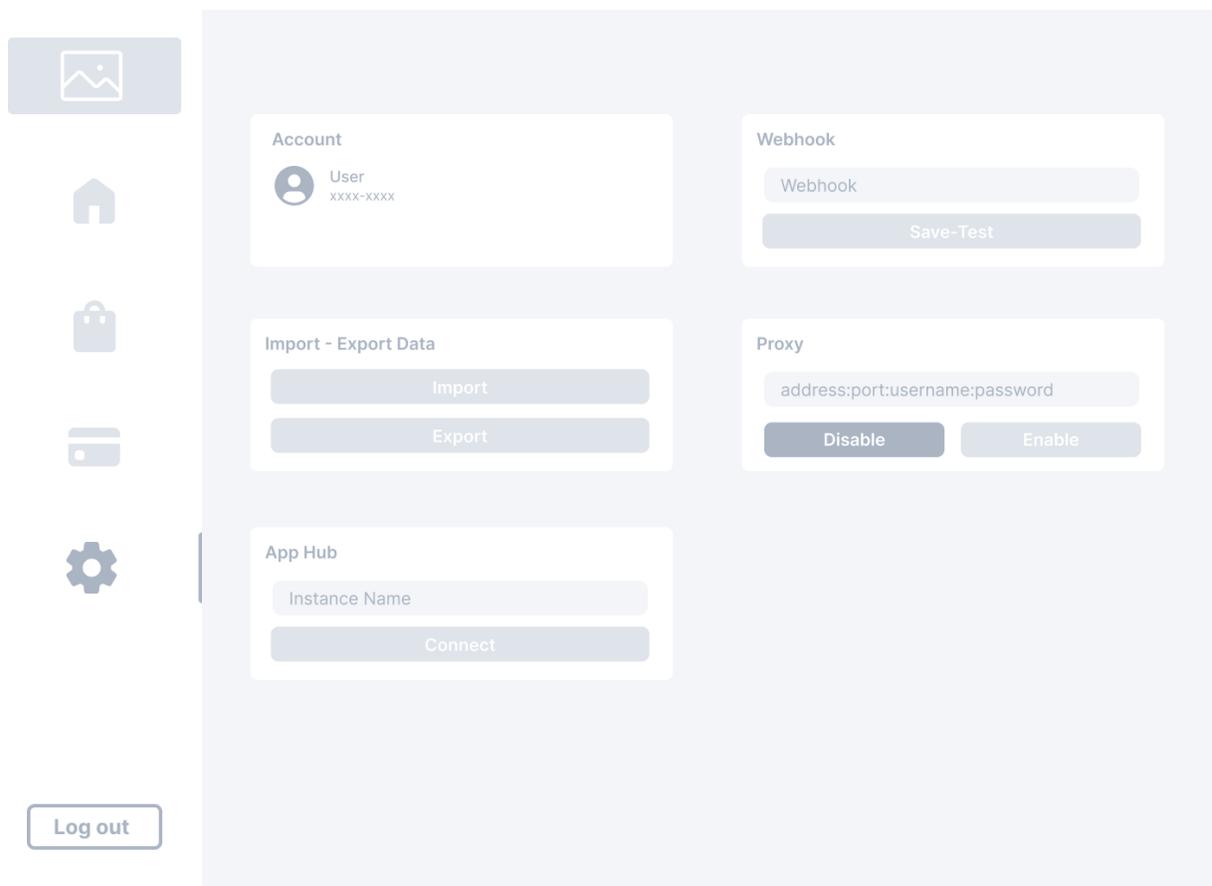
En la pestaña de perfiles podremos crear y modificar perfiles de envío. Cada perfil debe contener una dirección válida que se utilizará para el envío y facturación. El perfil viene ligado con una tarjeta de crédito. Toda esta información se guarda bajo el nombre del perfil, a la izquierda se podrá ver un resumen de todos los perfiles existentes. Hay campos que son obligatorios y no pueden dejarse en blanco (Ilustración 15).

The image shows a user interface for managing shipping profiles. On the left is a vertical sidebar with icons for home, shopping bag, credit card, and settings, along with a 'Log out' button. The main area contains a form with two main sections: 'Shipping Data' and 'Credit Card Data'. The 'Shipping Data' section has fields for First Name, Last Name, Mail, Phone Number, Street 1, House/apt number, Street 2, City, ZIP, State, and Country. The 'Credit Card Data' section has fields for Name, Number, and CVV. Below these is a 'Profile Name' field. A 'Save' button is at the bottom right of the form. To the right of the form is a 'Profiles' section with two placeholder icons for credit cards.

*Ilustración 15. Diseño de la página de perfiles de envío/pago*

## Diseño de la configuración

En la página de configuración (Ilustración 16) se podrá ver información básica en relación al perfil del usuario. Se tiene que poder guardar el 'endpoint' del 'webhook'. También tenemos que poder exportar el perfil (a un documento .txt o .json) para su posterior importación.



*Ilustración 16. Diseño de la página de configuración*

## 3.5 Implementación

En este apartado se van a comentar los resultados de la implementación, sobre todo lo relacionado a la lógica y al código implementado. Se explican solo aquellos scripts que merecen ser mencionados ya sea porque su función es interesante o única.

### Background.js

Este archivo define un script que se ejecuta en segundo plano como parte de la extensión de Chrome. Su objetivo principal es realizar tareas en segundo plano, como escuchar eventos, comunicarse con otras partes de la extensión y mantener un estado persistente. Un background script no tiene acceso a la misma API de Service Worker y no puede interceptar solicitudes de red o cachear. Su propósito principal es habilitar funcionalidades en segundo plano dentro de la extensión, como la administración de notificaciones, la actualización de datos y la interacción con APIs externas.

En este caso más específico se encarga de las siguientes funciones:

- Enviar mensajes a servidores externos: webhooks
- Manejar errores
- Manejar métodos de pago específicos
- Manejar la lógica del pop-up
- Calcular la ID del navegador

### CORS

Uno de los problemas a la hora de realizar los flujos de compra es que los diferentes “endpoints” se encuentran en diferentes dominios. Como nuestros content scripts se ejecutan en un dominio específico la directiva de seguridad de CORS[26] nos impide poder realizar estas llamadas. CORS es una política de seguridad en los navegadores que permite compartir recursos entre diferentes dominios, brindando un mecanismo controlado para solicitudes de origen cruzado. Es decir, permite al desarrollador controlar desde donde (dominio) se puede realizar una petición. Por ejemplo, no puedes acceder a Wikipedia si estás en YouTube, ya que no se permite el intercambio cruzado entre ellas.

En nuestro caso, el background.js se ejecuta en todas las webs que están declaradas en el manifest.json de la extensión, por lo tanto, técnicamente cuando pretendes realizar una petición desde una **Web A** que no tiene contemplado en CORS a una **Web B**, si añades la **Web B** al manifest.json, el background se ejecuta en la **Web B** y por lo tanto no hay origen cruzado y puedes realizar la petición. Un ejemplo de función sería `offspringPaypalPayment()`.

Las dos webs, **A** y **B**, están declaradas en el manifest.json y por lo tanto el background.js se ejecuta en las dos. Cuando sea necesario realizar una petición que necesita información de la **Web B** desde la **Web A**, puedes utilizar el background.js para realizar dicha petición y enviar la información a la **Web A** una vez esta petición se haya completado.

## Generador de tokens: ReCaptcha

Otra de las funcionalidades más interesantes implementadas es la recolección de tokens de ReCaptcha.

La historia de **reCAPTCHA**[27] se remonta a principios de la década de 2000, cuando el equipo de investigación de Luis von Ahn en la Universidad Carnegie Mellon comenzó a explorar formas de aprovechar el tiempo y la energía invertidos por los usuarios en actividades en línea para resolver problemas útiles. Esta investigación culminó en la creación de CAPTCHAs, acrónimo de "Completely Automated Public Turing test to tell Computers and Humans Apart", una tecnología que desafía a los usuarios a demostrar que son humanos, no bots, mediante la resolución de tareas que son difíciles para las máquinas.

La idea detrás fue transformar el proceso de digitalización de libros y documentos en una tarea colaborativa. Muchos CAPTCHAs presentaban palabras distorsionadas generadas a partir de escaneos de libros antiguos. Al implementar dos palabras en lugar de una, una de las cuales era conocida y verificada mientras que la otra no, se permitía que los usuarios contribuyeran a la transcripción de libros mientras demostraban su humanidad. Esto permitió digitalizar miles de libros mientras se protegían los sitios web de accesos automatizados.

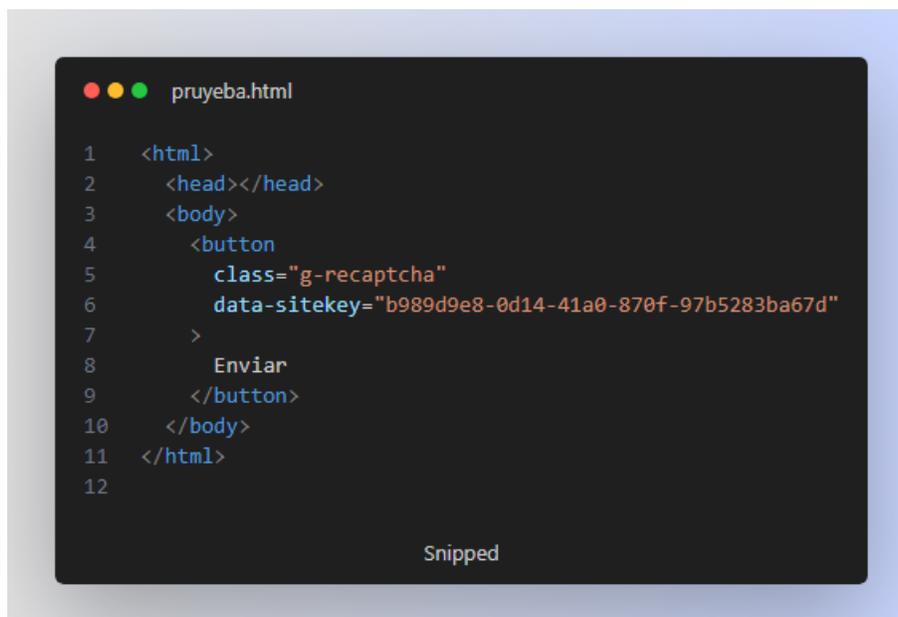
Con el tiempo, evolucionó y se adaptó para enfrentar desafíos más complejos. En reCAPTCHA v2, se introdujo la característica "No soy un robot", donde los usuarios debían marcar una casilla para demostrar que eran humanos. Además, la versión v2 introdujo la tarea de "Arrastrar y soltar" o seleccionar imágenes relacionadas con una descripción dada. Esto ayudó a mejorar la experiencia del usuario, reduciendo la necesidad de resolver CAPTCHAs difíciles y distorsionados.

