



Universidad Politécnica
de Madrid

**Escuela Técnica Superior de
Ingenieros Informáticos**



Grado en Ingeniería Informática

Trabajo Fin de Grado

**ESTUDIO y PROTOTIPO
EXPLORATORIO para el ANÁLISIS,
DETECCIÓN y PREDICCIÓN de
PATRONES OSCUROS WEB
HEURÍSTICOS con FIGMA**

Autor: EMILIE CELIE

Tutor(a): RICARDO IMBERT PAREDES

JOSE MARIA BARAMBONES RAMIREZ

Madrid, Abril 2024

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado
Grado en Ingeniería Informática

Título: ESTUDIO y PROTOTIPO EXPLORATORIO para el ANÁLISIS, DETECCIÓN y PREDICCIÓN de PATRONES OSCUROS WEB HEURÍSTICOS con FIGMA

Abril 2024

Autor: EMILIE CELIE

Tutor: RICARDO IMBERT PAREDES

JOSE MARIA BARAMBONES RAMIREZ

Departamento de Lenguajes y Sistemas Informáticos e Ingeniería de Software

Escuela Técnica Superior de Ingenieros Informáticos

Universidad Politécnica de Madrid

Resumen

Este Trabajo de Fin de Grado consiste en el estudio y desarrollo de un prototipo exploratorio para el análisis, detección y predicción de patrones oscuros en web, usando un método heurístico con una herramienta de desarrollo de prototipos de fidelidad. Los patrones oscuros se definen como técnicas de diseño engañosas que manipulan el comportamiento de los usuarios para beneficio del desarrollador UX o la empresa.

El trabajo se divide en varias partes esenciales. Primero, se presenta un marco teórico basado en la literatura existente sobre patrones oscuros, sus clasificaciones y el impacto negativo que tienen en la experiencia del usuario. A continuación, se describe el desarrollo de la solución basada en un plugin en la herramienta Figma, incluyendo su diseño, implementación y los algoritmos heurísticos utilizados para identificar patrones oscuros específicos, como pop-ups y cuadros emergentes. Para evaluar la eficacia del plugin, se realizaron treinta y ocho pruebas en diversas interfaces reales. Los resultados mostraron que el plugin tiene un 84,21 % de aciertos en la detección de patrones oscuros, superando las expectativas iniciales del 80 %. Sin embargo, también se identificaron áreas de mejora, como la reducción de falsos positivos, falsos negativos y la importancia del contexto.

Se alcanzaron los objetivos planteados al inicio del trabajo, el desarrollo del plugin en Figma, la aportación de detalles para mejorar interfaces, la validación del plugin mediante pruebas reales, menos la publicación del plugin. Para el trabajo futuro, se sugieren varias direcciones para mejorar y expandir el plugin, incluyendo la mejora de su precisión y la detección de una mayor variedad de patrones oscuros.

En conclusión, este Trabajo de Fin de Grado muestra que el plugin desarrollado es una herramienta efectiva para detectar patrones oscuros en interfaces, ofreciendo una solución transparente, fácil de mantener y adaptable a diferentes contextos, contribuyendo así a la creación de interfaces más éticas y centradas en el usuario.

Abstract

This Thesis involves the study and development of an exploratory prototype for the analysis, detection, and prediction of dark patterns in websites using a heuristic method with a fidelity prototyping tool. Dark patterns are defined as deceptive design techniques that manipulate user behavior for the benefit of the UX designer or the company.

The work is divided into several essential parts. First, a theoretical framework based on existing literature on dark patterns, their classifications, and their negative impact on user experience is presented. Next, the development of the solution based on a plugin in the figma tool is described, including its design, implementation, and the heuristic algorithms used to identify specific dark patterns, such as pop-ups and emergent boxes. To evaluate the effectiveness of the plugin, thirty-eight tests were conducted on various real interfaces. The results showed that the plugin achieved an 84.21% accuracy in detecting dark patterns, surpassing the initial expectation of 80%. However, areas for improvement were also identified, such as the reduction of false positives, false negatives, and the importance of context.

The objectives set at the beginning of the work were achieved, including the development of the plugin in Figma, the provision of details to improve interfaces, and the validation of the plugin through real tests, except for the publication of the plugin. For future work, several directions are suggested to improve and expand the plugin, including enhancing its accuracy and detecting a wider variety of dark patterns.

In conclusion, this Thesis demonstrates that the developed plugin is an effective tool for detecting dark patterns in interfaces, offering a transparent, easy-to-maintain, and adaptable solution to different contexts, thus contributing to the creation of more ethical and user-centered interfaces.

Tabla de contenidos

1. Introducción	1
1.1. Motivación y Justificación	1
1.2. Objetivos	2
1.3. Estructura de la Memoria	2
2. Estado del Arte	3
2.1. Teoría y Taxonomía de Patrones Oscuros	3
2.2. Figma y Trabajos Previos	5
3. Desarrollo	7
3.1. Taxonomía Propuesta	7
3.2. Propósito	10
3.2.1. Objetivo del Plugin y Elemento Detectado	10
3.2.2. Los Pop-ups	10
3.3. Arquitectura del Sistema	11
3.4. Algoritmo de Identificación de Patrones Oscuros	12
3.5. Datos de Entrada	14
3.6. Uso del Plugin	16
3.6.1. Preparación de los datos y requisitos	16
3.6.2. Llamada al Plugin	16
3.7. Resultados del Plugin	18
3.8. Estructura y Funcionamiento del Código	19
3.8.1. Archivo ui.html	19
3.8.2. Archivo code.ts	20
3.8.3. Paso de Mensajes	20
3.9. Construcción de la Base de Datos de Imágenes	20
3.10 Pruebas	21
3.10.1 Pruebas de Prototipo	21
3.10.2 Pruebas Reales	22
3.10.3 Ventajas de las Diferentes Pruebas	23
3.11 Detección Ampliada de Elementos	24
4. Resultados	27
4.1. Evaluación de Resultados	31
5. Discusión	33

TABLA DE CONTENIDOS

5.1. Discusión sobre los Resultados Obtenidos	33
5.2. Comparación con el Estado del Arte	34
5.3. Eficacia de la Solución Propuesta	34
5.4. Conclusiones	35
6. Análisis del Impacto	37
7. Conclusiones y Trabajo Futuro	39
7.1. Conclusiones	39
7.2. Trabajo Futuro	40
Bibliografía	41
Anexos	45
A. Primer anexo	45

Capítulo 1

Introducción

En la actualidad vivimos constantemente alrededor de pantallas e interactuamos aproximadamente tres horas y cuarenta y tres minutos con nuestros móviles [1]. Una de las razones que nos puede hacer interactuar tanto tiempo en ellas es el diseño de la interfaz, ya que juega un papel crucial en la experiencia del usuario influyendo en el tiempo que se pasa en una aplicación o en una página web. Uno de los principales componentes de una página web es el “frontEnd”, la parte del desarrollo web que permite al usuario ver e interactuar, esa parte que puede captar o no la atención del usuario, que hace que una interfaz le parezca atractiva y le despierte su curiosidad. Este esconde un complejo lleno de estrategias que pueden ser o no ser maliciosas para nosotros, estas estrategias se conocen como “patrones oscuros” o “patrones engaños”.

Para más contexto, un patrón oscuro se define como una instancia en las que los desarrolladores de experiencia del usuario (UX, sus siglas en inglés) utilizan su conocimiento del comportamiento humano y los deseos de los usuarios finales para implementar funcionalidades engañosas que no suelen ser beneficiosas para el usuario[2]. Es decir que los patrones oscuros se diseñan con el propósito de engañar al usuario y manipular su comportamiento.

Este Trabajo de Fin de Grado consiste en el estudio y desarrollo de un prototipo exploratorio para el análisis, detección y predicción de patrones oscuros en web usando un método heurístico a través de una herramienta de prototipado de alta fidelidad (Figma). Esta herramienta ofrece una amplia gama de funciones para crear prototipos, interfaces y diseños interactivos. En este caso, el reconocimiento heurístico es una técnica que se utiliza para identificar patrones oscuros de manera rápida y eficiente, apoyándose en la intervención humana. Se basa en una serie de reglas o principios generales que se aplican al diseño de interfaces de usuario.

1.1. Motivación y Justificación

La principal motivación para realizar este trabajo es la creciente presencia de patrones oscuros en las interfaces digitales que utilizamos diariamente. Estos

Capítulo 1. Introducción

patrones oscuros manipulan el comportamiento del usuario, haciendo que tome decisiones que no sean de su mejor interés.

Este trabajo tiene como objetivo proporcionar una herramienta que pueda utilizarse para identificar la presencia de algunos patrones oscuros de manera automática.

1.2. Objetivos

Se han definido varios objetivos específicos que guiarán el desarrollo de este trabajo:

- Desarrollar un prototipo de un plugin en Figma capaz de identificar pop-ups creados en el entorno de Figma.
- Desarrollar un plugin mejorado a partir del prototipo en Figma capaz de identificar pop-ups en las interfaces de usuarios.
- Ofrecer detalles después del análisis para mejorar las interfaces y eliminar los posibles patrones oscuros identificados.
- Probar el plugin en diversas interfaces reales para validar su eficacia y precisión, añadiendo cada interfaz real con la que se pruebe a la base de datos.
- Compartir los resultados y el plugin desarrollado, para permitir a los desarrolladores UX identificar durante el diseño los patrones oscuros. Además, aumentar la conciencia de los usuarios sobre la presencia y el impacto de aquellos en su interacción con interfaces digitales.

1.3. Estructura de la Memoria

Esta memoria se ha organizado en varios capítulos. Se empezará viendo el capítulo 2, “Estado del Arte”, en donde se habla de la literatura existente sobre patrones oscuros y sus clasificaciones, proporcionando un marco teórico sólido para el trabajo. El capítulo 3, “Desarrollo”, es el núcleo del documento y se abordará los siguientes temas, primero se verá el propósito de este plugin y que se quiere detectar, después se comentará la arquitectura del sistema. Se continuara hablando de como se usa el plugin, posteriormente se detallara sobre los resultados del plugin y también sobre la estructura y su funcionamiento. También se abordará sobre la base de datos que se construyó, las pruebas que se han realizado y para acabar se hablará de la detección ampliada de patrones oscuros. En el capítulo 4, “Resultados”, se presentan los diferente resultados posibles del plugin. El capítulo 5, “Discusión”, analiza los resultados realizados durante las pruebas, se discutirá sobre ellos y sobre la solución propuesta, luego en el capítulo 6 “Análisis del impacto”, se verá en que ámbitos el plugin tiene un impacto. Por ultimo, el capítulo 7 “Conclusiones y trabajos futuro”, resumirá los puntos importantes y discutirá el potencial trabajo futuro que podría realizarse.

Capítulo 2

Estado del Arte

Este Trabajo de Fin de Grado aborda dos temas fundamentales, los patrones oscuros y reconocimiento heurístico con Figma. Antes de profundizar en el proyecto, conviene expandir su conocimiento sobre temas previos.

Desde el principio, el objetivo era empezar por una investigación sobre que son los patrones oscuros y donde se encuentran. Además, buscar las diferentes taxonomías existentes y buscar como se podían reconocer los patrones oscuros. Una vez completada esta etapa, el enfoque era investigar sobre la herramienta de Figma para ver cuáles son sus límites y buscar potenciales plugins que puedan servir de ayuda.

2.1. Teoría y Taxonomía de Patrones Oscuros

Los patrones oscuros se definen como técnicas de diseño de interfaces que manipulan a los usuarios para que realicen acciones que no son beneficiosas para ellos sino que para los que los manipulan. No hay que confundir estos denominados patrones oscuros con los errores de diseño por parte de los diseñadores UX. Estos errores resultan de una mala planificación o ejecución y son involuntarios, a diferencia de los patrones oscuros que son voluntarios.

El objetivo principal de los patrones oscuros es aumentar la conversión, las ventas, la retención de usuarios u otros indicadores de éxito comercial a cambio de la experiencia del usuario y, en muchos casos, de su bienestar. Esta práctica ha provocado críticas éticas y legales, llevando a un creciente interés por parte de los investigadores y reguladores en identificar y mitigar los patrones oscuros para proteger y concienciar a los usuarios.

