

Treball de Fi de Grau

Títol

IMPLEMENTACIÓ DE LA TÈCNICA SCRAPPING PER
AUTOMATITZAR L'ENVIAMENT D'OFERTES EN EL
MODEL DE MÀRQUETING D'AFILIACIÓ

Autoria

PAU CARDÚS FILLAT

Professorat tutor

ADRIÀ PADILLA MOLINA
CELINA NAVARRO BOSCH

Grau

Comunicació Audiovisual	
Periodisme	
Publicitat i Relacions Públiques	
Comunicació Interactiva	X
Comunicació de les Organitzacions	

Tipus de TFG

Projecte	X
Recerca	

Data

Del 3 al 7 de juny de 2024	X
----------------------------	---

Full resum del TFG

Títol del Treball Fi de Grau:

Català:	IMPLEMENTACIÓ DE LA TÈCNICA SCRAPPING PER AUTOMATITZAR L'ENVIAMENT D'OFERTES EN EL MODEL DE MÀRQUETING D'AFILIACIÓ			
Castellà:	IMPLEMENTACIÓN DE LA TÉCNICA SCRAPPING PARA AUTOMATIZAR EL ENVÍO DE OFERTAS EN EL MODELO DE MARKETING DE AFILIACIÓN			
Anglès:	IMPLEMENTATION OF THE SCRAPPING TECHNIQUE TO AUTOMATE THE SENDING OF OFFERS IN THE AFFILIATE MARKETING MODEL			
Autoria:		PAU CARDÚS FILLAT		
Professorat tutor:		ADRIÀ PADILLA MOLINA CELINA NAVARRO BOSCH		
Curs:	2023/24	Grau:	Comunicació Audiovisual	
			Periodisme	
			Publicitat i Relacions Públiques	
			Comunicació Interactiva	X
			Comunicació de les Organitzacions	

Paraules clau (mínim 3)

Català:	SCRAPPING, PYTHON, PHP, MÀRQUETING D'AFILIACIÓ, AUTOMATITZACIÓ
Castellà:	SCRAPPING, PYTHON, PHP, MARKETING DE AFILIACIÓN, AUTOMATIZACIÓN
Anglès:	SCRAPPING, PYTHON, PHP, AFFILIATION MARKETING, AUTOMATION

Resum del Treball Fi de Grau (extensió màxima 100 paraules)

Català:	Creació d'un sistema de codis amb Python i PHP per automatitzar l'enviament d'ofertes d'Amazon a Telegram, fent servir la tècnica Scrapping. Posterior posada en producció i analítica de dades per millorar els resultats. Tot això és un experiment per comprovar si implementar aquest model de negoci és tan viable i rendible com es vol fer creure per internet.
Castellà:	Creación de un sistema de códigos con Python y PHP para automatizar el envío de ofertas de Amazon a Telegram, utilizando la técnica de Scrapping. Posterior puesta en marcha y análisis de datos para mejorar los resultados. Todo esto es un experimento para verificar si implementar este modelo de negocio es tan viable y rentable como se quiere hacer creer en internet.
Anglès:	Creation of a code system with Python and PHP to automate the sending of Amazon offers to Telegram, using the Scrapping technique. Deployment and data analytics to improve results. All of this is an experiment to verify if implementing this business model is as viable and profitable as shown on the internet.

IMPLEMENTACIÓ DE LA TÈCNICA *SCRAPPING* PER AUTOMATITZAR L'ENVIAMENT D'OFERTES EN EL MODEL DE MÀRQUETING D'AFILIACIÓ

Pau Cardús Fillat

Tutors: Adrià Padilla Molina i Celina Navarro Bosch

Universitat Autònoma de Barcelona, 2024

SUMARI DEL TREBALL

1 INTRODUCCIÓ	6
1.1 Precedents	6
1.2 Justificació del treball	10
1.3 Contingut del treball	10
2 MARC TEÒRIC I TÈCNIC	12
2.1 Màrqueting d'afiliació	12
2.2 Python	13
2.3 <i>Scrapping</i>	14
2.4 Servidor Cloud VPS	15
2.5 API	15
2.6 PHP	16
2.7 Base de dades	16
3 OBJECTIUS DEL TREBALL	17
4 PRESENTACIÓ FORMAL DEL PROJECTE	18
4.1 Negoci	18
4.1.1 Per què decidim automatitzar? Context social	18
4.1.2 Ètica de l'automatització	19
4.2 Explicació dels codis	21
4.2.1 Obtenció d'ofertes	23
4.2.2 Processament de dades	25
4.2.3 Enviament d'ofertes	26
4.2.4 Obtenció de dades	26