Hoy en día, reCAPTCHA v3 es la versión más reciente y avanzada. A diferencia de las versiones anteriores, opera en segundo plano sin intervención del usuario. Utiliza técnicas avanzadas de análisis de comportamiento y riesgo para evaluar la probabilidad de que un usuario sea humano o un bot. Basado en esta puntuación, el sitio web puede tomar decisiones sobre cómo tratar al usuario, ya sea permitiendo el acceso sin interrupciones, mostrando un desafío como los mencionados si hay dudas o bloqueando completamente el acceso si se considera que es un bot malicioso.

Hasta la versión reCAPTCHA v2, el funcionamiento era bastante sencillo y directo: se te presenta un desafío CAPTCHA siempre que vas a realizar una acción o tarea en una web que requiera que seas humano. Como ya hemos mencionado con anterioridad, uno de los paradigmas de la programación web es que el código fuente de la web puede ser observado y estudiado por el usuario que la visita, basta con abrir el inspector del navegador para poder leerlo.

Si se observa el código fuente de reCAPTCHA o se estudia su funcionamiento se puede llegar a deducir que el sistema es sencillo: se utiliza un ID para identificar la web desde la cual estás realizando la petición para realizar el desafío, debe haber en la web un elemento con una clase CSS específica en la cual se añadirá el desafío y debe de haber un 'callback' para tratar el resultado de este desafío cuando se resuelva.

En la mayoría de los sitios web el data-sitekey está expuesto al público y puedes encontrarlo en el código fuente tal que así:



```
prueba.html
1 <html>
2 <head></head>
3 <body>
4 <button
5   class="g-recaptcha"
6   data-sitekey="b989d9e8-0d14-41a0-870f-97b5283ba67d"
7 >
8   Enviar
9 </button>
10 </body>
11 </html>
12
```

Snipped

*Ilustración 17. Ejemplo de implementación de reCaptcha*

Esta propiedad data-sitekey es extremadamente importante ya que es la forma por la cual el sistema reCAPTCHA reconoce desde qué sitio web se está realizando la petición. Si sabemos esta propiedad y somos capaces de ejecutar el código fuente de reCAPTCHA en un dominio podemos realizar los desafíos y en consecuencia obtener los tokens.

El código fuente[28] de reCAPTCHA se puede obtener ya que es público, pero no puede ser modificado porque se presenta ofuscado.

Como también se ha explicado con anterioridad, se puede ejecutar la extensión en cualquier dominio que se considere pertinente.



```
{
  "matches": [
    "*://eu.kith.com/pages/inventada/?webCaptcha"
  ],
  "js": [
    "src/scripts/utils/api.js",
    "src/scripts/utils/ec.js"
  ]
},
```

*Ilustración 18. Ejemplo de manifest.json donde se ejecutan dos scripts en una web específica*

En la Ilustración 18 podemos observar que vamos a ejecutar los scripts `api.js` y `ec.js` en la web `'eu.bith.com/pages/inventada/...'`. Como es obvio esa página no existe dentro del dominio de `kith.com`, pero la extensión va a ejecutar los scripts igualmente. El archivo `api.js` es el código fuente de reCAPTCHA y `ec.js` es un script que se encarga de añadir HTML parecido a la Figura 18 en la cual exista un `data-sitekey` y un `callback`.

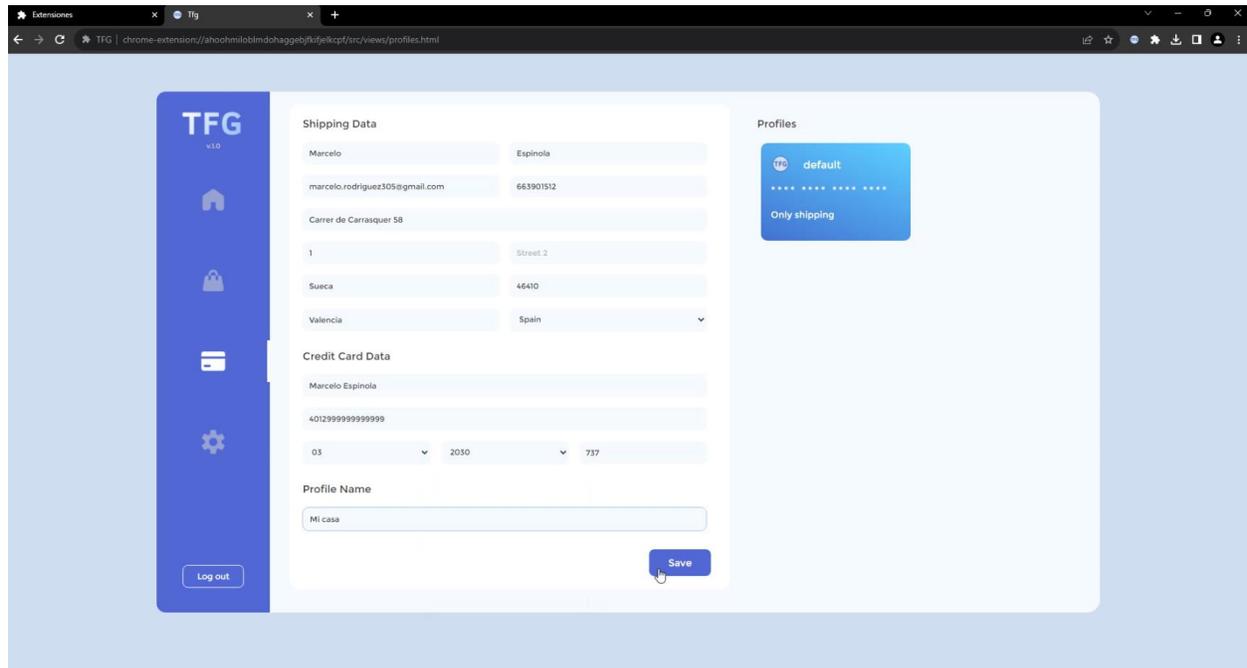
Y después de esto es cuando podemos engañar al sistema reCAPTCHA para que nos proporcione un desafío cuando queramos. Cuando se resuelve este desafío el sistema te proporciona un TOKEN que es válido durante dos minutos, una vez es enviado, se procesa y deja de serlo. El engaño consiste en que no se envía el token, como es válido durante dos minutos, se debe de realizar antes de que se produzca un lanzamiento de algún producto muy exclusivo y mientras el grueso de la población realiza el desafío, el usuario ya tiene el desafío completado con anterioridad y puede enviar el TOKEN que se le ha proporcionado para evitar tener que perder el tiempo cuando este apremia.

### 3.6 Casos de uso

En este apartado se pretende demostrar el funcionamiento de la extensión desarrollada dando como ejemplos diferentes casos de uso.

#### Crear un perfil

Para empezar, mostraremos como se configuraría un perfil nuevo en la sección de perfiles de la extensión.



The screenshot shows a web browser window with the URL `chrome-extension://ahoohtmlblmdohaggebjk5fjklcpf/stz/views/profiles.html`. The page features a blue sidebar on the left with the TFG logo and version `v1.0`, and icons for home, lock, card, and settings. The main content area is white and contains a form for creating a profile. The form is divided into three sections: 'Shipping Data', 'Credit Card Data', and 'Profile Name'. The 'Shipping Data' section has fields for 'Marcelo', 'Espinola', 'marcelo.rodriguez305@gmail.com', '663901512', 'Carrer de Carrasquet 58', '1', 'Street 2', 'Sueca', '46410', 'Valencia', and 'Spain'. The 'Credit Card Data' section has fields for 'Marcelo Espinola', '4012999999999999', '03', '2030', and '737'. The 'Profile Name' section has a text input field containing 'Mi casa'. A 'Save' button is located at the bottom right of the form. A 'Log out' button is located in the sidebar. On the right side of the form, there is a 'Profiles' section with a blue button labeled 'default' and a list of dots below it, with 'Only shipping' selected.

*Ilustración 19. Página de creación de perfil de envío*

En la Ilustración 19 podemos observar cómo se rellenaría la información en relación a la dirección de envío y su tarjeta de crédito. Existen mensajes de error para cuando un usuario no rellena la información correctamente.



En la Ilustración 20 podemos observar como la extensión muestra la información de la cuenta con la se ha accedido, en la Ilustración 21 también se enseña el resultado de exportar un perfil, con todos los datos visibles excepto la tarjeta de crédito, que por seguridad esta encriptada utilizando AES y una clave privada.