A continuación, se presentan algunos ejemplos concretos de patrones oscuros que se pudo encontrar a lo largo de las lecturas sobre este tema. Uno de ellos es cuando al final del proceso de compra, se añaden cargos adicionales como tarifas de envío o impuestos que no fueron previamente mencionados. También existen temporizadores de cuenta atrás que crean un sentido de urgencia falso para presionar al usuario a realizar una compra rápidamente. Del mismo tipo, los temporizadores en la publicidades que no te dejan cerrarlos a no ser que

Capítulo 2. Estado del Arte

haya pasado un mínimo de tiempo. Luego hay algunos patrones oscuros que su objetivo es hacer que los usuarios se sientan mal por no aceptar una oferta, utilizando mensajes como "No, gracias. Prefiero pagar el precio completo.", para que acaben aceptando la oferta. También un ejemplo muy común es cuando hacen que sea fácil registrarse para un servicio pero extremadamente difícil cancelarlo. A cerca de la investigación sobre las clasificaciones de aquellos patrones, se decidió adoptar la taxonomía propuesta por Gray et al. e inspirada de la de Harry Briggbull. La razón por la cual se continuo el trabajo con aquella es que engloba la mayoría de los patrones oscuros que se pueden ver en la actualidad y a su amplia citación en la literatura de este tema [3]. A continuación se presenta la taxonomía propuesta por Gray et al[2],

- Insistencia (Nagging): Redirección de la tarea que se esta realizando, una o más veces.
- Obstrucción (Obstruction): Hacer que un proceso sea mas difícil de acabar, teniendo malas intenciones.
- Ocultación (Sneaking): El intento de ocultar, disfrazar o retrasar la divulgación de información relevante para el usuario.
- Interferencia en la interfaz(Interface interference): manipulación de la interfaz del usuario que privilegia algunas acciones sobre otras.
- Acción forzada (Forced action): Requiere que el usuario realice una determinada acción para acceder o continuar accediendo a cierta funcionalidad.

En términos legales, varios países y regiones están comenzando a reconocer y regular el uso de patrones oscuros. Por ejemplo, la Unión Europea ha implementado regulaciones bajo el Reglamento General de Protección de Datos (RGPD), que de forma general protege las personas del tratamiento de sus datos y de su circulación. En el caso de los patrones oscuros, se podría decir que prohíbe prácticas que engañen o manipulen a los usuarios para que den su consentimiento para el tratamiento de sus datos personales [4]. Del mismo modo, en los Estados Unidos, la Comisión Federal de Comercio (FTC) ha comenzado a tomar medidas contra empresas que utilizan patrones oscuros para engañar a los consumidores. Estas regulaciones buscan proteger los derechos de los usuarios y promover prácticas comerciales justas. Sin embargo, la aplicación efectiva de estas leyes sigue siendo un desafío, dado que los patrones oscuros pueden ser difíciles de detectar, o simplemente errores de diseños, o depender del contexto y que las empresas siempre encuentran una forma de esquivar las leyes.

Desde una perspectiva ética, se puede decir que el diseño se centra en principios como la transparencia, la honestidad y el respeto por el usuario, que es todo lo contrario que hacen los patrones oscuros. Aquellas tácticas violan estos principios al manipular la percepción y las decisiones del usuario. Esto puede llevar a un sentimiento de desconfianza hacia las tecnologías digitales. Además, a largo plazo, la manipulación constante y al aprovechamiento de vulnerabilidades psicológicas pueden tener efectos negativos en la salud mental y emocional de los usuarios. Todo eso plantea serias cuestiones sobre la responsabilidad social de los diseñadores UX y las empresas.

2.2. Figma y Trabajos Previos

Se eligió Figma como herramienta no solo porque era un requisito del Trabajo de Fin de Grado, sino también por las ventajas que ofrece para el desarrollo y evaluación de diseños de interfaces. Figma permite a los desarrolladores UX evaluar sus diseños prototipados antes de su implementación, identificando y corrigiendo potenciales patrones oscuros de manera temprana. Esta ventaja hace que no se pierda recursos de tiempo ni de dinero al ser la detección antes del producto final. Además, Figma facilita la colaboración en tiempo real, permitiendo que múltiples miembros de un equipo trabajen simultáneamente en el mismo diseño, lo que permite también el ahorro de recursos de tiempo y de dinero.

El desarrollo de funcionalidades adicionales en Figma se realiza mediante plugins, son scripts o aplicaciones que extienden las capacidades de la herramienta. Estos plugins permiten automatizar tareas, integrar con otros servicios o añadir nuevas funcionalidades. Para crear esa funcionalidad, se puede elegir dos opciones, un plugin con una interfaz interactiva o uno sin interfaz. En el caso de este trabajo se eligió la opción de plugin con interfaz, por lo que se necesita escribir un código en JavaScript, junto con HTML y CSS, y para implementar su funcionalidad se requiere un código en TypeScript. Para el proceso de inclusión, se necesita crear un archivo de manifiesto JSON que define los permisos y capacidades del plugin, el desarrollo de la lógica principal, la interfaz de usuario, y la carga del plugin en Figma para su ejecución. Los plugins pueden ser ejecutados directamente desde el editor de Figma.

Durante la investigación se vio que Figma no cuenta con herramientas nativas para el reconocimiento de patrones oscuros. Lo único que se encontró era un plug-in que daba directrices sobre los patrones oscuros en una interfaz [5] y un plug-in que reconocía los errores en un diseño, llamado "design-lint"[6]. Sin embargo durante la búsqueda de trabajos previos, se encontró una herramienta relevante e interesante. Esa herramienta llamada "UIGuard" combina técnicas de visión por computadora y procesamiento de lenguaje natural para detectar automáticamente patrones oscuros en interfaces móviles. UIGuard se basa en una base de conocimiento estructurada a partir de taxonomías existentes y aplica heurísticas derivadas de esta base para identificar características de patrones oscuros. Esta herramienta alcanza un 93% de precisión a la hora de clasificar los patrones oscuros y ha sido validada a través de un estudio de usuario [7].

A pesar de los avances logrados por UIGuard, existen algunas limitaciones que justifican la necesidad de explorar nuevos enfoques. Una de las principales limitaciones es la dependencia de la inteligencia artificial (IA) lo cual puede no capturar todas las sutilezas y variaciones de los patrones oscuros en diferentes contextos y diseños de interfaces. Además, la implementación y mantenimiento de modelos de IA pueden ser costosos y requerir actualizaciones constantes para mantener la precisión en un entorno digital en rápida evolución. Otra principal limitación es que con UIGuard, la detección se hace en el producto final, lo que implica cambios adicionales después de la implementación del producto.

El plugin desarrollado en este TFG aborda estas limitaciones al enfocarse en

Capítulo 2. Estado del Arte

métodos que potencien el juicio humano para la detección de patrones oscuros. Este enfoque tiene varias ventajas clave, al no depender de modelos de IA complejos y que la detección se realiza antes de la implementación, el plugin puede ser más fácil de implementar, mantener y tener un mejor rendimiento para los desarrolladores UX. Esto también permite adaptarse a diferentes contextos y tipos de interfaces sin necesidad de reentrenar modelos. Con el método basado en el juicio humano, los usuarios pueden entender mejor cómo y por qué se detectan ciertos patrones oscuros, lo que puede mejorar su confianza en la herramienta.

Dada la limitación de las herramientas existentes y la importancia del factor humano en la detección de patrones oscuros, el enfoque en este TFG será explorar y desarrollar una herramienta que potencien el juicio humano para una detección más precisa y efectiva de estos patrones en el diseño de interfaces, sin requerir el uso de inteligencia artificial.

Capítulo 3

Desarrollo

En este capítulo se abordará una visión sobre el proceso detallado del desarrollo del plugin. Se empezará por hablar de la taxonomía propuesta en este trabajo, luego se especificará cuál es el propósito de este plugin, para poder entender mejor qué se quiere analizar y para qué. Después, se hablará de la arquitectura del sistema, también se verán los diferentes datos de entrada que se usaron, para seguir con un apartado que hablará de cómo usar el plugin. Se detallarán y explicarán los posibles resultados del plugin y cómo ocurren, después se proporcionará más información sobre la estructura y la lógica del análisis. Se abordará posteriormente la base de datos que se creó, y se comentarán las pruebas realizadas. Por último, se verá que se consiguió analizar otro tipo de patrón oscuro.

3.1. Taxonomía Propuesta

Los ejemplos de patrones oscuros que se han mencionado son solo una pequeña cantidad de cómo las interfaces de usuario pueden ser diseñadas para manipular a los usuarios. Existen tantos tipos que se crearon taxonomías para poder agruparlos, como ya se mencionó en el capítulo 2.

Se decidió modificar ligeramente aquella taxonomía para realizar otra posible clasificación. La razón por la cual se decidió este cambio es porque en la mayoría de las taxonomías vistas, muy pocas mencionaban los patrones oscuros que hacen poner más datos que los necesarios, para poder recoger información suplementaria, y parecía importante mencionarlo en este trabajo.

- **Insistencia (Nagging):** Redirección de la tarea que se está realizando, una o más veces. Algunos patrones oscuros de este tipo son, una ventana emergente que aparece repetidamente pidiendo al usuario que se suscriba a un boletín, también recordatorios insistentes para actualizar el software o la aplicación (ver figura 3.1).
- **Obstrucción (Obstruction):** Hacer que un proceso sea más difícil de acabar, teniendo malas intenciones. Por ejemplo, un proceso de cancelación de una

Capítulo 3. Desarrollo

suscripción que requiere múltiples pasos y confirmaciones, haciendo que el usuario se desanime y abandone la cancelación.

- **Ocultación (Sneaking):** El intento de ocultar, disfrazar o retrasar la divulgación de información relevante para el usuario. Un ejemplo de de este tipo sería cuando se añade automáticamente productos o servicios adicionales al carrito de compras sin el consentimiento del usuario, ocultando estos detalles en texto pequeño o en un lugar poco visible.

- **Interferencia en la interfaz (Interface interference):** Manipulación de la interfaz del usuario que privilegia algunas acciones sobre otras. Por ejemplo que los botones de “Aceptar” destaquen en colores brillantes y que los botones de “Rechazar” estén de colores apagados, dificultando al usuario rechazar una oferta o suscripción (ver figura 3.3).

- **Acción forzada (Forced action):** Requiere que el usuario realice una determinada acción para acceder o continuar accediendo a cierta funcionalidad. Como pro ejemplo, obligar al usuario a compartir su dirección de correo electrónico o a registrarse para continuar usando un servicio o para descargar un recurso (ver figura 3.2).

- **Datos (Data):** Prácticas diseñadas para recopilar más datos del usuario de los necesarios. Un ejemplo pueden ser los formularios de registro que solicitan información personal adicional, como el número de teléfono o la dirección postal, cuando solo se necesita un correo electrónico y una contraseña (ver figura 3.4).

El tipo de patrón oscuro que se quiere detectar en este trabajo pertenece al grupo de Insistencia, pero se detallará más adelante en la sección 3.2.