4.2.5 L'execució dels codis	27
4.2.6 Bases de dades.....	29
4.3 Desplegament tècnic	31
4.4 Explotació de dades i analítica	41
5 POSADA EN PRODUCCIÓ	42
5.1 Primers mesos en producció	42
5.2 Procés d'optimització	44
5.3 Dificultats en la posada en producció	45
6 MILLORES A FUTUR.....	46
7 CONCLUSIONS.....	47
8 WEBGRAFIA I BIBLIOGRAFIA.....	49
9 ANNEXOS	51

SUMARI DE FIGURES

Figura 1. Paràmetres UTM. Elaboració pròpia (2023)	13
Figura 2. Esquema de l'objectiu del projecte. Elaboració pròpia (2023)	21
Figura 3. Esquema de l'objectiu del projecte. Elaboració pròpia (2023)	22
Figura 4. Esquema de l'objectiu del projecte. Elaboració pròpia (2023)	22
Figura 5. Estructura crontab. Elaboració pròpia (2024)	28
Figura 6. Estructura crontab. Elaboració pròpia (2024)	29
Figura 7. Esquema del primer codi de Python. Elaboració pròpia (2024)	31
Figura 8. Captura de pantalla de la pàgina d'ofertes d'Amazon.es. Elaboració pròpia (2024)	33
Figura 9. Paràmetres d'un enllaç de producte d'Amazon. Elaboració pròpia (2024)	33
Figura 10. Captura de pantalla de la pàgina de detalls d'un producte d'Amazon. Elaboració pròpia (2024)	34
Figura 11. Captura de pantalla de Google Traductor. Elaboració pròpia (2024)	35
Figura 12. Esquema del segon codi de Python. Elaboració pròpia (2024)	36
Figura 13. Esquema del tercer codi de Python. Elaboració pròpia (2024)	38
Figura 14. Enllaç d'afiliació. Elaboració pròpia (2024)	40
Figura 15. Embut de conversió del model de negoci. Elaboració pròpia (2024)	42

SUMARI DE TAULES

Taula 1. Columnes de la taula d'ofertes de la base de dades. Elaboració pròpia (2024).....	29
Taula 2. Columnes de la taula de clics de la base de dades. Elaboració pròpia (2024).....	30

1 INTRODUCCIÓ

1.1 Precedents

Les xarxes socials han amplificat la idea que guanyar diners a través d'Internet és fàcil i senzill, que pots crear un negoci en línia sense coneixements amb el qual puguis viure. Així ho vaig demostrar, amb l'ajuda d'una companya de classe, en un treball de l'assignatura Analítica Digital el desembre de 2023, que s'adjunta als annexos.

L'interès ha pujat notablement en els últims anys i, consegüentment, la producció i publicació de vídeos a les plataformes digitals. Abans, el contingut publicat tenia la intenció d'ajudar als espectadors, però això ha canviat, i ara es publica pensant amb fer diners. Els canals que conformen aquest mercat plantegen la generació de diners d'una manera excessivament fàcil, captant les visualitzacions per part de molts usuaris.

Ara bé, quin interès tenen aquests canals per obtenir tantes visualitzacions? Com expliquem a l'estudi, la resposta recau en la generació d'ingressos, sobretot de dues fonts. La primera, directament des de YouTube, pagant el canal per cada visualització que tinguin els seus vídeos. La segona, amb els programes d'afiliació que paguen al canal si porten trànsit a una plataforma o aplicació.

Tot i que el “contingut brossa” ha aclaparat el mercat sobre guanyar diners a YouTube, no hem d'oblidar que encara hi ha vídeos de qualitat. Aquests vídeos mantenen l'objectiu principal: ajudar l'espectador a guanyar diners.