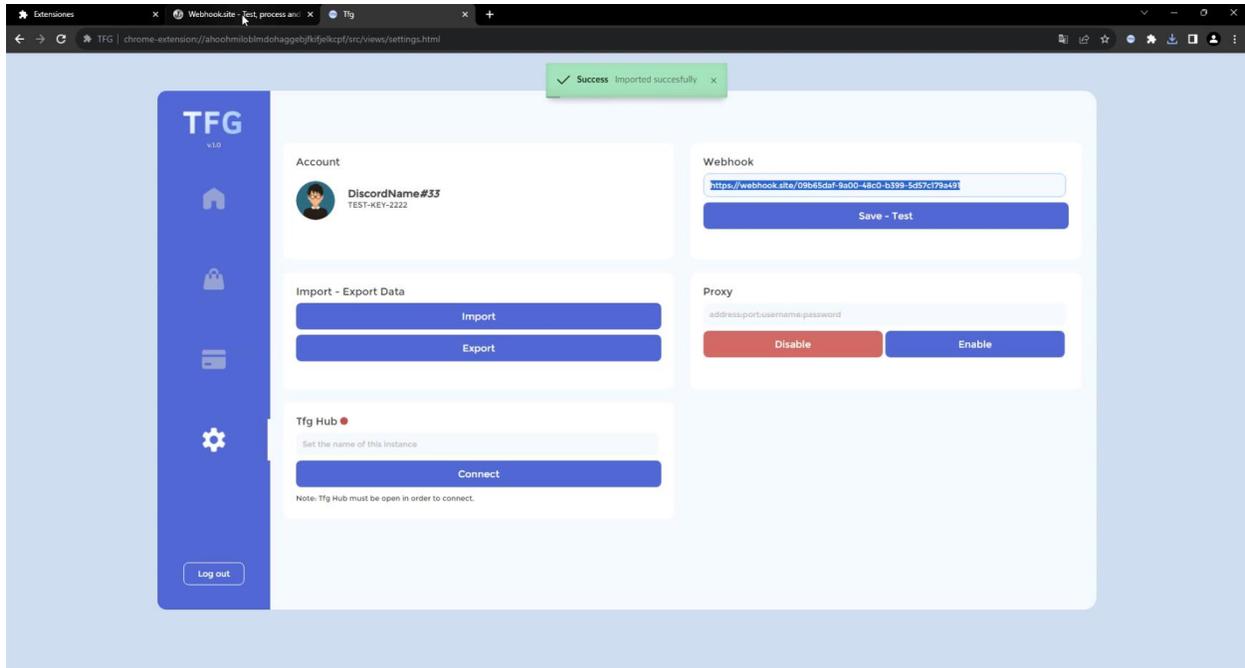


Ilustración 22. Se añade un webhook en la sección correspondiente.

Por último, se enseña el funcionamiento del webhook, se utiliza la web <https://webhook.site/> para generar momentáneamente un webhook que una vez añadimos y guardamos en la extensión, observamos que se envía uno de test en la Ilustración 23.

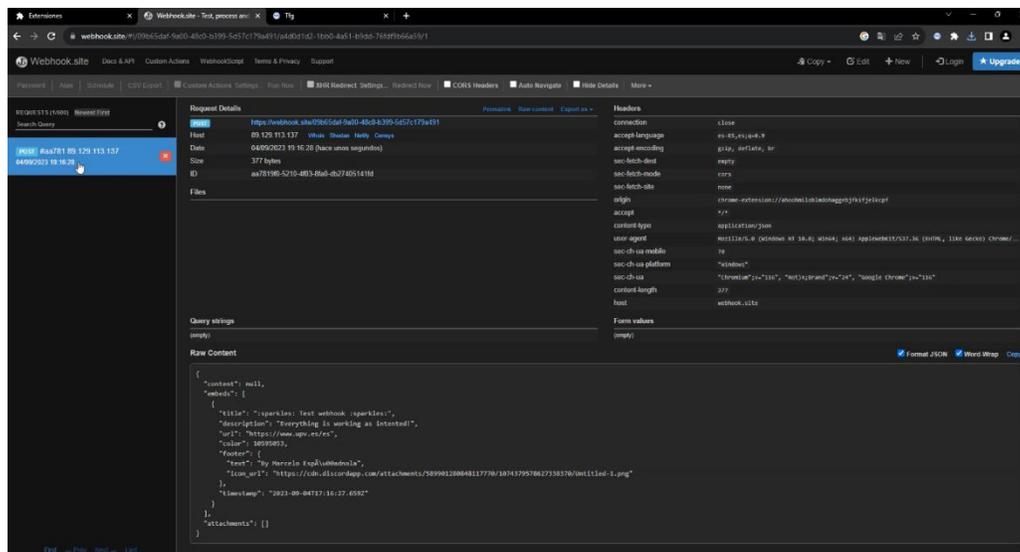


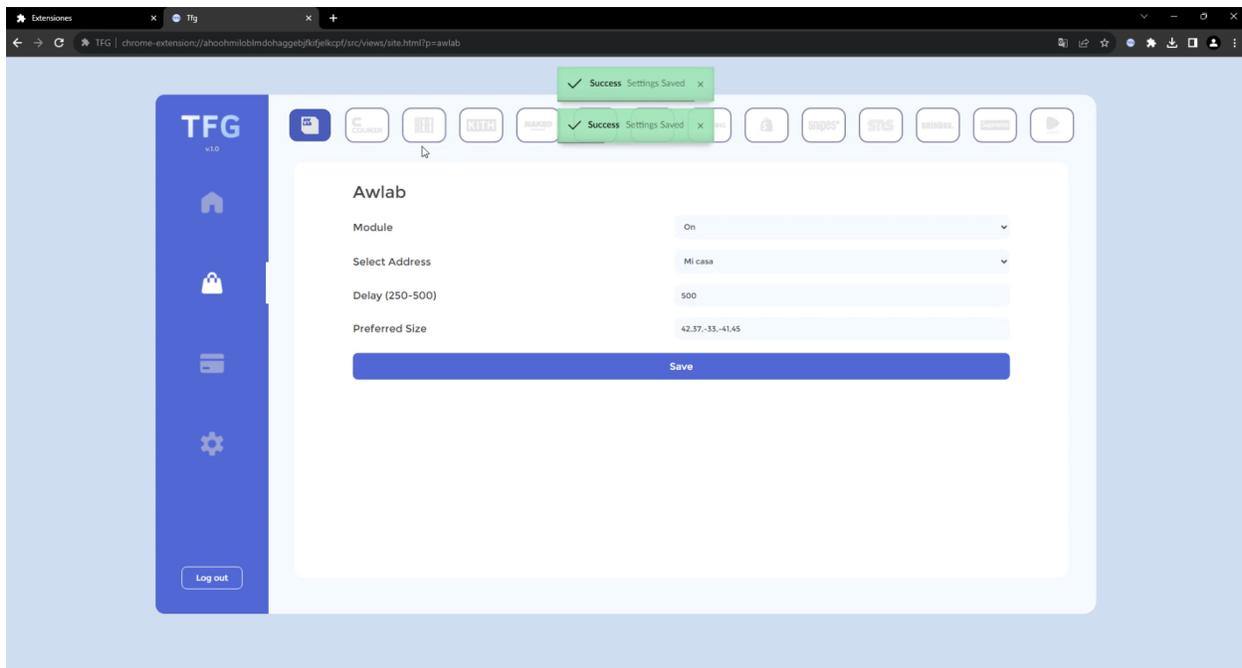
Ilustración 23. webhook.site enseñando el funcionamiento del webhook.

## Módulos

En la página de los módulos se pueden observar todos los módulos implementados en la extensión.

Sitio web	Estado
Zalando	Acabado
AW-LAB / HERE	Acabado
Kith	Acabado
Naked	En desarrollo
Luisaviaroma	Acabado
Office / Offspring	Acabado
Snipes /Solebox	En desarrollo
SNS	Acabado
Supreme	Acabado
Zalando	Acabado
Courir	En desarrollo
Shopify	Acabado*

\*Hay infinidad de páginas web que implementan Shopify “estándar” para la venta de productos, por lo tanto, no se puede afirmar que funcione en todas, debido a que tal vez algunas webs utilizan su propia implementación.

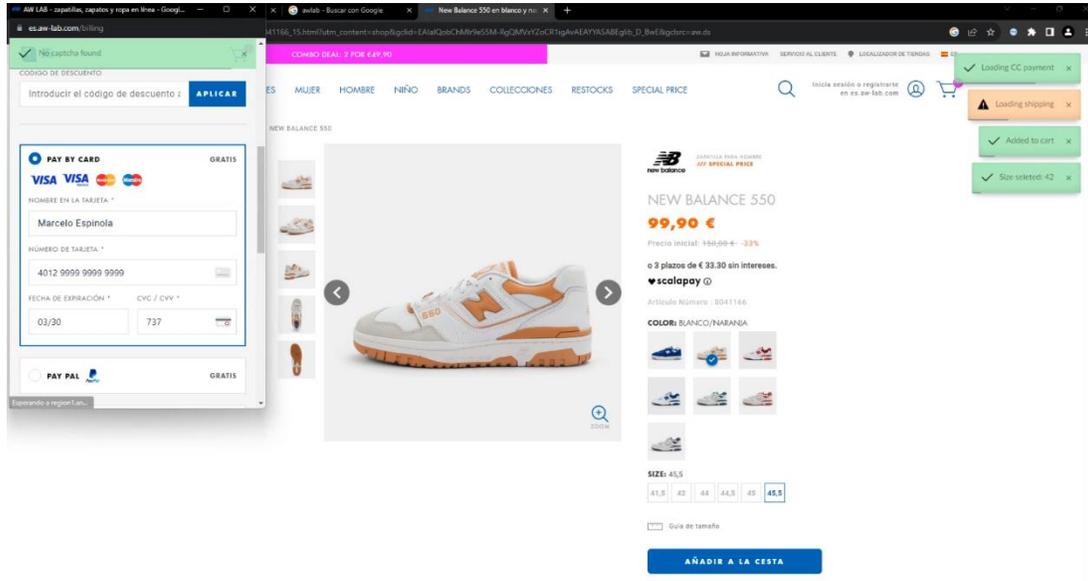


*Ilustración 24. Ejemplo de relleno de perfil*

En la Ilustración 24 se puede observar cómo se rellenaría de manera general todos los módulos. Aunque la mayoría de los módulos tiene la misma información, hay algunos para los que hay que rellenar información adicional. Entre otras cosas, puede seleccionar si encender o apagar el módulo, qué dirección va a ser la de reparto, el retraso entre las peticiones HTTP y por último la talla. En la talla todas las tallas han de ir separadas por comas y las tallas negativas se descartan. Si no existe una talla se seleccionará aleatoriamente.