3.1. Taxonomía Propuesta

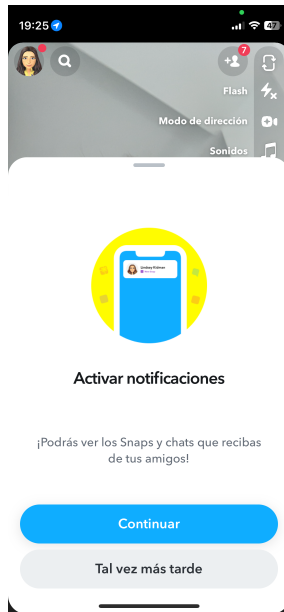


Figura 3.1: Ejemplo de un patrón oscuro de tipo insistencia, donde SnapChat nos dice de activar las notificaciones sin posibilidad de rechazarlo permanentemente

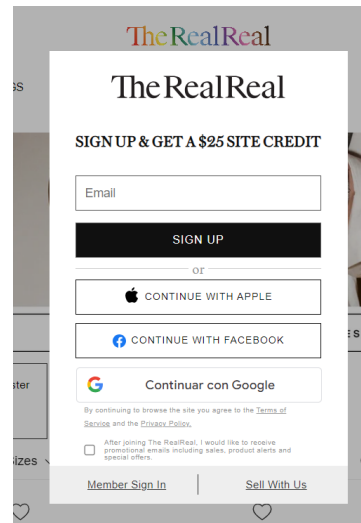


Figura 3.2: Ejemplo de un patrón oscuro de tipo acción forzada, donde The RealReal obliga a registrarse para poder seguir comprando en el sitio web



Figura 3.3: Ejemplo de un patrón oscuro de tipo interferencia de la interfaz, donde Shein nos propone ofertas, poniendo el botón de aceptar en negro y grande, pero la cruz y las condiciones para poder tener acceso a las promociones en pequeño y en gris

The image shows a browser window with a dark overlay at the top. The URL is `edreams.es/travel/secure/?locale=es_ES#details/16847694143/`. Below the URL, there are three green checkmarks: '¿Ventana o pasajer?', 'Personaliza tu viaje', and 'Reserva y relájate'. A blue '+ Añadir' button is on the right. The main content is a 'Detalles de contacto' form. It has two columns of input fields. The first column contains 'E-mail' (filled with 'emilie.celie.perez@gmail.com' and a green checkmark), 'Prefijo internacional' (dropdown with 'España (+34)'), 'Dirección' (empty, red border, 'x' icon, 'Introduce una dirección'), and 'Ciudad' (empty, red border, 'x' icon, 'Introduce un nombre de ciudad'). The second column contains 'Confirmar e-mail' (filled with 'emilie.celie.perez@gmail.com' and a green checkmark), 'Número de teléfono' (empty, red border, 'x' icon, 'Introduce un número de teléfono'), 'Código postal' (empty, red border, 'x' icon, 'Escribir código'), and 'País' (dropdown with 'Bahamas'). At the bottom, there is a checkbox 'Recomendado' (unchecked) and the text 'Guarda tus datos de manera segura para futuras compras'. A small disclaimer at the very bottom reads: 'Una vez hecha tu reserva, recibirás la confirmación en tu dirección de e-mail, así como otras ofertas de viajes similares. Si ya no deseas recibir ofertas exclusivas, haz clic en [este enlace](#).'

Figura 3.4: Ejemplo de un patrón oscuro de tipo Datos, obligando a rellenar un formularios sobre los datos que no son necesarios ay que para recibir el billete de avión en el correo solo hace falta el email

3.2. Propósito

3.2.1. Objetivo del Plugin y Elemento Detectado

Durante la investigación sobre los patrones oscuros, se descubrió que existía una gran cantidad y variedad de patrones engañosos. Lo cual es excesivo para este trabajo, por lo tanto se decidió centrarse en la detección de solo uno de ellos. Se ha elegido los pop-ups, debido a que sus características son más evidentes de detectar.

Poder identificar y señalar los problemas de la ventana emergente es una gran ventaja para evitar a los usuarios de aplicaciones o sitios web de ser manipulados, pero también permitir a los diseñadores UX de mejorar su interfaz y hacer que la experiencia de un usuario sea más fluida y menos intrusa. El plugin encima de solo detectar la presencia de un pop-up, como se dijo anteriormente, proporciona más detalles, es decir que indica cual cree que es la razón por la cual podría ser un pop-up.

3.2.2. Los Pop-ups

Los pop-ups son ventanas o diálogos que aparece sobre el contenido de la página. Pueden ser modales, es decir que desactivan el contenido de la pagina, o pueden ser no modales, que permite interactuar con el contenido de fondo. Si el fondo de la pantalla es más oscuro se le llama a este tipo "lightbox", de lo con-

trario, no tiene ningún nombre especial [8]. En la herramienta no se diferencian los pop-ups de tipo “lightbox”.

El plugin se diseñó para poder detectar lo pop-ups en Figma. La mayoría de estas ventanas emergentes son los típicos “aceptar los cookies”, pero también se encuentran muchos anuncios así, formularios de inscripción, notificaciones o cualquier otro contenido que interrumpa la navegación del usuario. Estos pop-ups pueden aparecer de manera repentina y en cualquier sitio web o aplicación. Suelen ocupar una gran parte de la pantalla, pudiendo así desviar la atención del usuario de su tarea.

Para lograr la detección de este patrón oscuro, el plugin se centra en parámetros visuales y de posición. Mira si el Pop-up ocupa un cierto tamaño de la pantalla, y también identifica si esta ventana emergente cubre por parte o tapa por completo algún texto importante.

3.3. Arquitectura del Sistema

El sistema desarrollado para el análisis, detección y predicción de patrones oscuros se estructura en una arquitectura modular, compuesta por varias capas. Cada capa tiene una responsabilidad específica y emplea tecnologías diferentes. Esta arquitectura facilita la actualización, escalabilidad y mantenimiento del sistema.

La primera capa es la del Front-End, que se encarga de la interacción con el usuario final. Aquí es donde los diseñadores y desarrolladores UX pueden cargar las interfaces que quieren analizar. Para permitir la visualización de la interfaz de entrada, se utilizó HTML, CSS y JavaScript, que son lenguajes utilizados para crear interfaces en aplicaciones o sitios web. En el caso del plugin, se usaron para crear una interfaz de entrada para una buena interacción con el usuario de la herramienta desarrollada.

El sistema también se compone de un capa de Análisis y detección, que se ocupa de la implementación de los algoritmos heurísticos para la identificación de patrones oscuros en las interfaces y de dar los resultados. Se usa para ello, TypeScript que es un lenguaje de programación tipado que se basa en JavaScript.

Luego, se encuentra también la capa de Back-End, que contiene toda la lógica encontrada en la implementación y el despliegue del plugin. Se utiliza Node.js como entorno de ejecución para el desarrollo del servidor del plugin, permite la ejecución de JavaScript y de TypeScript. Node.js se utiliza en conjunto con TypeScript para aprovechar el tipado estático y es conocido por su modelo de entrada/salida no bloqueante (permite manejar múltiples operaciones de manera eficiente) lo que mejora la robustez del código, da respuestas rápidas y facilita el mantenimiento a largo plazo. Node Package Manager (NPM), es el gestor de paquetes de Node.js que facilita la gestión de dependencias y la actualización de paquetes. En la figura 3.5, se puede observar el diagrama de la arquitectura del sistema.

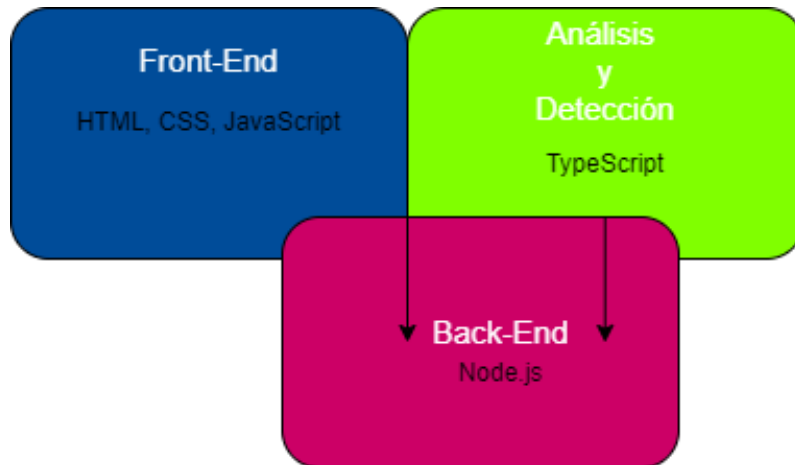


Figura 3.5: Diagrama de la arquitectura del sistema

3.4. Algoritmo de Identificación de Patrones Oscuros

El algoritmo de identificación de patrones oscuros funciona de la siguiente manera, primero se identifica todos los nodos dentro del frame, verificando la presencia de nodos de texto y formas geométricas. En el caso de que se haya encontrado alguna forma pero ningún nodo de texto, se verifica si aquella cubre más del 25% del área total del frame, lo que podría indicar la presencia de un pop-up. Si se ha detectado la presencia de nodos de texto y alguna forma, se mira si aquella cubre las zonas de texto, lo cual puede considerarse un patrón oscuro si dicha superposición dificulta la visibilidad del texto. Si no se ha detectado ninguna forma o si hay nodos de textos sin formas, no se realiza el análisis en estos casos. Los resultados se muestran, indicando si se detectaron patrones oscuros y proporcionando detalles sobre cada caso. Para una mejor visualización, en la figura 3.6 se puede ver el diagrama de flujo.

3.4. Algoritmo de Identificación de Patrones Oscuros

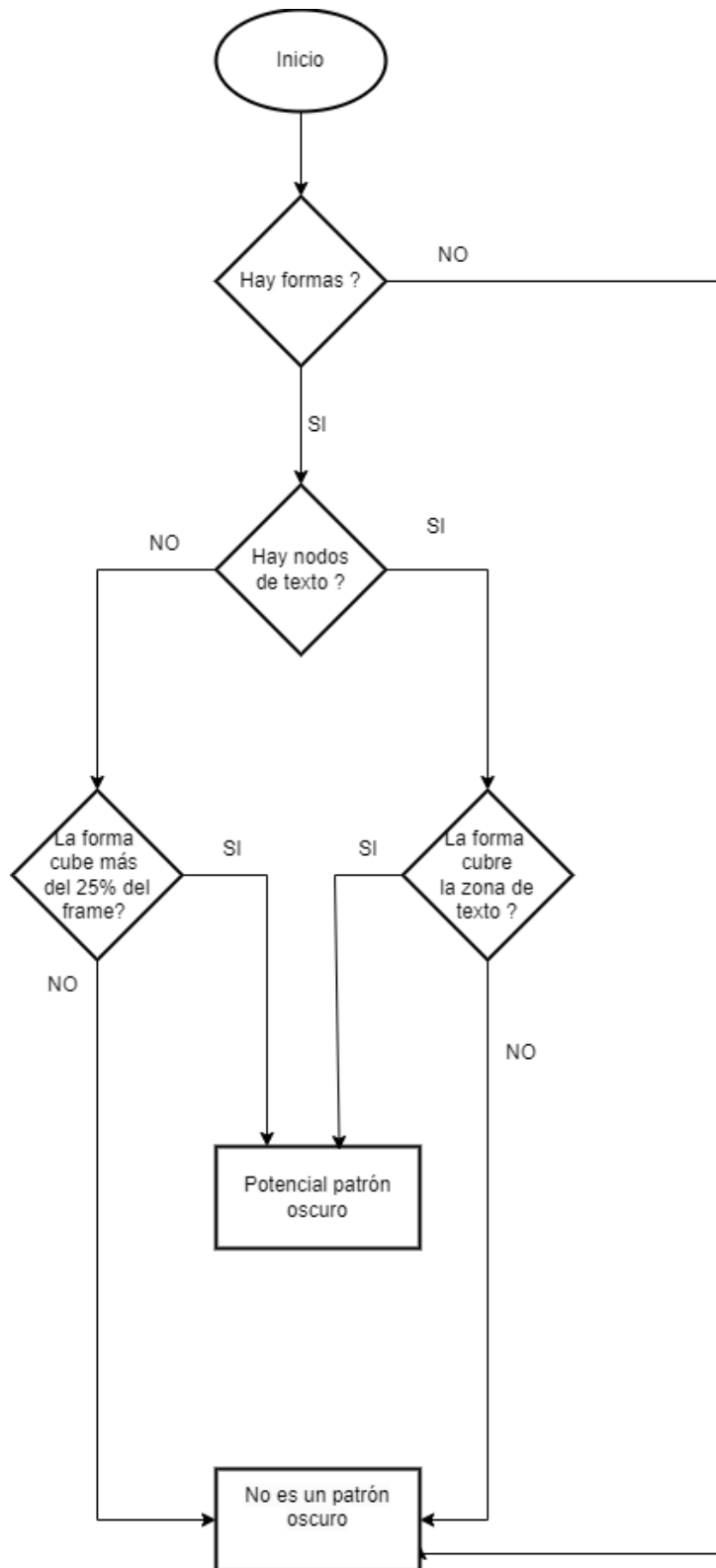


Figura 3.6: Diagrama de flujo

3.5. Datos de Entrada

Durante el primer desarrollo del plugin, para poder realizar los primeros test, los datos de entrada eran simplemente unos “prototipos”. Es decir con la herramienta del plugin se creaba un frame y se añadía un cuadro simple o con texto para poder ver si lo realizado funcionaba. En las figuras 3.7 y 3.8 se pueden observar los primeros datos de entrada con lo cual se empezó.

Una vez el plugin desarrollado por completo se realizaron las pruebas con ejemplos reales. Para poder realizar un análisis, los datos de entrada tienen que ser 2 imágenes, un antes y un después, como se puede observar en las figuras 3.9, 3.10, 3.11 y 3.12. Es decir, la primera imagen es antes de que aparezca el patrón oscuro y la segunda imagen es cuando se puede observar el patrón oscuro. Estos datos deberán ser manipulados previamente para poder usar el plugin correctamente.