Em defineixo com a emprenedor i m'agrada intentar tots els projectes i models que em passen pel cap. I això, els algorismes de les xarxes socials ho saben a la perfecció. Per tant, ni que no m'agradin els continguts sobre guanyar diners, m'apareixen constantment com a recomanats.

Vaig començar a treballar com a autònom la primavera del 2021 perquè tenia un projecte que funcionava prou bé. Això va deixar de ser així el mateix octubre, i havia de prendre una decisió sobre la meua figura fiscal. No volia donar-me de baixa, perquè això significava que no podria continuar provant projectes còmodament. Per tant, vaig buscar, per primera vegada, maneres de guanyar diners a YouTube.

Vist des d'ara, vaig passar a ser un mer espectador que lucrava als altres sense saber-ho. En un dels molts vídeos que vaig veure, explicava per sobre el model d'afiliació d'Amazon. Era tan fàcil com recomanar productes i que es comprassin, amb només això ja començaves a facturar. Bé, "fàcil" d'aquella manera.

Com ja he explicat abans, sempre m'ha agradat provar projectes, gaudeixo molt el camí per dur-los a terme. Aquella mateixa nit ja havia creat el compte d'Amazon Afiliats i una pàgina web amb ofertes de Halloween. La idea era fer-la córrer per Facebook i Instagram pagant publicitat.

Aquest primer intent no va funcionar com esperava, i Halloween va passar sense haver obtingut cap venda associada. Ara bé, tampoc tenia gaires expectatives. Durant els següents mesos, vaig intentar-ho un parell de vegades més, aprofitant que era Nadal i tothom compra força. D'aquí en vaig aconseguir tres, de vendes associades. Va ser l'anhel que em feia falta per continuar amb aquest projecte.

Tothom es pot registrar com afiliat d'Amazon, però no et fan cap ingrés fins que has generat 3 vendes associades i t'han aprovat el compte. Jo, el primer requisit ja l'havia aconseguit, però no vaig passar el següent. Em van bloquejar el compte perquè "no proporcionava informació addicional més enllà de la que ofereix Amazon" a la meua pàgina web.

Aleshores, convençut que el projecte podia avançar favorablement, hi vaig centrar esforços. Tenia un pla d'acció amb l'estratègia de per on començaria i

com continuaria expandint el model a diferents sectors i països. Vaig decidir començar per casa, segmentar amb el públic català i construir una petita comunitat d'ofertes.

Com que no em decidia amb el nom, un bon amic em va recomanar que la comunitat es digués PREUS, amb el seu respectiu domini, preus.cat. Així que dit i fet, el vaig comprar per construir un espai d'ofertes actualitzades de diferents categories. Aquesta vegada, però, amb més “informació addicional”.

Tenia interès a tenir el compte d'afiliat aprovat, per una banda, per poder rebre els pagaments de les comissions i, per l'altra, per tenir accés a l'API d'Amazon, que em permetia obtenir informació de manera instantània de tots els productes. Així, podia automatitzar un sistema d'enviament d'ofertes sense haver-hi de dedicar hores de treball.

Havia de tornar a començar de nou, i no va ser tan fàcil, ni aconseguir les vendes ni que m'aproveïssin el compte. Quan en vaig aconseguir 3, Amazon va revisar la pàgina web i va decidir no aprovar el compte perquè la pàgina no estava en “castellà”, l'idioma oficial de la plataforma amazon.es. Vaig insistir i parlar amb responsables del programa d'afiliació i, tot i que compartien que el català era un idioma oficial i habitual dins de l'àmbit d'amazon.es, les seves polítiques no ho permetien. Com tampoc ho haguessin fet si la pàgina estigués en anglès.

En aquell moment em vaig frustrar força, ja ho tenia tot bastant estudiat i treballat, amb la pàgina web feta, i ho havia de tornar a canviar tot. També tenia l'Instagram de PREUS en marxa, on anava penjant contingut i els seguidors pujaven. Com que tenia més projectes paral·lels, vaig decidir deixar de banda PREUS, el març de 2022.