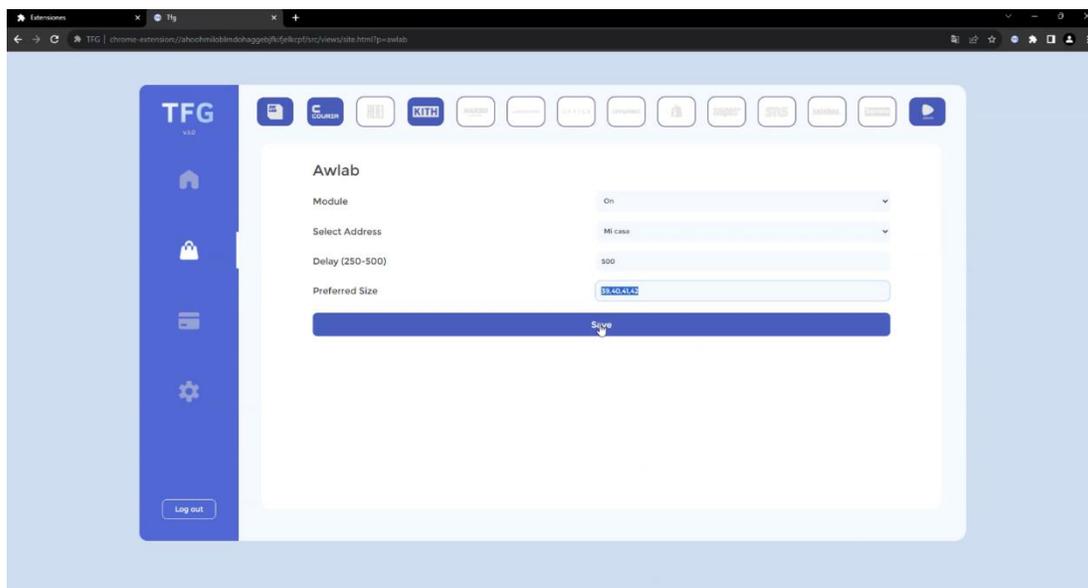
## Aw-lab/Here

En este caso específico, tanto Aw-lab como Here funcionan de la misma manera, simplemente cambia el dominio de las llamadas HTTP.



*Ilustración 25. El módulo de AW-LAB en acción*

En la Ilustración 25 podemos ver un flujo de compra de unas zapatillas. Como se puede observar, hay una serie de mensajes que te indican cómo está avanzando el proceso, la mayoría verdes porque todo está saliendo bien. Se selecciona la talla 42 porque es la única que está disponible de todas las encontradas en la lista de tallas.



*Ilustración 26. Configuración de AW-LAB*

En la Ilustración 26 podemos ver la lista de tallas que hemos introducido.

## Office / Offspring

En este caso específico, tanto Office como Offspring funcionan de la misma manera, simplemente cambia el dominio de las llamadas HTTP.

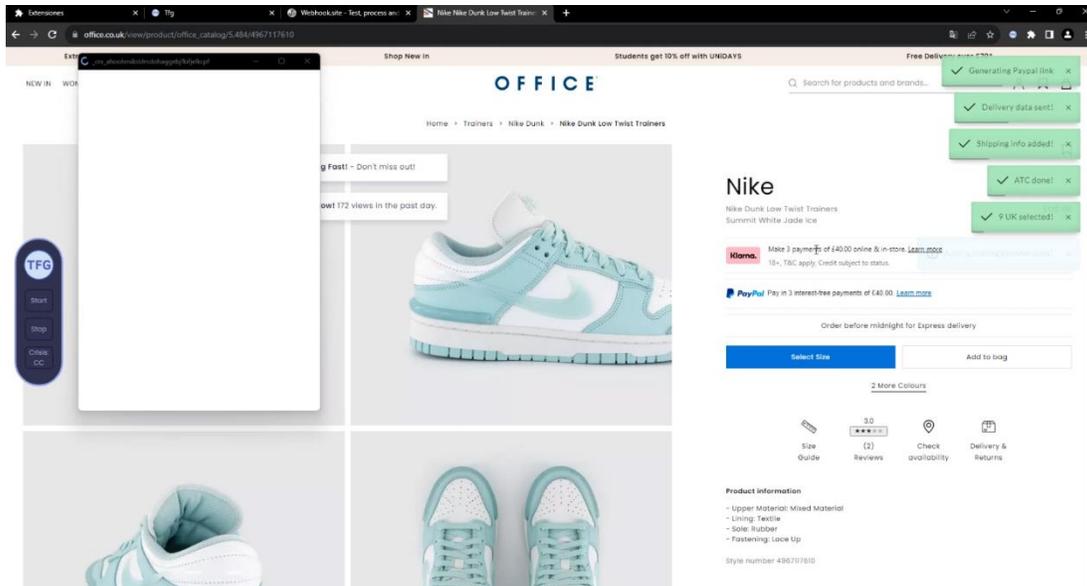


Ilustración 27. El módulo de Office/Offspring en funcionamiento

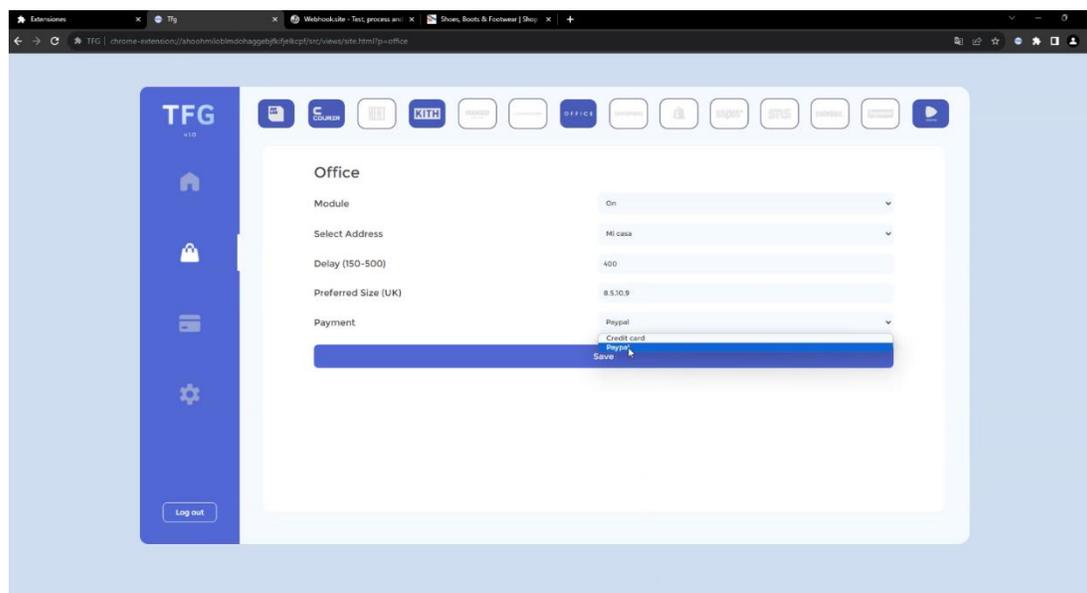
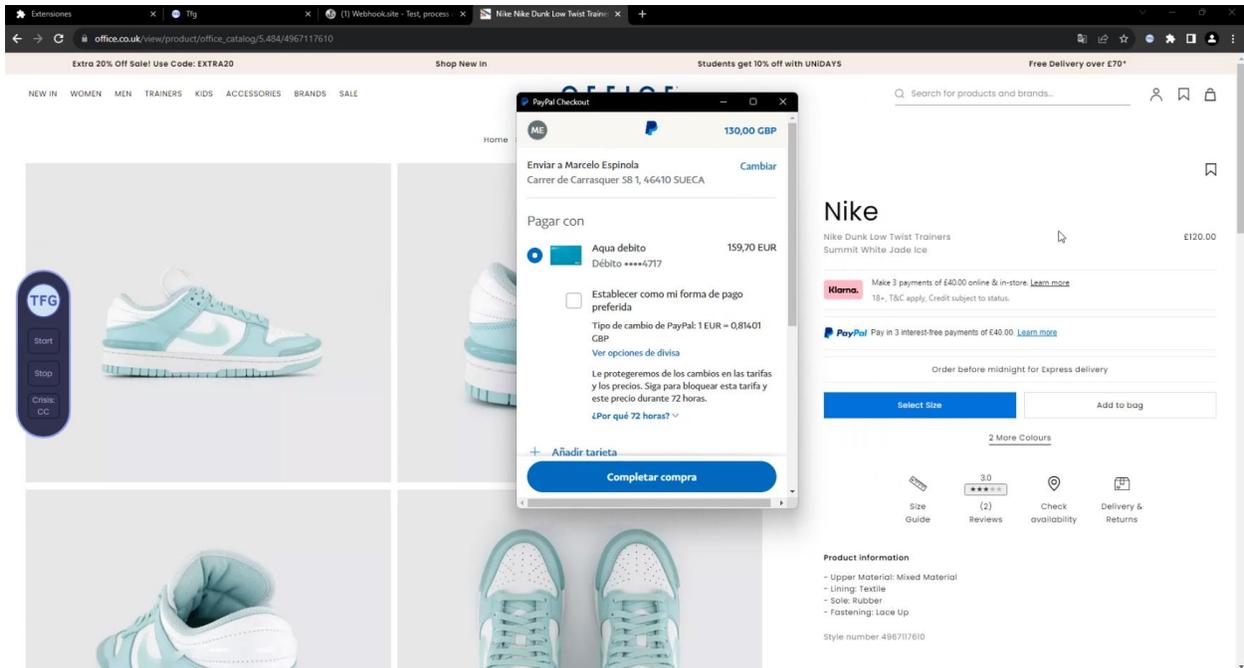