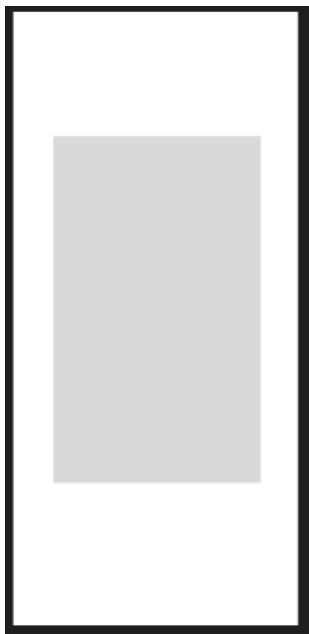


Figura 3.7: Imagen del ejemplo prototipo 1

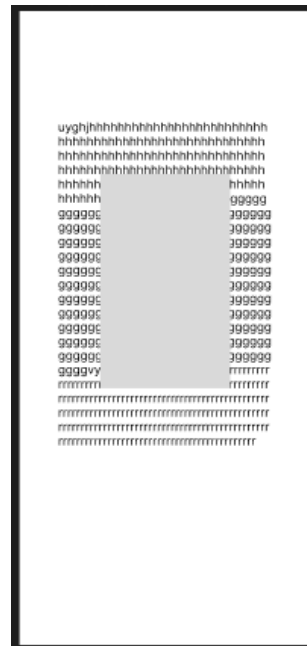


Figura 3.8: Imagen del ejemplo prototipo 2

3.5. Datos de Entrada

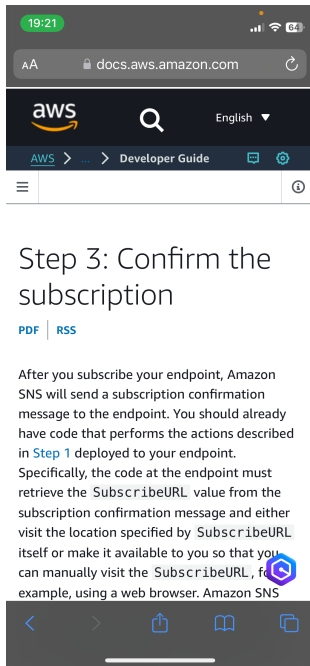


Figura 3.9: Imagen del ejemplo real 1 (antes)

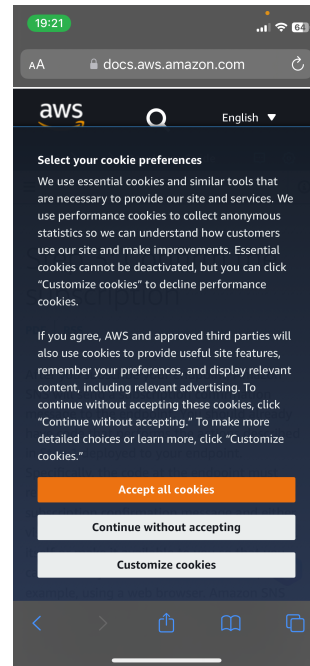


Figura 3.10: Imagen del ejemplo real 1 (durante)

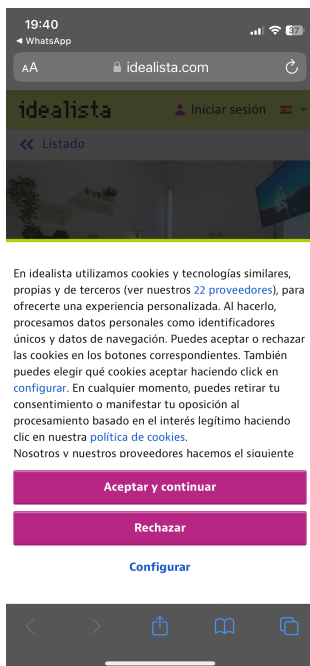


Figura 3.11: Imagen del ejemplo real 2 (antes)



Figura 3.12: Imagen del ejemplo real 2 (durante)

3.6. Uso del Plugin

Para poder usar el plugin es necesario explicar cuales son los requisitos de entrada para que todo funcione correctamente y como se llama al plugin para iniciar el análisis.

3.6.1. Preparación de los datos y requisitos

El requisito principal de entrada para el plugin es que el contenido a analizar debe ser un “frame”. Es decir, que el fondo en donde se encuentra el patrón oscuro tiene que ser de tipo “frame”. Se ha elegido trabajar con frames porque son más fáciles de manejar en Figma y permiten superponer elementos, y trabajar con ellos, lo cual es esencial para recrear y analizar patrones oscuros.

Antes de llamar al plugin, hay que efectuar algunos cambios a lo que se quiere analizar. Para empezar, cabe destacar que hacen falta dos imágenes, una foto del “antes” de que aparezca el potencial patrón oscuro y una foto “cuando” esté el patrón oscuro.

Para poder realizar el frame que se va a analizar, primero se importan las dos imágenes, luego hay que redimensionarlas a un tamaño estándar de móvil, y para ello se usa un plug-in externo “Image Resizer Plugin”, encontrado en la API de Figma [9]. Una vez esta etapa realizaba, hay que transformar la imagen del “antes” a un “Frame”, ya que es el único tipo de entrada aceptado por el plug-in. Para aquello se usa el plug-in llamado “Node to Frame”, también encontrado en la API de Figma [10]. Por ultimo, en la imagen donde hay el potencial patrón oscuro se recorta el pop-up y se pone en la imagen del “antes” que es ahora un frame. En el caso de que haya una zona de texto, hay que añadir con la herramienta de Figma una zona de texto en el frame antes de colocar el patrón oscuro por encima. De tal forma que se recrea el patrón oscuro que se quiere analizar.

3.6.2. Llamada al Plugin

Una vez que se han realizado las maniobras previas, se ejecuta el plugin y se siguen las instrucciones proporcionadas en la interfaz de usuario.

La llamada al plugin se realiza en el entorno de Figma, primero aparece un interfaz de usuario en la cual se pide seleccionar un frame para poder empezar el análisis, como se puede ver en la figura 3.13. Se dispone de dos botones, primero el botón de cancelar “Cancel” que si se da se cerrará el plugin y luego el botón “Next”. Se selecciona el “frame” creado para el analisis y se cliqua en el botón “Next”, lo cual lanzara el análisis y mostrara el resultado con mas detalles en Figma. En el caso que se seleccione otro tipo que no sea un frame o que no se seleccione nada, un mensaje de error aparecerá en la interfaz, tal y como se puede observar en las figuras 3.14 y 3.15.

Si los pasos previos al llamar el plugin se han realizado correctamente y se ha seleccionado el frame, el análisis empezará y generará los resultados en un cuadro de texto, para una mejor visión de aquellos.

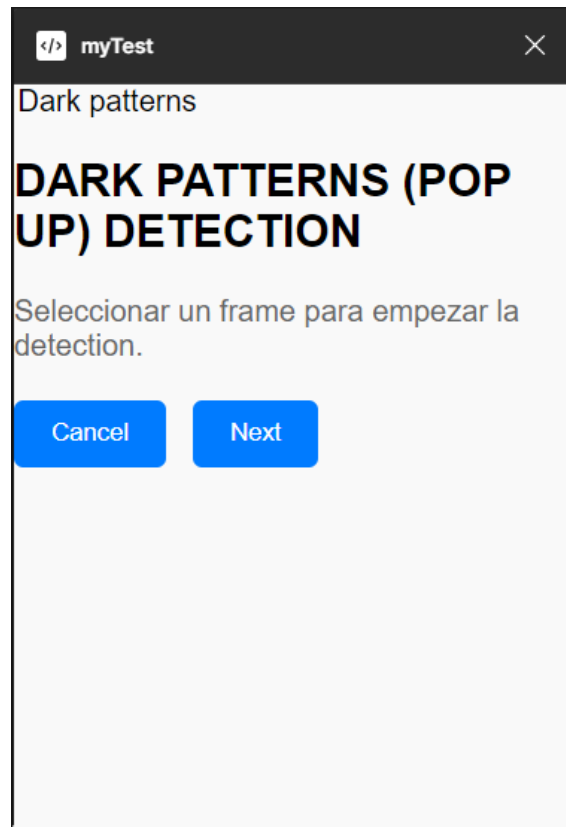


Figura 3.13: Imagen de la interfaz

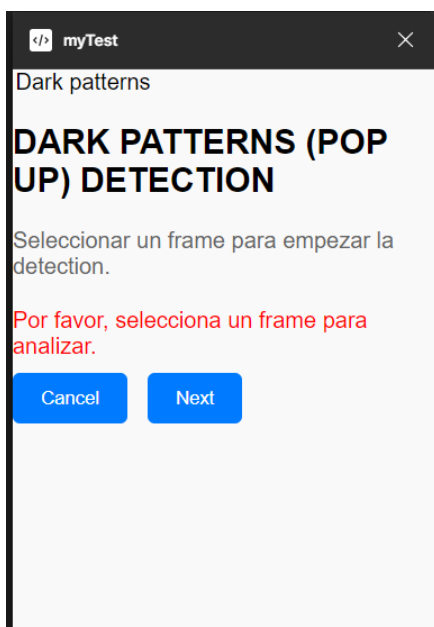


Figura 3.14: Mensaje de error 1

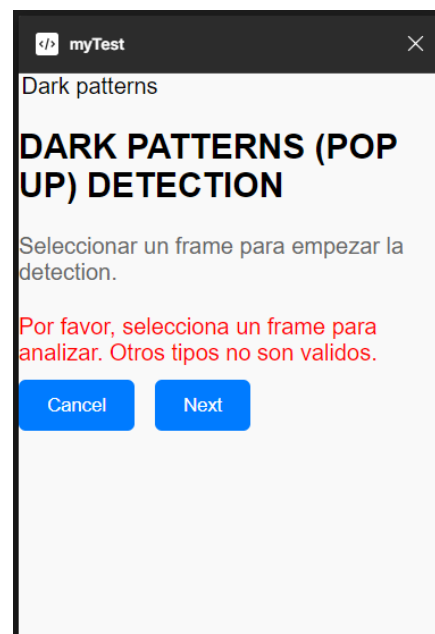


Figura 3.15: Mensaje de error 2

3.7. Resultados del Plugin

Una vez que el código realizó el análisis aparece en Figma un frame que contiene los resultados. Los posibles resultados son los siguientes:

- “Potential dark pattern detected: shape over text zone” (traducido al español, “Un potencial patrón oscuro ha sido detectado: forma encima de una zona de texto”)
- “Potential dark pattern detected: Shape covers too much area” (traducido al español, “Un potencial patrón oscuro ha sido detectado: forma ocupa demasiada superficie”)
- “A shape is present, but no dark pattern detected” (traducido al español, “Una forma esta presente, pero no se ha detectado un patrón oscuro”)
- “All good! Text is present, but no shapes were found.” (traducido al español, “Todo bien! Hay una zona de texto, pero no se ha encontrado formas”)
- “All good! No potential dark pattern detected” (traducido al español, “Todo bien! No se ha detectado un potencial patrón oscuro”)

A continuación se hablará con más detalles sobre los diferentes resultados. La primera posibilidad, es cuando se ha detectado una forma que sea cuadrada o redonda que tapa una zona de texto. El segundo resultado, es cuando no hay zona de texto pero se ha detectado una forma que cubre más de un cuarto del tamaño de la imagen. La tercera posibilidad, ocurre cuando hay una zona de texto y se a detectado un forma sin embargo aquella no cubre el texto, por lo cual no tendría que ser un patrón oscuro. Los dos últimos resultados, es cuando no se a encontrado un potencial patrón oscuro, pero en uno de los dos nos informa que se ha encontrado una zona de texto pero que nada la cubre.

Los resultados obtenidos a través de este análisis deben ser considerados con precaución, ya que existe la posibilidad de que sean erróneos. Esto se debe a la presencia potencial de falsos positivos o también falsos negativos, es decir, situaciones en las que el análisis indica la presencia de un patrón oscuro cuando en realidad no está presente o que no detecta la presencia de patrón oscuros cuando sí existe.