Durant l'estiu del 2023, vaig conèixer un noi, també emprenedor, que havia aconseguit força ingressos amb els programes d'afiliació d'Amazon i d'altres

plataformes. Va ser l'espurna que em va fer aixecar PREUS novament, més d'un any després.

La meua intenció, però, va ser prendre-m'ho amb calma, i provar si realment el model tenia potencial, o era més complicat del que pensava. Per tant, l'enfocament que li vaig donar va ser més experimental que no pas professional.

Sabent que tenia el TFG a la vista, vaig voler aprofitar-ho i unir-ho tot. Ara bé, encara em faltava concretar com ho volia dur a terme.

Un negoci centrat en el màrqueting d'afiliació té dos pilars fonamentals. La plataforma d'on ser afiliat i obtenir els ingressos, i l'espai on oferir aquestes recomanacions a un públic com més gran millor. Tanmateix, aquests dos pilars necessiten una connexió, i en aquest cas era la feina d'anar enviant missatges a la plataforma cada cert temps.

Havia de trobar la manera d'automatitzar l'enviament de productes recomanats sense comptar amb la disponibilitat de l'API, perquè abans m'havien d'aprovar el compte. L'objectiu era fer servir Amazon com a programa d'afiliats i Telegram com a canal de difusió.

Amazon és la plataforma de comerç electrònic més popular i la que genera més confiança entre els compradors, que es tradueix en una alta probabilitat que els clients acabin comprant. Amb una àmplia gamma de productes, ressenyes dels usuaris i un servei al client de qualitat, Amazon s'ha convertit en la primera opció per a molts.

Telegram és una plataforma que destaca per la seva capacitat de gestionar eficaçment els canals i grups de difusió, mantenint un alt nivell de privacitat per als usuaris. A més d'això, ofereix eines avançades que faciliten la automatització de l'enviament de missatges.

Tenia clar que l'automatització d'enviament d'ofertes la duria a través de codis de programació, ja que disponia del coneixement necessari. A partir d'aquí, tocava escollir el llenguatge de programació i començar a estructurar el codi i les funcions.

1.2 Justificació del treball

El desenvolupament d'aquest projecte té un doble propòsit. D'una banda, pretén desenvolupar de manera eficient una aplicació real fent servir Python i PHP. Aquest procés implica no només la programació i el disseny inicial, sinó també la posada en pràctica i la millora contínua de l'aplicació, garantint que s'adapti a les necessitats reals i als possibles canvis tecnològics.

D'altra banda, el projecte busca determinar la viabilitat del model proposat. Això implica identificar si l'aplicació pot funcionar de manera efectiva en un entorn real, responant adequadament a les demandes dels usuaris i als requisits del mercat. Aquesta anàlisi de viabilitat és crucial per decidir si val la pena invertir més recursos en el seu desenvolupament i posada en marxa. En resum, el projecte no només és un exercici tècnic, sinó també una investigació sobre la seva aplicació.

1.3 Contingut del treball

En aquest treball, el primer que trobarà el lector és la introducció, per contextualitzar els motius que han incitat a desenvolupar-lo. A continuació, un glossari amb els conceptes tècnics que s'han de conèixer per entendre el treball a la perfecció, seguit dels objectius i la metodologia.

Seguidament, trobarà la presentació formal, on primer es plantejarà una breu reflexió sobre l'automatització i els precedents històrics. En segon lloc, s'abordarà el camí a seguir per dur a terme el projecte de manera correcta.

Tot seguit, s'explicaran els codis de programació que s'han fet servir en aquest treball, acompanyat d'un esquema per facilitar-ne l'entesa.

Finalment, es comentarà la posada en producció, què s'ha fet per millorar-ne els resultats i quines dificultats han aparegut. Per concloure el treball, es farà èmfasis a les millores a futur i la conclusió per conèixer si el treball ha complert amb els objectius i si el model és viable.