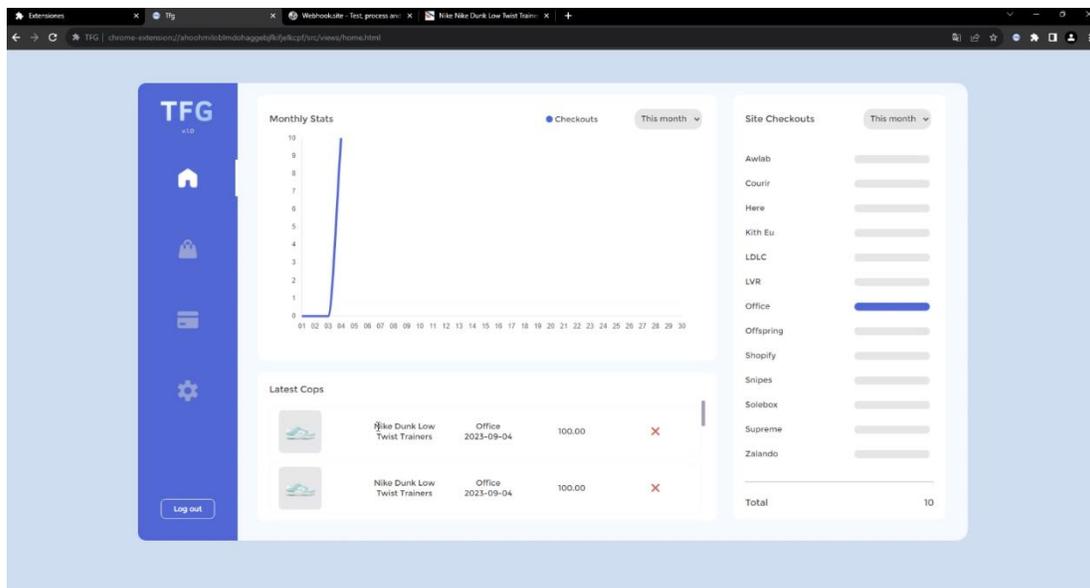
Ilustración 28. Configuración del módulo de Office

En la Ilustración 27 se puede observar un flujo básico de office.co.uk, es la primera web que se introduce que tiene pago por PayPal), como se observa en la Ilustración 28, el método de pago es seleccionable en la página del módulo. Una vez configurado nos dirigimos a un producto aleatorio y observamos que la compra es exitosa y se nos proporciona el hipervínculo para realizar el pago de PayPal.

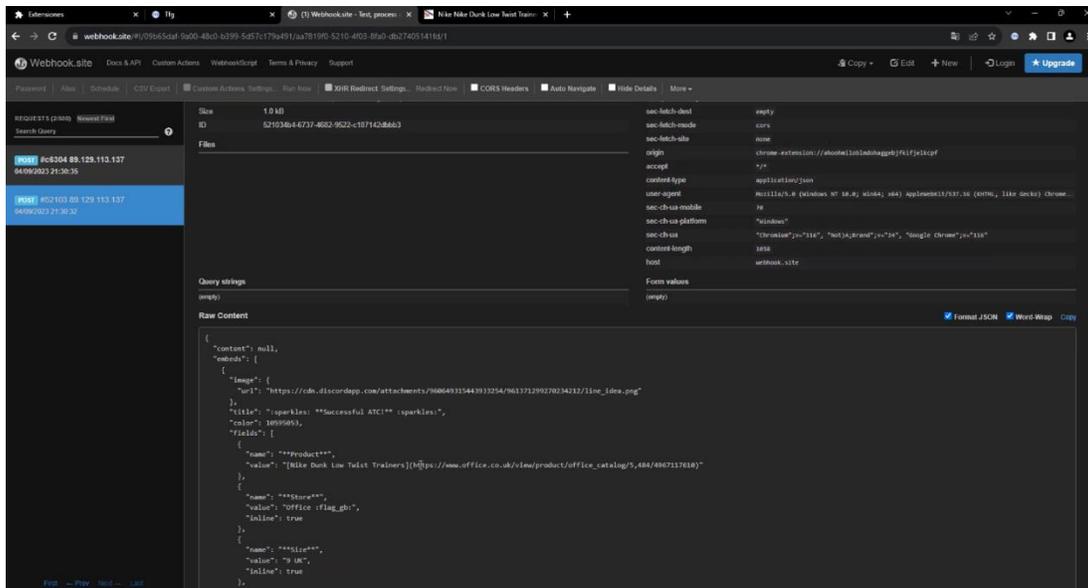


*Ilustración 29. Hipervínculo de PayPal proporcionado por el módulo*

Una vez se genera el hipervínculo, la compra se da por finalizada y se envían los webhooks, cosa que se comprueba en la web [webhook.site](http://webhook.site), y se registra en la base de datos. Como se observa en las siguientes ilustraciones.



*Ilustración 30. Actualización del frontal debido a la compra*



*Ilustración 31. Webhook enviado por la compra realizada*

## Shopify

Es la web que más ha costado de realizar por su gran número de implementaciones diferentes. En este caso mostraremos solo una de ellas, visualmente todas son iguales, pero cambian en base a los datos que piden en las APIs.

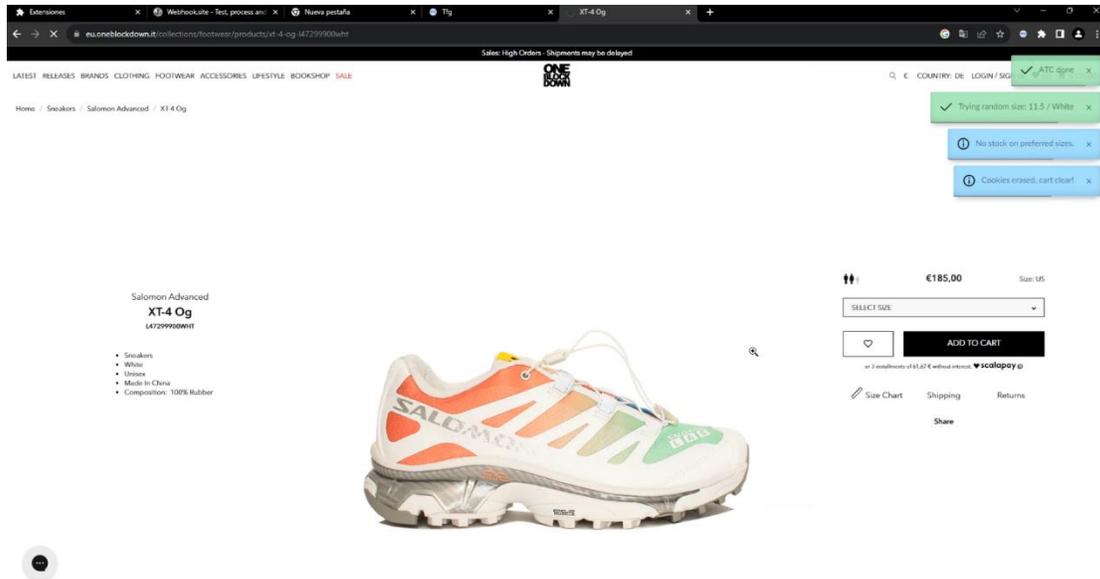


Ilustración 32. El módulo de Shopify añadiendo al carrito

En la Ilustración 32 podemos observar cómo se realiza una compra en una página web que tiene implementado Shopify para tratar sus productos. No hay tallas de las que se encuentran en la configuración y por lo tanto selecciona una aleatoriamente.

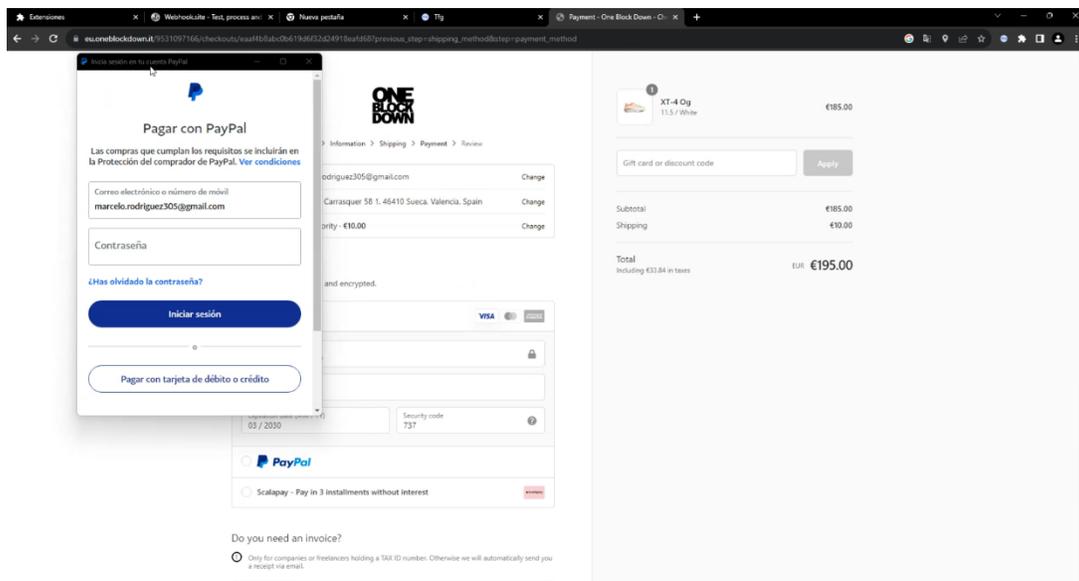


Ilustración 33. El módulo de Shopify nos proporciona un hipervínculo de PayPal

## Rendimiento general

Se ha podido observar que los métodos que se han utilizado para poder realizar las compras hacen que la misma sea mucho más rápida que los métodos manuales. En la siguiente tabla se detallan datos aproximados relacionados con los tiempos de compra tanto manuales como utilizando la extensión.