Los falsos positivos pueden surgir debido a diversas razones. Una de las principales razones es el contexto de la situación, es decir si por ejemplo alguien ha hecho un pedido y aparece en la pantalla un cuadro con detalles de su pedido diciendo que se ha realizado correctamente, este caso no es realmente un patrón oscuro ya que es acorde a lo realizado previamente y es informativo. También pueden ser debidos a que el cuadro que aparece en la pantalla es más pequeño que el cuarto de la pantalla por lo que el plugin no lo consideraría como un patrón oscuro.

Los falsos negativos, ellos, podrían surgir en una mal diseño, que por mala suerte una parte de una forma tape sin querer un poco de texto y se podría detectar como un patrón oscuro sin llegara a realmente serlo, también podría depender del contexto.

3.8. Estructura y Funcionamiento del Código

Por lo tanto, es importante revisar cuidadosamente los resultados y considerar el contexto específico antes de tomar decisiones basadas en ellos.

3.8. Estructura y Funcionamiento del Código

El código consta de dos partes principales, un archivo HTML llamado "ui.html" y un archivo en TypeScript llamado "code.ts".

A continuación se puede observar en la figura 3.16 Todos los archivos que componen la carpeta del plugin. Para entender más sobre la estructura de los archivos se puede ver más en detalla con la figura 3.17.

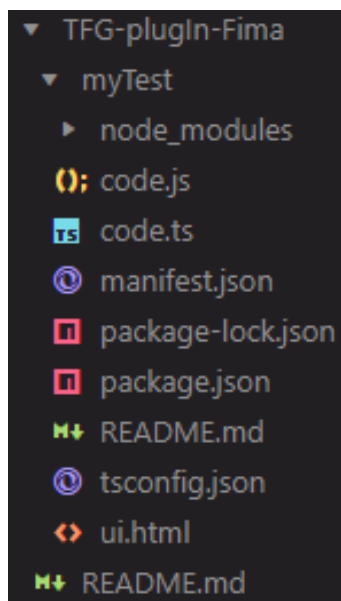


Figura 3.16: Imagen de la estructura

Acto seguido se detallara las dos partes principales y como se logran comunicar entre ellas.

3.8.1. Archivo ui.html

El archivo ui.html permite proporcionar la interfaz de usuario para la interacción con el plugin. Es decir, se define la interfaz de usuario para el análisis de detección de patrones oscuros. Esta interfaz incluye un título, un mensaje de instrucción y dos botones: Cancelar "Next".

Se encuentra en el archivo un elemento div con el id ".error-msg" que se utiliza para mostrar mensajes de error al usuario. Este elemento se actualiza dinámicamente según el estado de la selección del frame en Figma, gracias al paso de mensajes entre los archivos.

Se han asignado funciones de manejo de eventos a los botones Cancelar "Next".

Capítulo 3. Desarrollo

Cuando el usuario hace clic en estos botones, se envían mensajes al entorno de Figma a través de `postMessage` para realizar las acciones especificadas.

3.8.2. Archivo `code.ts`

El archivo `code.ts` contiene la lógica principal del plugin, que se ejecuta en el entorno de Figma. Lo primero que hace el código es mostrar la interfaz de usuario definida en el archivo `ui.html` mediante el método `figma.showUI`. Después de eso, procede a comunicarse con el archivo `ui.html` y se realiza o no el análisis querido.

3.8.3. Paso de Mensajes

Como se menciono ligeramente anteriormente estos dos archivos se comunican entre si, para poder detectar si los requisitos de entrada son los correctos o si se quiere seguir o salir.

Para poder ver si los requisitos previos son correctos, se define un evento `onmessage` en la ventana para manejar los mensajes enviados desde el entorno de Figma. Este evento se activa cuando se reciben mensajes de verificación sobre la selección del frame y se actualiza dinámicamente el mensaje de error en la interfaz de usuario. Estos mensajes son como códigos numéricos para indicar el estado del proceso o mensajes de texto para mostrar al usuario. Cuando se lanza el plugin sin que se seleccione un frame o se selecciona otro tipo que no sea un frame, la verificación de la selección se hace en `code.ts`. Luego, se envía a `ui.html` a través de mensajes con códigos numéricos: 1 cuando es correcto, 0 cuando no se ha seleccionado nada y 2 cuando se selecciona otro tipo que no sea un frame.

Para saber si se quiere seguir o cancelar todo, se define un evento `onmessage` para el objeto `figma.ui`, que escucha los mensajes enviados desde la interfaz de usuario. Cuando se recibe un mensaje, se ejecuta la lógica correspondiente según el tipo de mensaje. Es decir, si se recibe un mensaje del tipo `'next-page'`, se inicia la lógica para analizar el frame seleccionado en Figma. Se obtiene el frame seleccionado y se realizan comprobaciones para verificar si es válido para el análisis. Si el mensaje recibido es `'actionExit'`, se cerrará automáticamente el plugin con el método `figma.closePlugin`.

3.9. Construcción de la Base de Datos de Imágenes

Se creo una base de datos de imágenes reales durante el desarrollo del plugin y su evaluación. Esta base de datos se construyó con el propósito de proporcionar ejemplos de patrones oscuros para la prueba y validación del plugin.

Después de la primera parte del desarrollo, es decir cuando las pruebas de prototipo se consideraron correctas, se empezó a buscar ejemplos de patrones oscuros de tipo pop-ups en la vida real. Mientras íbamos navegando por internet en el móvil se iba recopilando todos los patrones oscuros de tipo pop-up y también

de tipo "banners". Cada vez que se veía uno, como se comentó anteriormente, se hacía una captura de pantalla del "antes" de que llegue el patrón oscuro y una "durante" el patrón oscuro, para así poder realizar las manipulaciones previas antes del análisis. Estos patrones se encontraron principalmente en sitios web y en aplicaciones móviles, y la gran mayoría de ellos son sobre los "cookies". Esta metodología permitió obtener una amplia variedad de ejemplos de pop-ups y de "banner" en situaciones reales de uso.

Al finalizar el período de recopilación, se logró obtener un total de 78 imágenes, de las cuales 39 ejemplificaban distintos tipos de potenciales patrones oscuros. Estas imágenes se organizaron y categorizaron para facilitar su utilización en las pruebas y evaluaciones del plugin.

La creación de esta base de datos de imágenes proporcionó una colección diversa y representativa de ejemplos de patrones oscuros, lo que permitió una evaluación del funcionamiento y rendimiento del plugin en la detección de diferentes tipos de patrones oscuros en entornos de usuario reales.

has usado la para hacer la base de datos, porque has escogido esas fotos, que es lo que estabas mirando

3.10. Pruebas

Durante el desarrollo del plugin, se realizaron varios tipos de pruebas para poder asegurar el buen funcionamiento y lógica de lo programado. Se realizaron primero pruebas de "prototipos" para poder detectar los posibles fallos, mejorar la lógica o algunas características. Para así tener una versión más compleja y afinadas para poder probar con pruebas reales".

3.10.1. Pruebas de Prototipo

Inicialmente, el prototipo del plugin fue creado y probado con ejemplos simples generados en Figma manualmente. En esta primera versión, el plugin solo era capaz de identificar si existía un patrón oscuro en la interfaz sin proporcionar detalles adicionales y enseñando el resultado en la consola de Figma. Esta etapa fue fundamental para establecer la funcionalidad básica del plugin y asegurar que pudiera detectar los elementos emergentes (pop-ups). En las figuras 3.18 y 3.19 se pueden observar como eran las pruebas prototipadas.

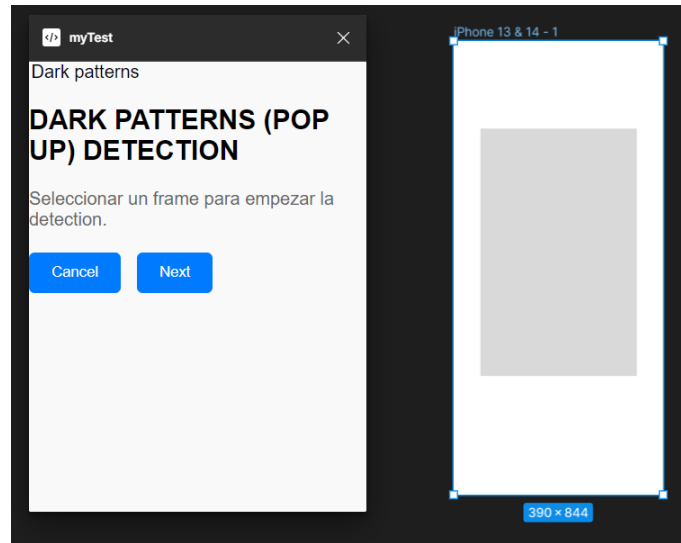


Figura 3.18: Imagen de una prueba de prototipo realizada

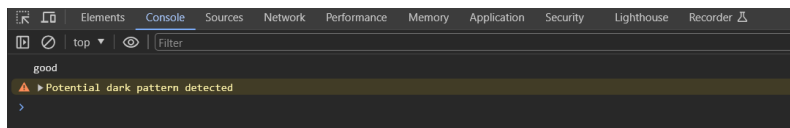


Figura 3.19: Imagen del resultado sin detalles en la consola

A medida que el desarrollo progresaba, se introdujeron mejoras significativas. Una de las primeras mejoras fue la incorporación de la capacidad del plugin para detectar zonas de texto dentro de los frames. Esta funcionalidad se añadió para mejorar la precisión del análisis y permitir la identificación de patrones oscuros que incluían elementos textuales, como mensajes engañosos o llamadas a la acción manipulativas.

Con la adición de la detección de zonas de texto, el plugin comenzó a proporcionar detalles más específicos sobre los patrones oscuros identificados. Esto incluyó información sobre la ubicación de los elementos detectados y una descripción de los posibles efectos negativos sobre la experiencia del usuario.

3.10.2. Pruebas Reales

Estas mejoras permitieron realizar pruebas más exhaustivas utilizando ejemplos reales. Para esto, se utilizaron las imágenes recopiladas en la base de datos mencionada anteriormente. Cada prueba consistió en analizar un par de imágenes (una del “antes” y otra del “después” de la aparición del patrón oscuro) para evaluar la capacidad del plugin de identificar y describir correctamente los patrones oscuros presentes. Los resultados de estas pruebas se describen en el apartado 4. En las figuras 3.20 y 3.21 se pueden observar como eran las pruebas reales que se realizaron.

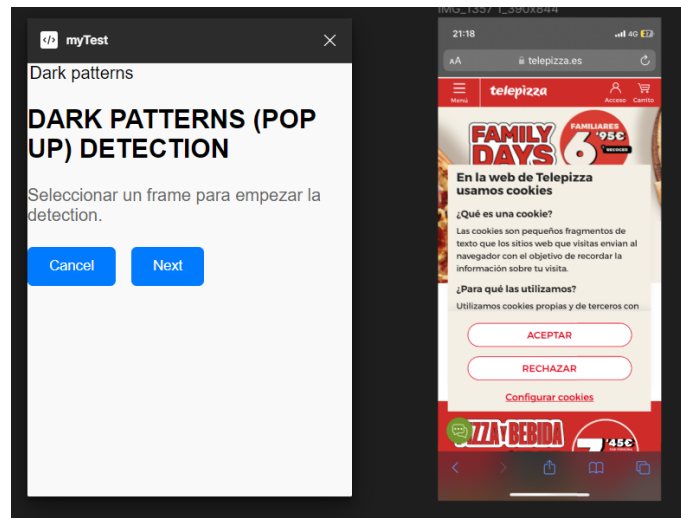


Figura 3.20: Imagen de una prueba real realizada

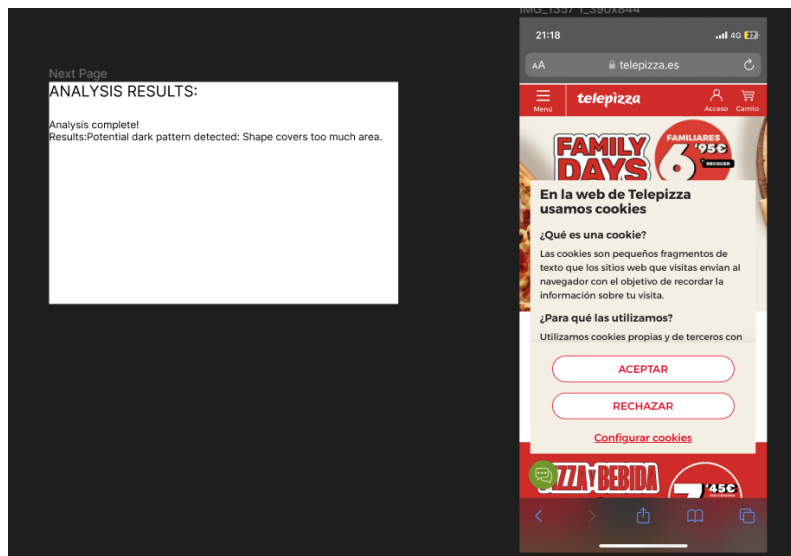


Figura 3.21: Imagen de los resultados de la prueba real realizada

3.10.3. Ventajas de las Diferentes Pruebas

El enfoque iterativo de pruebas y desarrollo ayudó a refinar las capacidades del plugin y también permitió la identificación y corrección de errores en etapas tempranas. A través de cada iteración, se recogieron comentarios y se realizaron ajustes necesarios para mejorar la funcionalidad y la usabilidad del plugin.