2 MARC TEÒRIC I TÈCNIC

2.1 Màrqueting d'afiliació

L'afiliació és una tècnica de màrqueting utilitzada en la majoria de plataformes de venda de productes. Consisteix a recompensar econòmicament aquells usuaris que portin nous consumidors. També hi ha altres maneres de bescanviar els resultats, com per exemple, oferint descomptes i xecs a la mateixa plataforma. El model és molt rendible per les botigues en línia, ja que els seus afiliats treballen per aconseguir portar nous clients que acabin consumint. El cost d'aquesta feina és proporcional a les vendes, per la qual cosa no s'acostumen a perdre diners fent servir aquest tipus de màrqueting (Sánchez, 2023).

D'entre totes les plataformes amb aquest sistema, Amazon té el programa d'afiliats més gran del món, amb 900.000 associats (Forrest, 2023). De totes maneres, és dels programes amb les comissions més baixes. Aquestes depenen de la categoria del producte comprat pel client. Mentre que a electrònica les comissions voregen el 3%, les categories relacionades amb esports, aire lliure, casa, cotxe i similars orbiten del 6 al 8%. La moda és el segment amb més comissió, un 10% (Amazon Afiliados, s. f.). Convé tenir en compte les comissions per saber en què centrar els esforços.

Per entendre el funcionament tècnic del model i saber quins clients han comprat gràcies a quins afiliats, primer hem de conèixer com funcionen els enllaços, ja que és la manera com la plataforma els reconeix. Com a qualsevol *URL*, el primer que trobem és el protocol de seguretat, que sol ser HTTPS (Hypertext Transfer Protocol Secure). A continuació, localitzem el subdomini i el domini. Finalment, l'adreça de la pàgina que volem acabar visitant. És a partir d'aquest punt, que entren en joc els paràmetres UTM (Urchin Tracking Module), fragments d'enllaç que permeten rastrejar d'on ve un usuari. Aquest

ús està instaurat a moltes pàgines web, que tenen com a objectiu observar i entendre els hàbits dels usuaris (Moreno, 2023).

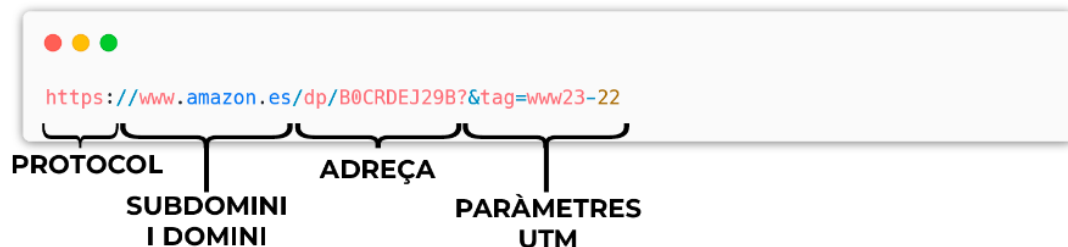


Figura 1. Paràmetres UTM. Elaboració pròpia (2023)

D'aquesta manera, la pàgina web detecta el paràmetre UTM i emmagatzema la informació en una *galleta*, que és una manera de guardar informació. En el cas d'Amazon, durant 24 hores. Això vol dir que si un usuari accedeix a Amazon amb aquest enllaç, però efectua la compra al cap de 15 hores, també s'atribueix la venda. Hem de tenir en compte que la informació de la *galleta* se substitueix si accedeix a Amazon amb l'enllaç d'un altre afiliat (Cookie Information, 2022).

Avui dia, Europa té el debat sobre la legalització de les *galletes* obertes. Tot apunta que es prohibiran i s'hauran de trobar noves tecnologies que les substitueixin. L'objectiu continua sent protegir la privacitat dels usuaris, i per això els navegadors estan treballant durament per trobar solucions que les puguin reemplaçar. Tot i que encara es desconeixen els efectes que podria tenir sobre el màrqueting d'afiliació, de moment es continua treballant de la manera tradicional (Jurídicas, 2024).

2.2 Python

Python és un llenguatge de programació de propòsit general àmpliament utilitzat amb moltes aplicacions, com ho poden ser el processament de

dades, l'automatització de tasques, el desenvolupament d'intel·ligències artificials i moltes altres. La seva simplicitat i eficiència l'han convertit en un dels codis més instaurats a escala internacional. Per això s'ha generat una gran comunitat de programadors que han desenvolupat nombrosos complements i integracions.