<b>Web</b>	<b>Manual</b>	<b>Extension</b>
Zalando	2 minutos	10 segundos
Aw-lab / HERE	1 minuto 10 segundos	5-7 segundos
Snipes / Solebox	30 segundos	7 segundos
Shopify*	1 minuto 10 segundos	10 segundos
Supreme	50 segundos	20 segundos
Luisaviaroma	30 segundos	7 segundos
Naked	50 segundos	10 segundos
SNS	-	-
Office / Offspring	1 minuto 10 segundos	10 segundos
Courir	-	-

\*Es una media de los métodos que estan contemplados.

Teniendo en cuenta los datos proporcionados entendemos que utilizar la extensión proporciona una clara ventaja a intentar realizar la compra manualmente.

## 4. Conclusiones y trabajo futuro

En conclusión, se considera que todos los objetivos planteados para el TFG se han cumplido de manera exitosa, siendo en la mayoría de los casos un resultado incluso mejor que el planteado inicialmente.

La extensión está preparada para ser comercializada, como hemos podido observar el éxito conseguido con ella es significativo. La extensión da una solución al problema de presteza necesaria para poder realizar alguna compra cuando se trata de algún producto exclusivo.

La realización de este trabajo ha ayudado a profundizar en los conocimientos que se han impartido durante la duración de este grado. Ha sido muy interesante poder observar el comportamiento de las diferentes páginas web y poder interactuar con ellas mediante código JS. Por otra parte, tener que desarrollar una interfaz ha ayudado a asentar todo el conocimiento en lo que se refiere a ‘frontend’ sobre todo debido a que en su gran mayoría se ha programado dinámicamente.

Ha habido una parte importante del desarrollo dedicado al ‘backend’ de la extensión, esto no es necesario y, como se comentará en las propuestas de mejora, debería de cambiar, pero ha sido interesante poder configurar un servidor para proporcionar archivos JS a la extensión para su funcionamiento y su actualización automática.

Como trabajo futuro se propone el cambio de versión del Manifest de la extensión a **V3**. Se ha desarrollado en V2 debido a que la documentación existente para la misma era infinitamente más abundante que la V3, por lo menos al principio del desarrollo. V2 va a ser abandonado durante el 2023 y con ello se desactivarán muchos servicios y vulnerabilidades que hemos utilizado para que se pudieran realizar los flujos de compra. Uno de los cambios más importantes es la capacidad de evaluar código JS, en la V3 no se podrá evaluar código JS. En este caso evaluábamos el código JS que se envía desde el servidor, lo que nos permitía actualizar este código sin necesidad de actualizar la extensión.

Como es obvio, siempre se puede **mejorar** cualquier modulo en lo que a **velocidad** se refiere, aunque en la Ilustración 25 podemos ver que la mejora es sustancial, por lo que se puede afirmar que todos los módulos son mejorables. Incluso en el mejor de los casos, que es en el que el código JS tiene la mejor complejidad posible, se pueden utilizar otras formas de realizar la compra, ya sea mediante abusos en la seguridad o diferentes peticiones (muchas veces escondidas) de la web.

Por último, se podría prescindir de la extensión como tal, y convertir lo ya hecho en un bot, con todas las mejoras que eso conllevaría. No es muy realista pensar que se podría hacer manteniendo ni siquiera el esqueleto del código que se ha desarrollado, pero sería el siguiente paso para seguir aprendiendo y mejorando. El trabajo tendría menos componente de desarrollo, y la gran mayoría del tiempo se dedicaría a hacer ingeniería inversa de los sistemas antibot que tienen las diferentes webs, esta tarea se puede realizar debido a que las páginas web, por paradigma, ofrecen el código fuente que se ejecuta en el navegador y por lo tanto se pueden observar y entender.

## 5. Objetivos de desarrollo sostenible

Los Objetivos de Desarrollo Sostenible (ODS) son un conjunto de 17 metas globales establecidas por la Asamblea General de las Naciones Unidas en 2015, que pretenden abordar los desafíos sociales, económicos y medioambientales más acuciantes que enfrenta la humanidad. Estos objetivos son una extensión de los Objetivos de Desarrollo del Milenio (ODM) y tienen el propósito de ser alcanzados para el año 2030. Desde la erradicación de la pobreza y el hambre hasta la promoción de la igualdad de género y la protección del medio ambiente, los ODS ofrecen un marco integral para hacer frente a los problemas globales de forma colaborativa y sostenible.

Dada la naturaleza del trabajo realizado se entiende que, aunque podría incidir en varios ODS, el trabajo incide principalmente en el ODS 10 - Reducción de las Desigualdades.

El ODS 10, aborda la necesidad de reducir la desigualdad dentro de los países y entre ellos. Este objetivo no solo se enfoca en la desigualdad económica, sino que también abarca otros tipos de desigualdad, como la discriminación por género, raza, etnia, religión, orientación sexual, edad, discapacidad, entre otros. La desigualdad económica es una de las más palpables y de rápido crecimiento, y tiene un efecto profundo en otros aspectos de la vida, incluido el acceso a bienes y servicios.

Cuando se trata de compras en línea de productos exclusivos, muchas veces estos productos se agotan en cuestión de segundos, lo que lleva a que se revendan a precios exorbitantemente altos en el mercado secundario. Esto crea una barrera económica que excluye a las personas que no pueden permitirse pagar más que el precio original del producto. En este contexto, una extensión de las características que se ha desarrollado, permite a los usuarios comprar productos a precios de venta al público lo que puede contribuir de manera significativa a la reducción de desigualdades económicas, al menos en este ámbito específico.

### **Mecanismos de Impacto:**

- **Acceso Equitativo:** Al garantizar que más personas puedan acceder a productos a precios de venta al público, la extensión estaría democratizando el acceso a bienes que de otro modo serían inalcanzables para muchos.
- **Desincentivo a la Reventa Especulativa:** La extensión podría desincentivar la compra de productos exclusivos solo para revenderlos a un precio más alto, lo cual a menudo hace que los precios sean inaccesibles para la mayoría de la gente.
- **Transparencia y Justicia:** Al ofrecer un método más transparente y justo para la adquisición de productos, se podría reducir la percepción y la realidad de la desigualdad en el acceso a bienes exclusivos.

## 4. Relación con las asignaturas del grado

En general, este proyecto se conecta estrechamente con varias asignaturas del grado.

### **Plataformas e-commerce**

Esta asignatura me proporcionó la base para entender el ecosistema de comercio electrónico. Los conceptos clave como modelos de venta online, pasos, casuísticas, etc. Todo esto ha sido crucial para desarrollar y optimizar la extensión.

### **Diseño de Interfaces y Experiencia de Usuario**

Los principios de usabilidad y diseño centrado en el usuario aprendidos aquí me permitieron desarrollar una extensión intuitiva y funcional, garantizando una buena experiencia para el usuario final.

### **Programación 1 y Programación 2**

Las habilidades básicas de programación y diseño de arquitectura adquiridas en Programación 1 fueron fundamentales. Programación 2 lo aprendido en relación con patrones de diseño y programación cliente-servidor han sido esenciales para hacer la extensión eficiente y efectiva.

### **Proyecto Diseño y Programación Web**

Este proyecto es crucial para entender un proyecto web desde la idea inicial hasta la implementación, un conocimiento práctico que resulta útil a la hora de desarrollar una extensión.

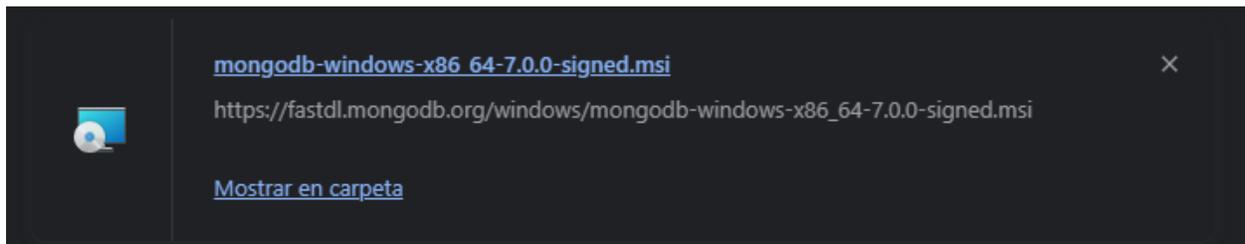
## 5. Anexos

### Manual de usuario

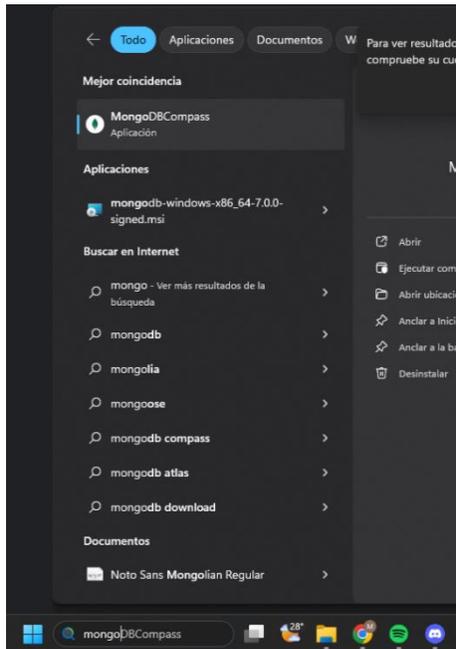
#### Instalación de la base de datos

En este manual de usuario preferiremos instalar la base de datos en local que utilizar la distribuida de Mongo.