En resumen, el proceso de pruebas y desarrollo iterativo fue crucial para garantizar que el plugin cumpliera con sus objetivos de detectar y analizar patrones oscuros en interfaces de usuario de manera efectiva. Este enfoque metódico permitió mejorar continuamente el plugin y asegurar su fiabilidad y precisión en entornos de uso real.

3.11. Detección Ampliada de Elementos

Como se dijo anteriormente, este plugin fue diseñado principalmente para detectar pop-ups en las pantallas de los usuarios, sin embargo logro identificar un tipo de patrón oscuro adicional.

Durante el desarrollo y las pruebas del plugin, se observó que, además de detectar los patrones oscuros planeados a analizar, el algoritmo implementado también era capaz de identificar elementos como cuadros emergentes que aparecen en la parte inferior de las interfaces, como se puede ver en la figura 3.22. Estos elementos, aunque no formaban parte del objetivo inicial del plugin, resultaron ser detectables debido a ciertas características comunes que comparten con los patrones oscuros, como su disposición visual distintiva, el hecho de cubrir ciertas partes del texto, el texto completo o el hecho de ocupar partes importantes de la pantalla.



Figura 3.22: Ejemplo de un cuadro emergente que aparecen en la parte inferior

Esta capacidad adicional de detectar elementos emergentes puede resultar beneficiosa en la identificación de elementos intrusos en las interfaces de usuario, lo que podría contribuir a mejorar la experiencia del usuario al eliminar distracciones no deseadas y mejorar la accesibilidad de las aplicaciones y sitios web.

Esta capacidad para detectar elementos adicionales más allá de los patrones oscuros originales amplía su utilidad y potencial impacto en la mejora de la calidad y usabilidad de las interfaces de usuario digitales.

3.11. Detección Ampliada de Elementos



Figura 3.17: Diagrama de relación

Capítulo 4

Resultados

En este apartado se hablara de los objetivos y de los resultados obtenidos. A lo largo del desarrollo, se han realizado diversas pruebas utilizando un conjunto de imágenes para evaluar la eficacia del plugin en la detección de los pop-ups y adicionalmente de los cuadros emergente que aparecen por la parte inferior.

Hubo un total de treinta y ocho pruebas, evaluando la capacidad del plugin para detectar correctamente los patrones oscuros. Se decidió clasificar los resultados en varias categorías. Cuando el plugin consiguió detectar correctamente el patrón oscuro o cuando no había patrón oscuro y no detecto ninguno, se clasificaba ese resultado como "correcto". Si el plugin detectaba un patrón oscuro donde no había uno, el resultado se clasificaba como "falso positivo", en la figura 4.4 se puede ver un ejemplo de este tipo. Algunos resultados se clasificaban como "falso negativo" cuando el plugin no detecto un patrón oscuro cuando si había uno, se puede observar un ejemplo de este caso en la figura 4.3. La ultima categoría "lógica", es cuando la imagen que se analiza tiene un pequeño pop-up informativo, como por ejemplo el numero de la reserva o que no existe un producto que se quiere buscar. La lógica del plugin es que si el cuadro del pop-up es mas pequeño que el cuarto del tamaño del frame, entonces no se considera un patrón oscuro, y estos pop-ups informativos no suelen ocupar una gran porción de la pantalla, por lo que se detecto correctamente pero por la lógica del plugin, no por otra razón. En la figura 4.5, se puede ver uno de estos ejemplos de resultados de tipo lógica.

En cuanto a los resultados que se esperaban obtener, se había previsto lograr un 80% de resultados correctos ya que a la hora de realizar las pruebas de prototipo se conseguía identificar la gran mayoría de ellas. Se esperaba un 7% de falsos positivos y un 13% de falsos negativos, la razón por la cual se había previste en total un 20% de falsos resultados es debido a la lógica del plugin y al contexto de la interfaz que se quiere analizar. No se había previsto obtener ningún resultado de tipo "lógica", debido a su dependencia del contexto específico de las imágenes analizadas.

A continuación se verán los porcentajes de los tipos de resultado de las pruebas. Se logro conseguir un 84,21% de resultados de tipo correcto, es decir que el

Capítulo 4. Resultados

plugin detecto correctamente la presencia de un patrón oscuro. Para los falsos positivos se obtuvieron un 5,26%, igual para los falsos negativos y para los de tipo "lógica". En la figura 4.2 se puede observar un diagrama de barras para visualizar mejor los resultados obtenidos, y en la figura 4.1 se muestra una matriz de confusión que compara los resultados reales con los previstos.