Python s'executa al costat servidor, i destaca per la seva naturalesa interpretada, ja que el codi es llegeix línia per línia, facilitant-ne la lectura, el desenvolupament i la depuració durant la creació d'aplicacions. Tot i ser un llenguatge simplificat i fàcil d'aprendre, la infinitat d'aplicacions i la compatibilitat d'usos l'han portat a ser una eina indispensable per tots els sectors tecnològics.

S'elabora el 1991 per només una persona, Guido van Rossum. Amb l'objectiu d'oferir un llenguatge fàcil i de codi obert, von Rossum li adjudica el nom de Python, fruit de la seva admiració als *Monty Python's Flying Circus* (About Python, s. f.).

2.3 Scrapping

Scrapping és la tècnica més popular per extreure informació d'una pàgina web. S'acostuma a fer servir si aquesta no disposa d'una API pública. El procediment és força senzill, consisteix a obtenir i guardar el codi font HTML, per després buscar-hi elements en concret. Per tant, per poder efectuar amb èxit una tasca amb *scrapping*, s'han de tenir coneixements avançats de programació.

El 2004, Jason Huggins va desenvolupar una llibreria de Python que facilitava l'extracció d'informació, anomenada *Selenium*. Això va permetre que molts programadors obtinguessin dades de pàgines web a l'instant. De totes maneres, no tot era tan fàcil, ja que les empreses van començar a destinar recursos per evitar aquesta tècnica. Fins el dia d'avui, la majoria de pàgines

tenen sistemes anti-*scrapping*, fet que dificulta força la tècnica. (*About Selenium*, s.f.).

Google, per exemple, va destinar molts esforços i recursos a desenvolupar la tecnologia *CAPTCHA*, que impedeix l'accés a pàgines web per part de robots. Ho fa mostrant una graella d'imatges on s'han de clicar les que segueixin un cert patró, fent que l'accés per part de robots sigui impossible (ReCAPTCHA, s. f.)

Amazon també treballa per aconseguir-ho, actualitzant el codi HTML font de la pàgina sovint i controlant que el comportament de l'usuari sigui humà. De totes maneres, sempre hi ha maneres d'extreure informació mentre estigui controlat i no sigui abusiu. (Is It Legal To Scrape Amazon And Never Getting Blocked | Octoparse, 2023)

2.4 Servidor Cloud VPS

Per poder posar en producció aquest projecte, es fa servir un Servidor Cloud VPS a IONOS. Un servidor VPS (Servidor Privat Virtual) consisteix en llogar les prestacions d'un ordinador virtual que no has de tenir físicament. Això ofereix la flexibilitat d'escalar els recursos segons les necessitats del projecte, sense les limitacions dels servidors físics tradicionals.

L'accés al servidor VPS es realitza a través de terminal, utilitzant comandaments tècnics i específics que permeten gestionar i configurar l'entorn de manera remota. (VPS | Servidores Virtuales A Partir de 1€/Mes | IONOS, s. f.)

2.5 API

Una API o Application Programming Interface és un conjunt de procediments que permeten integrar funcionalitats a l'aplicació. Funciona

com un transport d'informació des d'un origen fins a un destí. Existeixen diferents APIs en funció de les tasques requerides. Per una banda, hi ha les que permeten rebre informació d'una base de dades, com és el cas de la d'Amazon. Per tant, es genera una petició a l'API indicant l'accés i la informació necessària, i aquesta s'encarrega de transportar-la a la seva base de dades, comprovar que l'accés és correcte i retornar la documentació sol·licitada.

Per altra banda, trobem les que ens permeten enviar informació, com és el cas de Telegram, que pot servir, d'entre altres funcions, per enviar missatges a un grup o canal. Es proporciona la informació de l'usuari i del missatge a l'API, que Telegram rep i s'ocupa de satisfer la petició instantàniament.