En el siguiente enlace podemos encontrar el instalador de la base de datos: <https://www.mongodb.com/try/download/community>

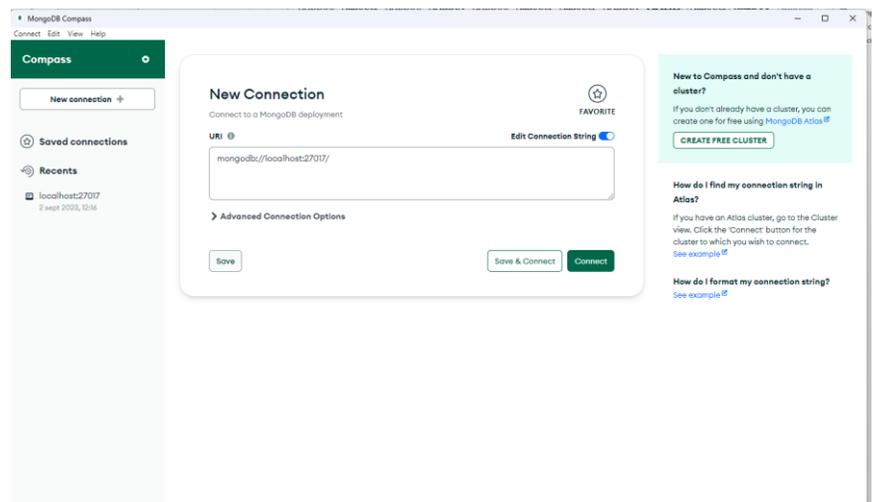


Una vez descargado podemos ejecutar y seguir los pasos de instalación. Recordemos instalar el MongoDB Compass durante el proceso.

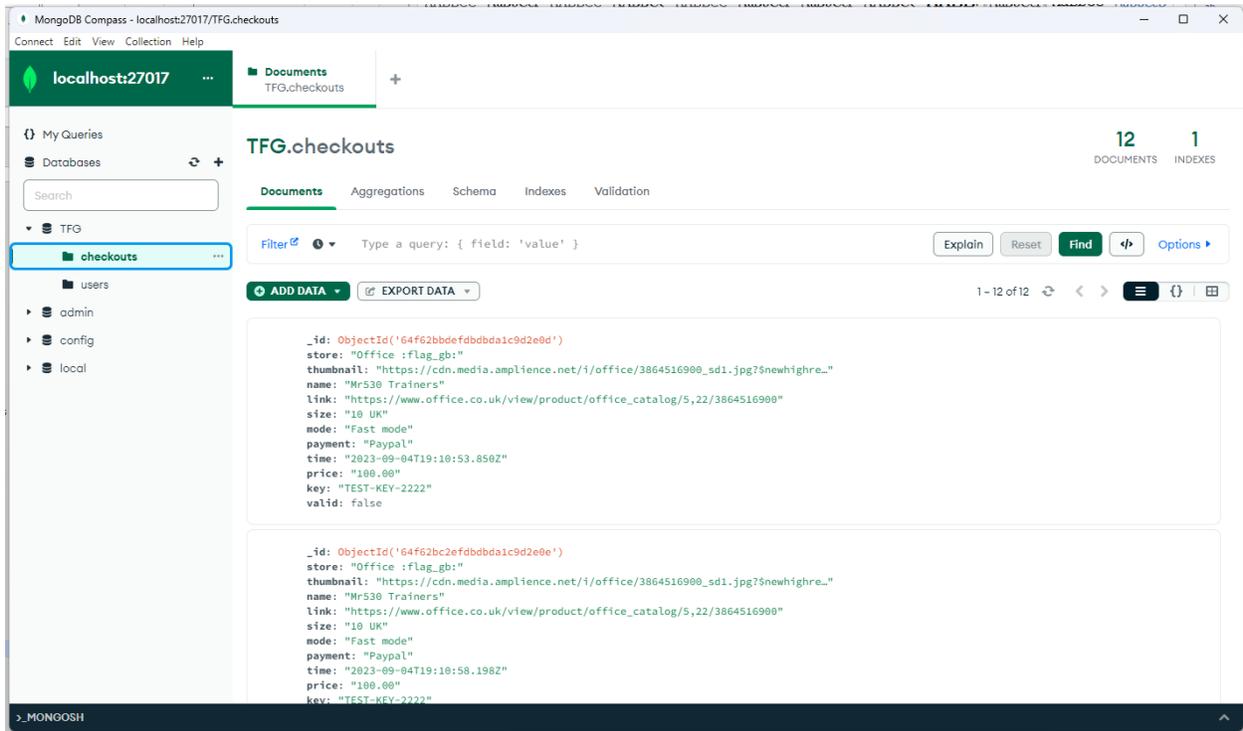


En el buscador de Windows podremos encontrar el MongoDB Compass para poder gestionar nuestras colecciones. Recordemos que, aunque esta aplicación esté cerrada, mongo siempre ejecuta los servicios para mantener tus bases de datos levantadas.

Una vez abierto nos encontraremos con una interfaz bastante amigable donde podemos conectarnos directamente clicando en 'Connect' a las bases de datos que están en estos instantes en local.

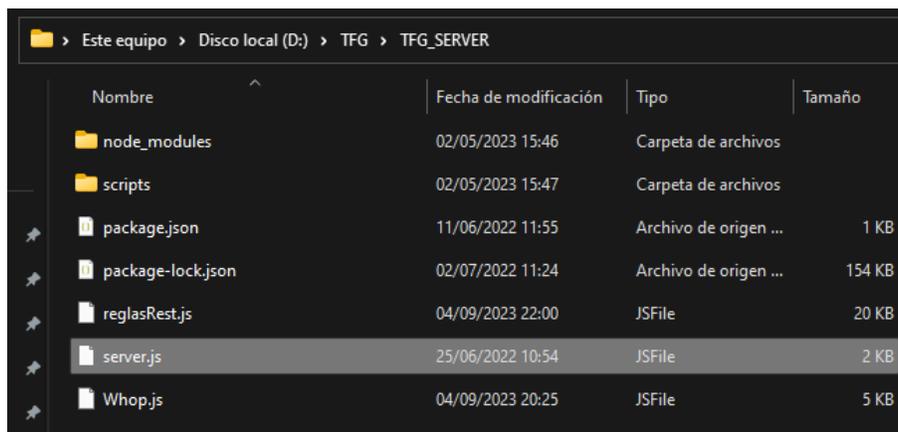


Una vez conectados podemos proceder a crear una base de datos que se llame TFG y dentro de ella, dos colecciones, una llamada 'checkouts' y la otra 'users'. En la siguiente imagen se ve el resultado.



## Servidor

Una vez el usuario ha instalado la base de datos y se ha hecho con el código fuente de la extensión poseerá dos carpetas. La primera llamada "TFG\_SERVER" y la otra "TFG\_EXT".



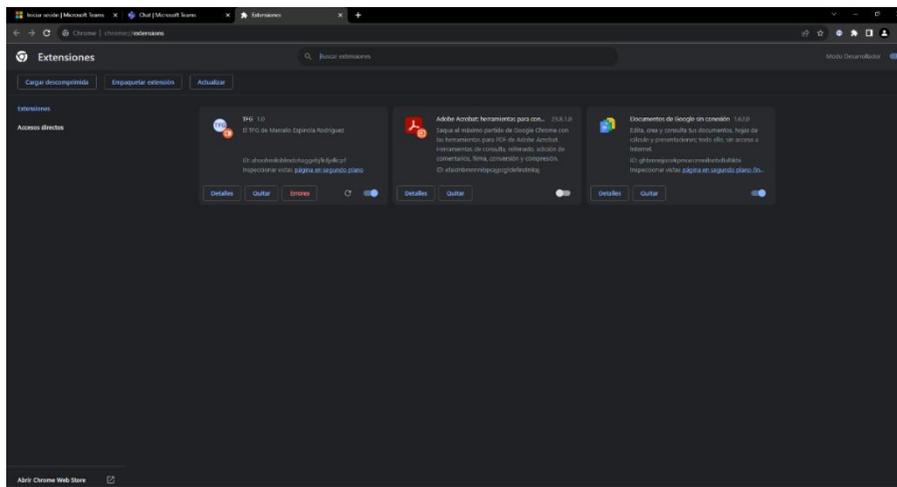
Se accede a la carpeta y cómo podemos observar en la Ilustración 26 se nos presentan una serie de archivos, es conveniente comprobar siempre el archivo ‘package.json’ antes de instalar dependencias. Procedemos a abrir un terminal y ejecutar los siguientes comandos.

- npm i
- node server.js

Con esto el servidor estará levantado y esperando.

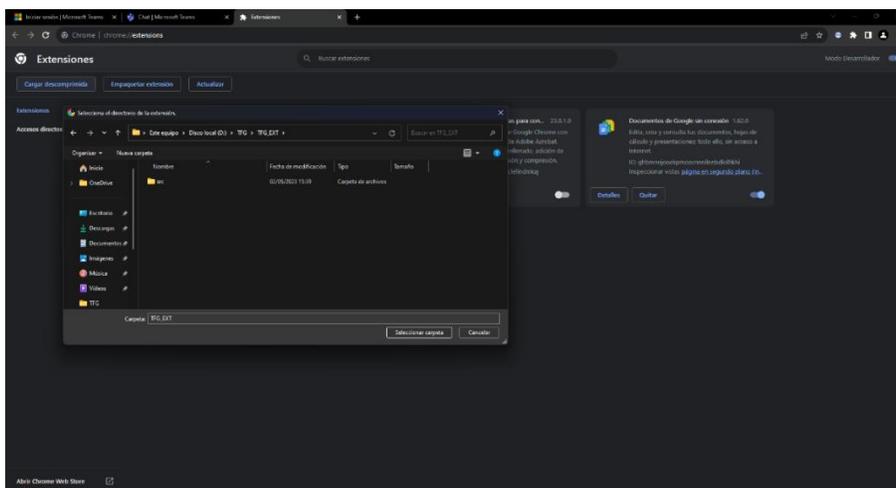
## Extensión

Accedemos a Chrome y al apartado de extensiones, lo primero que tendremos que configurar es la instalación de extensiones no firmadas, es decir, las que aún están en desarrollo.