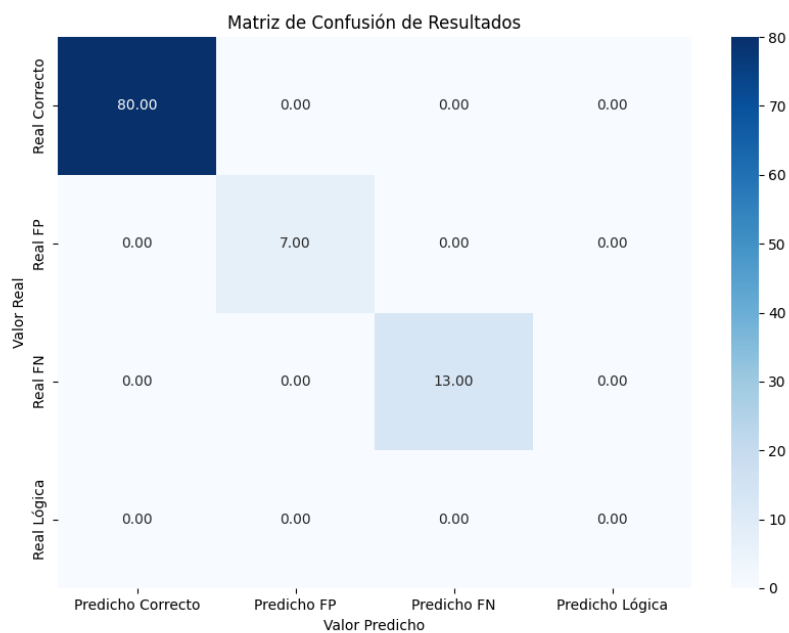


Figura 4.1: Matriz de confusión

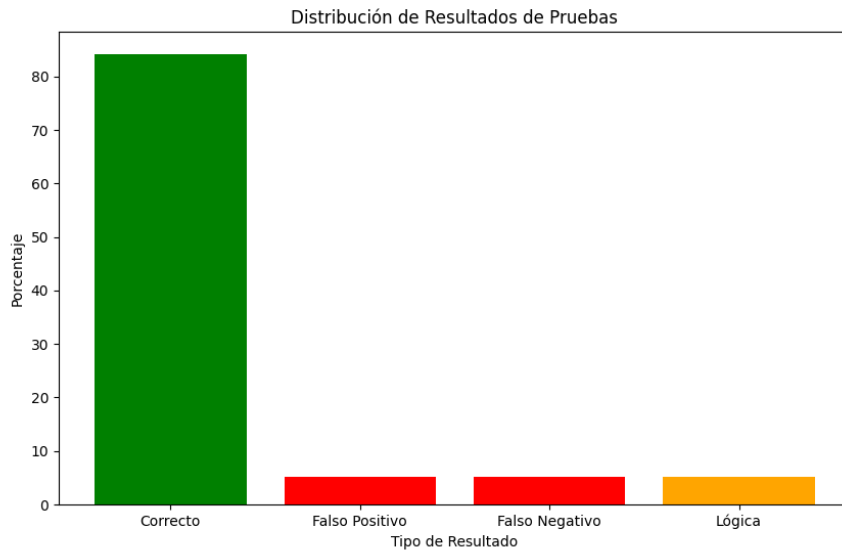


Figura 4.2: Diagrama de barras de los porcentajes de los resultados

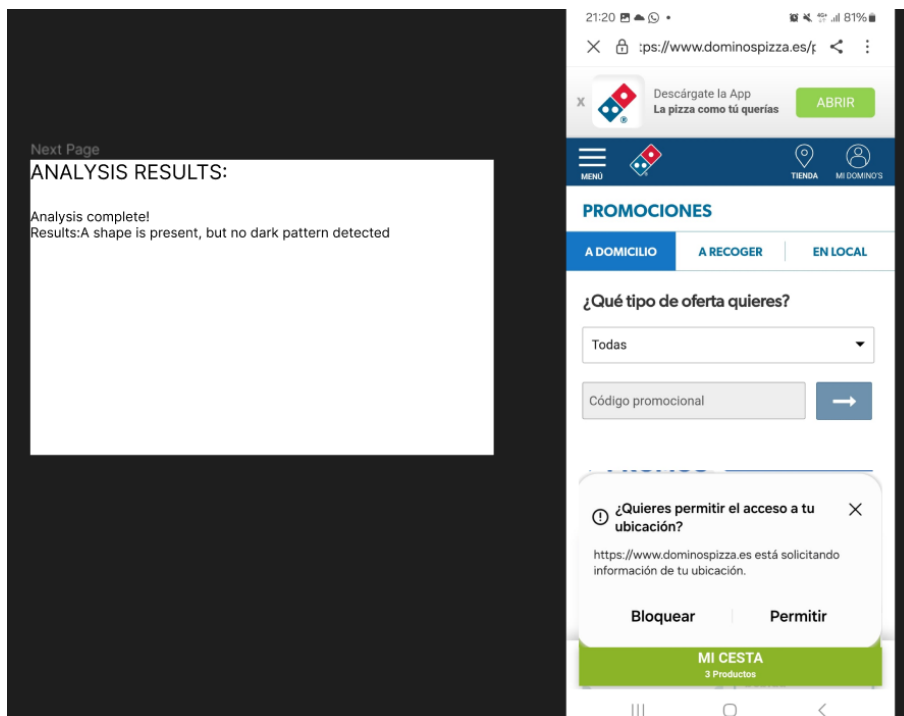


Figura 4.3: Ejemplo de un falso negativo

Capítulo 4. Resultados

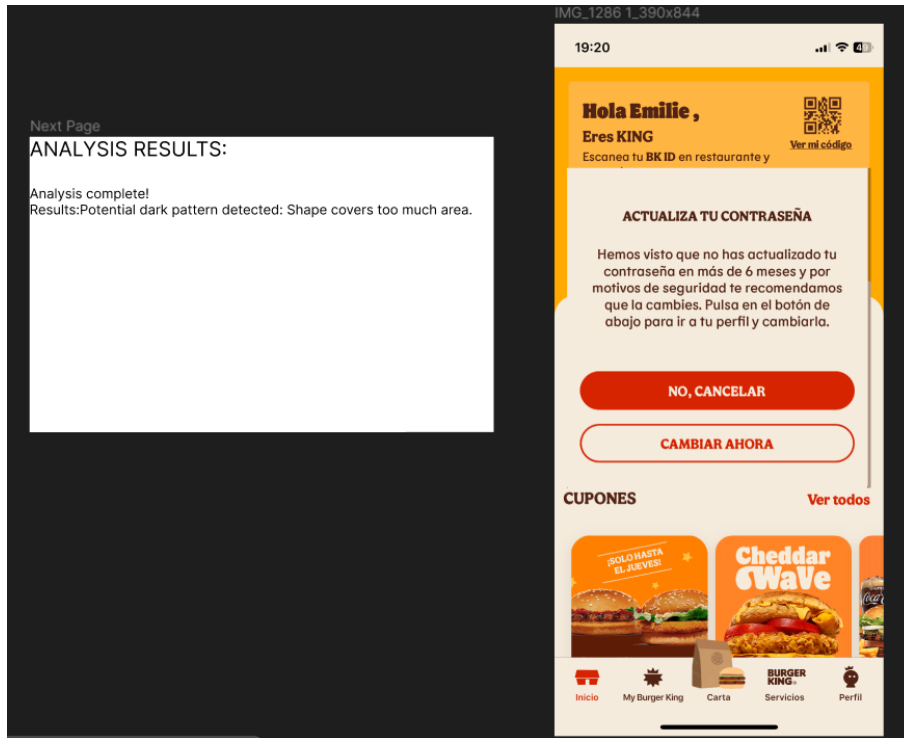


Figura 4.4: Ejemplo de un falso positivo

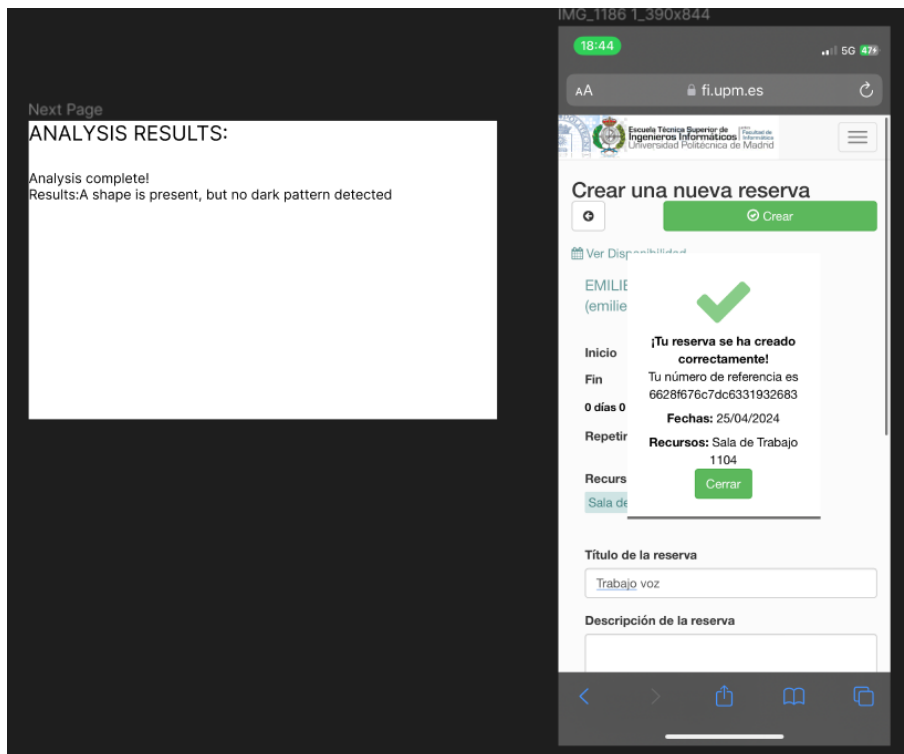


Figura 4.5: Ejemplo de un resultado de tipo lógica

4.1. Evaluación de Resultados

Tras estudiar los resultados obtenidos y los resultados previstos, se puede observar que se supero de un 4,21 % las expectativas previstas sobre los resultados de tipo correcto, se alcanzo una tasa de acierto de 84,21 % frente a al 80 % esperado. Por lo que indica una alta efectividad de la herramienta en identificar patrones oscuros. Sin embargo, a cerca de los falsos positivos y falsos negativos se puede observar una diferencia mas importante entre los que se habían previstos. Para los falsos positivos se había previsto un 7 % y se obtuvo un 5,26 %, lo que sugiere una ligera mejora de lo previsto. Los falsos negativos tiene más diferencia entre lo previsto y lo obtenido, el objetivo era un 13 % pero finalmente se obtuvo un 5,26%. La mayor diferencia se debe a que se pensó que la lógica del plugin era más factible a detectar patrones oscuros donde realmente no había, debido al contexto y a los tamaños reducidos de los pop-ups. Esta diferencia indica una mejor capacidad de detección que lo esperado, pero sigue siendo casos donde el plugin no detecta patrones oscuros presentes, lo que podría llevar a que estos pasen desapercibidos para los usuarios y los diseñadores UX. Al reflexionar sobre los resultados se tuvo que crear un resultado clasificado como "lógica", que son cuando se ha detectado correctamente que la imagen no contiene un patrón oscuro debido al contexto del pop-up. La ventana emergente en este caso es simplemente informativa pero el plugin no lo detecto por el contexto sino por su lógica, que no considera que un pop-up más pequeño que el cuarto del frame sea un patrón oscuro. Este tipo de resultado dio un 5,26 %, lo que destaca la importancia de considerar todos los posibles contextos en los que se puedan presentar los patrones oscuros.

En resumen, el plugin muestra una buena capacidad para detectar patrones oscuros de tipo pop-up, sin embargo existe margen para mejorar su precisión. La reducción de falsos positivos y falsos negativos, así como la consideración de resultados contextuales adicionales, podrían lograrse ajustando criterios de detección, añadiendo otros criterios no definidos y mejorando el algoritmo desarrollado. Estos cambios podrían ayudar a aumentar la eficacia del plugin y a garantizar que los usuarios y los diseñadores UX se salten patrones oscuros ni se enfrenten a falsas alarmas.

Capítulo 5

Discusión

En este apartado se discuten los resultados obtenidos, comparándolos con el estado del arte y evaluando la eficacia de la solución propuesta.

5.1. Discusión sobre los Resultados Obtenidos

El plugin ha demostrado una buena capacidad para detectar patrones oscuros, sin embargo existen áreas para mejorar su precisión. La reducción de falsos positivos y falsos negativos, así como la consideración de resultados contextuales adicionales, podrían lograrse mejorando el algoritmo y añadiendo ciertos criterios. Es esencial considerar todos los posibles contextos en los que pueden presentarse los patrones oscuros para garantizar una detección más precisa y evitar errores.

Es importante considerar que podría existir sesgos en los resultados debido al tipo y cantidad de ejemplos utilizados. Aunque se realizaron treinta y ocho pruebas en diferentes interfaces reales, este número podría no ser suficiente para capturar toda la diversidad de contextos y patrones oscuros posibles. Es primordial aumentar el número de ejemplos pero sobre todo los que podrían llevar a potenciales errores, es decir incluir diferentes pruebas para garantizar que el algoritmo funcione bien en diferentes situaciones. Probar el plugin con más ejemplos, especialmente aquellos que presenten contextos variados, ayudaría a identificar y corregir falsos positivos, falsos negativos y resultados basados en la lógica del plugin.

Por otra parte, el plugin ha mostrado ser efectivo independientemente del idioma utilizado en las pruebas realizadas. El algoritmo se basa en la detección de elementos visuales y patrones en la interfaz, más que en el contenido textual específico. Esto permite que el plugin funcione en cualquier idioma, ya que detecta patrones oscuros basándose en la presencia y disposición de elementos visuales que se pueden encontrar en las interfaces.

5.2. Comparación con el Estado del Arte

En estado del arte se dice que los patrones oscuros son un desafío persistente en el diseño de interfaces de usuario, y pocos enfoques se han desarrollado para su detección. Una de las herramientas que se encontró en este campo es UIGuard, que utiliza técnicas de visión por computadora y procesamiento de lenguaje natural para identificar patrones oscuros, alcanzando una precisión del 93%. Sin embargo, UIGuard presenta limitaciones significativas, como la dependencia de modelos de inteligencia artificial que pueden no tener en cuenta el contexto de la situación.

El plugin desarrollado en este Trabajo de Fin de Grado se diferencia de UIGuard al no depender de la inteligencia artificial, sino de un enfoque heurístico y basado en el juicio humano. Este método ofrece varias ventajas clave, incluyendo una mayor transparencia, facilidad de implementación y mantenimiento. La diferencia más relevante entre la herramienta desarrollada y UIGuard es el contexto en el que se ejecuta. El plugin se utiliza dentro de Figma, lo que puede proporcionar un valor añadido para que los diseñadores UX identifiquen en sus propios diseños posibles patrones oscuros no intencionados en fases previas a la implementación. Mientras que UIGuard, evalúa los patrones oscuros sobre el producto final, con lo que los costes de tiempo, recursos y dinero son significativamente mayores al tener que modificar un producto ya terminado.

5.3. Eficacia de la Solución Propuesta

La solución desarrollada en este TFG demuestra una alta eficacia en la detección de patrones oscuros en interfaces de usuario. A pesar de no alcanzar la precisión del 93% de UIGuard, el enfoque heurístico y basado en el juicio humano presenta varias ventajas como la transparencia y comprensibilidad, es decir que los usuarios pueden entender mejor cómo y por qué se detectan ciertos patrones oscuros, lo que mejora la confianza en la herramienta. La facilidad de implementación y mantenimiento también puede ser una buena ventaja ya que al no depender de modelos complejos de inteligencia artificial, el plugin es más sencillo de mantener y actualizar. Otra ventaja es la adaptabilidad, el plugin puede ajustarse rápidamente a diferentes contextos y tipos de interfaces sin necesidad de volver a entrenar modelos.

Este enfoque puede resultar útil y fácil de manipular para los usuarios de Figma, aunque puede no ser tan rápido como la inteligencia artificial, esta herramienta desarrollada ofrece la ventaja de permitir la modificación previa de los datos de entrada. Esto hace que el usuario del plugin se haya familiarizado con la imagen y puede ser más consciente sobre el contexto, lo cual puede hacer que se debaten los resultados obtenidos y determinar si son coherentes. A diferencia de la inteligencia artificial, que puede ofrecer una respuesta directa que podría aceptarse sin cuestionar. El enfoque basado en el juicio humano fomenta una evaluación crítica y detallada de los resultados. Por tanto, aunque la inteligencia artificial puede proporcionar resultados más rápidos, este método permite desarrollar un pensamiento crítico y una mejor comprensión, lo que puede ser más

valioso en ciertos escenarios.

5.4. Conclusiones

En conclusión, la herramienta desarrollada en este trabajo muestra ser eficaz en la detección de patrones oscuros, alcanzando un 84,21 % de resultados correctos. El plugin ofrece una solución útil que tiene la ventaja de ser más transparente, fácil de mantener y adaptable a diferentes contextos. Aunque no es tan rápida como la inteligencia artificial, su enfoque basado en el juicio humano permite una evaluación más crítica y detallada de los resultados, fomentando una mayor comprensión y personalización. Sin embargo, para aumentar aún más su eficacia y reducir la incidencia de errores, es necesario continuar ajustando los criterios de detección y mejorar la lógica del plugin. Con estos ajustes, el plugin puede convertirse en una herramienta valiosa para diseñadores de UX y desarrolladores, ayudándolos a crear interfaces más intuitivas y libres de patrones oscuros.

Capítulo 6

Análisis del Impacto

La herramienta desarrollada a lo largo del trabajo, que detecta los patrones oscuros de tipo pop-ups, puede generar un impacto varios contexto, como el impacto personal o el impacto cultural... A continuación, se verán los diferentes efectos posibles en cada contexto que sean beneficiosos o adversos.

El impacto a nivel personal se centra en mejorar la experiencia del desarrollador UX al diseñar interfaces digitales. Los usuarios que utilicen el plugin incrementarán su capacidad para identificar patrones oscuros, lo cual contribuirá a su formación profesional y a una mejor comprensión del diseño ético de interfaces, implicando a una mejora de su consciencia sobre el tema. No obstante, el uso del plugin podría llevar a una sobrecarga cognitiva, la necesidad de evaluar manualmente los patrones oscuros podría resultar en una carga adicional para los usuarios, y les consumaría bastante tiempo.

En el ámbito empresarial, el plugin puede tener un impacto significativo en la reputación de la marca y la retención de clientes. Al usar la herramienta, se garantizara una experiencia de usuario más positiva, las empresas pueden aumentar la fidelidad de los clientes y mejorar su imagen de marca. Además, al identificar y corregir patrones oscuros en sus propias interfaces, las empresas pueden evitar posibles problemas legales y regulaciones sobre prácticas desleales. Por otro lado, la implementación de la herramienta en sus entornos, y su uso podría llevar a un aumento de tiempo en las tareas, lo cual es crítico en entornos de trabajo con tiempos ajustados.

En cuanto al impacto económico, el plugin podría mejorar la satisfacción y lealtad del cliente, lo que hará que a largo plazo se mejore la eficiencia en las transacciones en línea y una reducción en las tasas de abandono de carrito, implicando beneficios económicos. Sin embargo, a corto plazo podría resultar en una pérdida de dinero para las empresas. Las razones por lo cual podría ocurrir, es que al eliminar los patrones oscuros en sus interfaces los usuarios no llegarían a ser manipulados y probablemente no acepten una oferta que si hubieran aceptado si hubiera un patrón oscuros. Además, es posible que no acepten de activar las notificaciones, o que no acepten las políticas de privacidad involuntariamente, lo cual impediría que la empresa pueda realizar publicidad personalizada al

Capítulo 6. Análisis del Impacto

usuario. Todo eso impactaría temporalmente los ingresos.

En el ámbito cultural, el plugin impulsa un cambio cultural hacia prácticas de diseño más éticas y transparentes, influyendo positivamente en las normas culturales del diseño digital. También promueve la creación de interfaces más inclusivas y accesibles, teniendo en cuenta la diversidad de los usuarios.

Considerado los Objetivos de Desarrollo Sostenible (ODS), la implementación del plugin puede contribuir a varios objetivos, como el ODS 8 titulado "Trabajo decente y crecimiento económico", donde el plugin fomenta prácticas empresariales más éticas, contribuyendo a largo plazo a un crecimiento económico inclusivo y sostenible. Otro objetivo podría ser el ODS 9 llamado "Industria, innovación e infraestructura", ya que el plugin promueve la innovación en el diseño de interfaces y la infraestructura digital, apoyando el desarrollo de tecnologías más éticas. El uso del plugin elimina los patrones oscuros y consciencia a los usuarios sobre aquellos, promoviendo un consumo responsable y ético, alineando las prácticas de diseño con un uso sostenible de los recursos, lo que contribuye al ODS 12 titulado "Producción y consumo responsables".

En resumen, el desarrollo del plugin tiene un impacto significativo en varios contextos, desde mejorar la experiencia del usuario hasta promover la transparencia y la ética en el diseño de productos digitales. Sin embargo, es importante considerar cuidadosamente cualquier efecto adverso potencial y tomar decisiones informadas.

Capítulo 7

Conclusiones y Trabajo Futuro

7.1. Conclusiones

Este trabajo tenía como objetivo desarrollar un plugin en Figma para la detección de patrones oscuros en interfaces de usuario utilizando un método heurístico. Se quería ayudar con esta herramienta a los desarrolladores UX de identificarlos de manera automática, y además sensibilizar los usuarios del plugin sobre la presencia de patrones oscuros.

Sobre los objetivos planteados y los que se han logrado, se consiguió desarrollar el prototipo y la herramienta para la detección de pop-ups en las interfaces de usuario. El plugin también cumple el segundo objetivo de ser capaz de proporcionar detalles útiles para la eliminación de patrones oscuros, luego se realizó también el objetivo de probarlo en interfaces reales que se iban encontrando en el día a día. Para acabar, el último objetivo de compartir la herramienta para concienciar a las personas sobre los patrones oscuros y mejorar la experiencia de los usuarios, aun no se ha cumplido pero se espera poder realizarlo.

A cerca de los resultados obtenidos, para recapitular, se realizaron un total de treinta y ocho pruebas para evaluar el plugin en la detección de los pop-ups y adicionalmente los cuadros emergentes desde la parte inferior. De todas estas pruebas se logró un 84,21 % de aciertos, este porcentaje superó las expectativas que era de un 80% lo que sugiere una buena efectividad de la herramienta. No hay que olvidar los falsos positivos, falsos negativos y los de "lógica", que todos juntos representan aproximadamente un 15% de los resultados, lo que demuestra la importancia del contexto en todas las situaciones.

En resumen, la herramienta desarrollada alcanza las expectativas, tiene una buena efectividad para la detección de patrones oscuros y cumple los principales objetivos, excepto el último que se espera que se cumpla.

7.2. Trabajo Futuro

Aunque el plugin desarrollado ha mostrado resultados positivos, hay algunas áreas que se podrían mejorar para enriquecer su funcionalidad.

Para poder mejorar la herramienta, es esencial seguir refinando el algoritmo para reducir los falsos positivos y los falsos negativos. Ajustar ciertos criterios o implementar nuevos, pudiendo probarlos con una mayor variedad de pruebas. Incorporar más contextos específicos para la detección de patrones oscuros, para que el plugin pueda identificarlos en una variedad más amplia de situaciones. Sería crucial después de haber implementado las mejoras, realizar pruebas adicionales en una variedad de contextos y con diferentes tipos de interfaces para comprobar si lo implementado funciona y ver si su eficacia mejora. Al realizar todas estas pruebas será necesario mantener el repositorio de imágenes ya creado, añadiendo las nuevas imágenes probadas, lo cual ayudara a los próximos usuarios de la herramienta.

Uno de los trabajos futuros, sería ampliar la funcionalidad del plugin para detectar no solo pop-ups, sino también otros patrones oscuros como “Botones de colores apagados”, “precios ocultos” y “opciones predeterminadas engañosas”. Es decir, expandir la herramienta para la detección de más tipos de patrones oscuros.

En conclusión, aunque el plugin ha demostrado ser una herramienta eficaz para detectar patrones oscuros, existen muchas mejoras realizables. Seguir trabajando sobre ellas es importante para seguir desarrollando una herramienta más poderosa y versátil, ayudando a crear interfaces de usuario más transparentes y éticas.

Bibliografía

- [1] R. MOODY. «Estadísticas sobre el tiempo de pantalla: tiempo de pantalla medio en Estados Unidos y el resto del mundo». (2024), dirección: <https://www.comparitech.com/es/transmisiones-de-video/estadisticas-tiempo-pantalla/#:~:text=A%20nivel%20mundial%2C%20un%20usuario,de%20la%20de%20su%20m%C3%B3vil>. (visitado 02-06-2024).
- [2] C. M. Gray, Y. Kou, B. Battles, J. Hoggatt y A. L. Toombs. «The Dark (Patterns) Side of UX Design». (2018), dirección: <https://dl.acm.org/doi/pdf/10.1145/3173574.3174108> (visitado 19-04-2024).
- [3] L. D. Geronimo, L. Braz, E. Fregnan, F. Palomba y A. Bacchelli. «UI Dark Patterns and Where to Find Them: A Study on Mobile Applications and User Perception». (2020), dirección: <https://dl.acm.org/doi/pdf/10.1145/3313831.3376600> (visitado 19-04-2024).
- [4] M. V. Suárez. «La protección frente al uso de patrones oscuros en la Experiencia de Usuario en la Unión Europea». (2020), dirección: https://www.institutopascualmadoz.es/wp-content/uploads/2016/06/252708_MARIA_VARELA_SUAREZ_TFM_MD_Teleco_-_Maria_Varela_2491027_472412411-1.pdf (visitado 29-05-2024).
- [5] A. Mitterhauser. «Ethicaly». (2024), dirección: <https://www.figma.com/community/widget/1271398165783851031> (visitado 26-05-2024).
- [6] D. Destefanis. «Design Lint». (2023), dirección: <https://www.figma.com/community/plugin/801195587640428208/design-lint> (visitado 20-04-2024).
- [7] J. Chen, J. Sun, S. Feng et al. «Unveiling the Tricks: Automated Detection of Dark Patterns in Mobile Applications». (2023), dirección: <https://arxiv.org/pdf/2308.05898> (visitado 26-05-2024).
- [8] N. N. G. (Kaley). «Popups: 10 Problematic Trends and Alternatives». (2019), dirección: <https://www.nngroup.com/articles/popups/> (visitado 20-05-2024).
- [9] KoA. «Image Resizer Plugin». (2023), dirección: <https://www.figma.com/community/plugin/1218937991051926218/image-resizer-plugin> (visitado 20-05-2024).
- [10] K. Koch. «Node»frame». (2022), dirección: <https://www.figma.com/community/plugin/952557548511056612/nodeframe?searchSessionId=luk64su0-ot0kl3se5vr> (visitado 20-05-2024).

Anexos

Apéndice A

Primer anexo

Memoria_TFG (3).pdf

ORIGINALITY REPORT

8%

SIMILARITY INDEX

6%

INTERNET SOURCES

2%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universidad Politécnica de Madrid Student Paper	1%
2	oa.upm.es Internet Source	1%
3	Submitted to Pontificia Universidad Catolica de Chile Student Paper	<1%
4	Submitted to University of Brighton Student Paper	<1%
5	moam.info Internet Source	<1%
6	Submitted to Universidad Internacional de la Rioja Student Paper	<1%
7	Submitted to University of Warwick Student Paper	<1%
8	eprints.ucm.es Internet Source	<1%

9	Submitted to UNIV DE LAS AMERICAS Student Paper	<1 %
10	www.coursehero.com Internet Source	<1 %
11	hdl.handle.net Internet Source	<1 %
12	ichi.pro Internet Source	<1 %
13	Submitted to Queensland University of Technology Student Paper	<1 %
14	siliconcanals.com Internet Source	<1 %
15	www.sernac.cl Internet Source	<1 %
16	Submitted to Curtin University of Technology Student Paper	<1 %
17	Submitted to Universitat Politècnica de València Student Paper	<1 %
18	edoc.pub Internet Source	<1 %
19	Submitted to Escuela Superior Politécnica del Litoral Student Paper	<1 %

48 www.revistafarmaciahospitalaria.es <1 %
Internet Source

49 www.tatsachen-ueber-deutschland.de <1 %
Internet Source

50 J.C. Rajapakse, J. Piyaratna. "Bayesian approach to segmentation of statistical parametric maps", IEEE Transactions on Biomedical Engineering, 2001 <1 %
Publication

51 www.dropbox.com <1 %
Internet Source

52 translate.evernote.com <1 %
Internet Source

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off

40 Juan Manuel Campayo Esteban. "Indicadores de calidad en imágenes digitales en programas de control de calidad en mamografía", Universitat Politecnica de Valencia, 2005 <1 %
Publication

41 Segundo Castro-González, Jesús Peña-Vinces, Alex J. Ruiz-Torres, Juan Carlos Sosa. "Estudio intrapaíses de la competitividad global desde el enfoque del doble diamante para Puerto Rico, Costa Rica y Singapur", Investigaciones Europeas de Dirección y Economía de la Empresa, 2014 <1 %
Publication

42 digibug.ugr.es <1 %
Internet Source

43 dspace.unl.edu.ec <1 %
Internet Source

44 giig.ugr.es <1 %
Internet Source

45 windows-defender.softonic.com <1 %
Internet Source

46 www.fepasde.org.co <1 %
Internet Source

47 www.perfil.com <1 %
Internet Source

30

Internet Source

<1 %

31

docshare.tips

Internet Source

<1 %

32

listens.online

Internet Source

<1 %

33

panchoriveros.blogspot.com

Internet Source

<1 %

34

patents.google.com

Internet Source

<1 %

35

repositorio.ucsg.edu.ec

Internet Source

<1 %

36

www.vergantmagazine.com

Internet Source

<1 %

37

1library.co

Internet Source

<1 %

38

David Griol Barres. "Desarrollo y evaluación de diferentes metodologías para la gestión automática del diálogo", Universitat Politecnica de Valencia, 2007

Publication

<1 %

39

Francisco Rodríguez Ballester. "Detección concurrente de errores en el flujo de ejecución de un procesador.", Universitat Politecnica de Valencia, 2016

Publication

<1 %

20 theses.cz Internet Source <1 %

21 biblioteca.inap.gob.ar Internet Source <1 %

22 www.bcnpc.com Internet Source <1 %

23 Walter Escajadillo Chimayco. "La productividad laboral y la competitividad logística de la economía peruana en relación los otros países de la región (2010-2022)", Ciencia y Práctica, 2023
Publication <1 %

24 bonga.unisimon.edu.co Internet Source <1 %

25 catarina.udlap.mx Internet Source <1 %

26 openaccess.uoc.edu Internet Source <1 %


27 pure.strath.ac.uk Internet Source <1 %

28 support.udemy.com Internet Source <1 %

29 www.ebizlatam.com Internet Source <1 %

www.idemsoft.com

Este documento esta firmado por

	Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=ETS Ingenieros Informaticos - UPM, C=ES
	Fecha/Hora	Mon Jun 03 20:54:44 CEST 2024
	Emisor del Certificado	EMAILADDRESS=camanager@etsiinf.upm.es, CN=CA ETS Ingenieros Informaticos, O=ETS Ingenieros Informaticos - UPM, C=ES
	Numero de Serie	561
	Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)