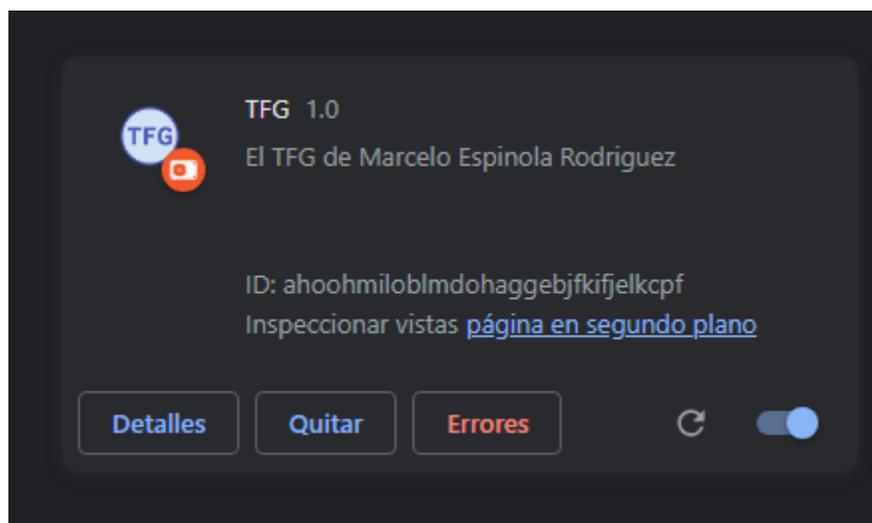


Como podemos observar arriba a la derecha hay un deslizable que nos permitirá hacer clic en el botón ‘Cargar descomprimida’.

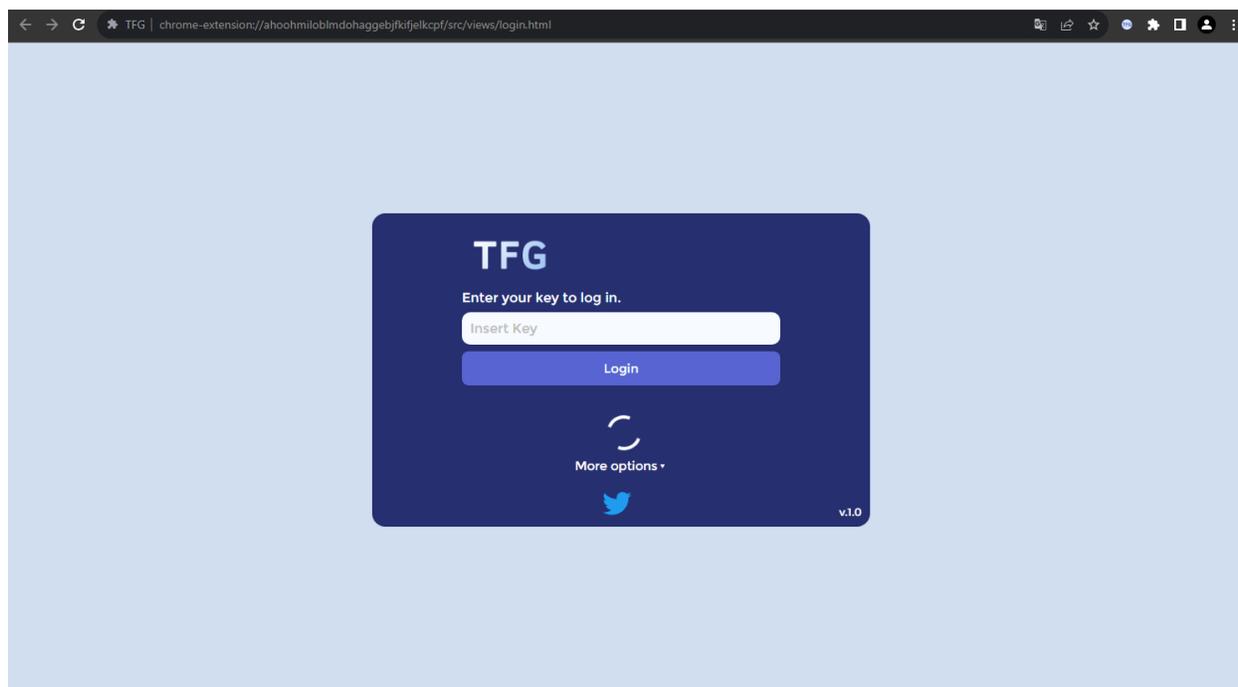
Clicamos el botón y seleccionamos la carpeta ‘TFG\_EXT’.



Clicamos en 'Seleccionar carpeta' y la extensión estará preparada para ser abierta.



Si abrimos la extensión ya nos encontramos con la página de acceso.



## 6. Bibliografía

- [1] *Tema: El comercio electrónico dentro de España.* (2023, August 30). Statista. <https://es.statista.com/temas/3167/el-comercio-electronico-dentro-de-espana/#topicOverview>
- [2] *New Balance 2002R Protection Pack Rain Cloud.* Retrieved August, 2023, from <https://wethenew.com/en/products/new-balance-2002r-protection-pack-rain-cloud>
- [3] *SO33R100 - MISSION TO MARS - Swatch® España.* Retrieved August, 2023, from <https://www.swatch.com/es-es/mission-to-mars-so33r100/SO33R100.html>
- [4] Rhinehart, C. (2021, March 7). *Billionaire Kanye West's New Yeezy Shoes Sold Out in Under 60 Seconds.* Black Enterprise. <https://www.blackenterprise.com/billionaire-kanye-west-s-new-yeezy-shoes-sold-out-in-under-60-seconds/>
- [5] Funk, L. (2023, February 16). *MSCHF Big Red Boots sell out in seconds.* FOX 5 New York. <https://www.fox5ny.com/news/mschf-big-red-boots-sell-out-in-seconds>
- [6] *Omega x Swatch "Mission To Mars" SO33R100 - Relojes exclusivos.* (2022, December 21). Relojes Exclusivos -. <https://relojesexclusivos.com/en/product/omega-x-swatch-mission-to-mars-so33r100/>
- [7] *New Balance 2002R Protection Pack Rain Cloud.* Retrieved August, 2023, from <https://wethenew.com/en/products/new-balance-2002r-protection-pack-rain-cloud>
- [8] *Streetwear.* (2023, August 16). Wikipedia. [https://en.wikipedia.org/wiki/Streetwear#Hypebeast\\_culture](https://en.wikipedia.org/wiki/Streetwear#Hypebeast_culture)
- [9] *Bun — A fast all-in-one JavaScript runtime.* Bun. Retrieved August, 2023, from <https://bun.sh/>
- [10] *Deno, The next-generation JavaScript runtime.* Deno. Retrieved August , 2023, from <https://deno.land/>
- [11] *JavaScript | MDN.* (2023, July 8). <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [12] *HTML: HyperText Markup Language | MDN.* (2023, July 17). <https://developer.mozilla.org/en-US/docs/Web/HTML>
- [13] *CSS: Cascading Style Sheets | MDN.* (2023, July 22). <https://developer.mozilla.org/en-US/docs/Web/CSS>
- [14] *Streetwear Market Research Report on Regional Size and Status 2023-2030.* (2023, June 15). MarketWatch. <https://www.marketwatch.com/press-release/streetwear-market-research-report-on-regional-size-and-status-2023-2030-2023-06-15>
- [15] *Google Chrome.* (2023, September 2). Wikipedia. [https://en.wikipedia.org/wiki/Google\\_Chrome#Extensions](https://en.wikipedia.org/wiki/Google_Chrome#Extensions)

- [16] Ismail, H. L. (2022, December 14). *Our favorite Chrome extensions of 2022*. Google. <https://blog.google/products/chrome/our-favorite-chrome-extensions-of-2022/>
- [17] *PepperScripts*. Twitter. Retrieved August, 2023, from <https://twitter.com/pepperscripts>
- [18] *MAD Scripts*. Twitter. Retrieved August, 2023, from [https://twitter.com/MAD\\_scripts](https://twitter.com/MAD_scripts)
- [19] *Chromium projects*. Retrieved August, 2023, from <https://www.chromium.org/chromium-projects/>
- [20] *Antibot*. Retrieved August, 2023, from <https://bot.sannysoft.com/>
- [21] *Bot Manager - Bot Detection, Protection and Management | Akamai*. Akamai. Retrieved August 5, 2023, from <https://www.akamai.com/products/bot-manager>
- [22] *Standards - Ecma International*. (1985). Ecma International. Retrieved August 5, 2023, from <https://www.ecma-international.org/publications-and-standards/standards/?order=last-change>
- [23] W3C. (1994). W3C. Retrieved September 5, 2023, from <https://www.w3.org/>
- [24] *Desarrollo en cascada*. (2023, May 7). Wikipedia, La Enciclopedia Libre. Retrieved August 5, 2023, from [https://es.wikipedia.org/wiki/Desarrollo\\_en\\_cascada](https://es.wikipedia.org/wiki/Desarrollo_en_cascada)
- [25] *El modelo en cascada: desarrollo secuencial de software*. (2019, March 11). IONOS Digital Guide. Retrieved August 5, 2023, from <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>
- [26] *Intercambio de recursos de origen cruzado*. (2021, June 24). Wikipedia, La Enciclopedia Libre. Retrieved August 5, 2023, from [https://es.wikipedia.org/wiki/Intercambio\\_de\\_recursos\\_de\\_origen\\_cruzado](https://es.wikipedia.org/wiki/Intercambio_de_recursos_de_origen_cruzado)
- [27] *reCAPTCHA*. (2021). reCAPTCHA. Retrieved August 5, 2023, from <https://www.google.com/recaptcha/about/>
- [28] *reCaptcha source code*. (2022). Retrieved August 5, 2023, from <https://www.google.com/recaptcha/api.js>