2.6 PHP

PHP és un llenguatge de programació del tipus interpretat que s'executa al cantó del servidor. S'utilitza per generar pàgines web dinàmiques, introduint components de lògica a l'estructura HTML. En el context d'aquest treball, es farà servir per enregistrar certa informació a una base de dades des del dispositiu de l'usuari, ja que s'allotja a internet.

2.7 Base de dades

Una base de dades és un sistema d'emmagatzematge d'informació que es fa servir per enregistrar, actualitzar i gestionar dades de manera eficient. Funciona mitjançant taules, on les dades s'organitzen en files i columnes, com si es tractés d'un full de càlcul. Cada fila en una taula representa un registre específic d'informació, mentre que cada columna defineix cada tipus de dada. Això permet estructurar-les d'una manera coherent per a la posterior gestió.

3 OBJECTIUS DEL TREBALL

El propòsit principal d'aquest treball és tècnic: aprendre a desenvolupar una aplicació avançada capaç d'enviar ofertes d'Amazon directament a Telegram. Aquest objectiu implica dominar diverses tecnologies i processos, com ara *l'scrapping* per extreure dades de les pàgines d'Amazon, la utilització d'APIs per a la comunicació amb altres serveis, i la gestió de dades per tractar i emmagatzemar la informació recopilada. A més, és necessari automatitzar tasques mitjançant cron jobs per assegurar que les ofertes es publiquin de manera regular i puntual.

El desenvolupament d'aquesta aplicació també requereix coneixements en servidors VPS per desplegar i mantenir el sistema, així com en el llenguatge de programació Python per crear els scripts necessaris. La integració amb Telegram suposa la creació de bots que reben i distribueixen les ofertes als usuaris. Finalment, la posada en producció implica assegurar que tot el sistema funcioni de manera fluida i contínua, mentre que l'analítica de dades permetrà monitoritzar el rendiment de l'aplicació i fer ajustos per optimitzar-ne el funcionament.

D'altra banda, el propòsit secundari és comprovar i validar el model de negoci d'afiliació amb una intencionalitat experimental. Aquest aspecte del projecte busca avaluar si l'enviament automatitzat d'ofertes d'Amazon a través de Telegram pot generar ingressos mitjançant programes d'afiliació.

4 PRESENTACIÓ FORMAL DEL PROJECTE

4.1 Negoci

4.1.1 Per què decidim automatitzar? Context social

La destinació de recursos per poder automatitzar tasques ha sigut constant durant la història de la humanitat, des de les primeres invencions fins als últims avenços tecnològics. Però, per què optem per substituir tasques humanes amb màquines i tecnologies?

Les primeres invencions daten a l'edat de pedra, quan es construïen trampes per caçar animals. Els motius no eren pas econòmics, ja que es feien servir per facilitar i augmentar la quantitat d'animals caçats, a part de reduir el risc dels humans en enfrontar-s'hi.

La ciutat de Babilònia, situada a l'actual Iraq, va destacar com a pionera en aquest àmbit, gràcies al sistema de reg més avançat del moment. Un dels seus monuments més emblemàtics eren els jardins que adornaven la part alta de la ciutat. Ara bé, la gestió de l'aigua es presentava com un desafiament, que requeria portar una gran quantitat d'aigua des del riu fins a dalt de tot.

En un principi, aquesta tasca es duia a terme amb un gran nombre de treballadors que carregaven aigua en múltiples trajectes diaris. Aquest mètode implicava un desgast físic considerable, lentitud en el procés i sous pels treballadors.

Per abordar aquestes limitacions, es va implementar un sistema de reg innovador, capaç de conduir l'aigua des del riu fins als jardins, eliminant la necessitat de carregar-la manualment. Aquesta solució va suposar un gran avanç en termes d'eficiència i productivitat, ja que es va implementar l'any 2.000 a.C. (Brunete, San Segundo i Herrero, 2020).

Aquesta prorrogada tendència s'ha vist impulsada per diversos factors. Un dels principals és la cerca de l'eficiència i productivitat. Les màquines poden

dur a terme tasques més precises, ràpides i econòmiques que els humans. Això no només redueix els recursos destinats a la fabricació o realització d'un procés, sinó que també minimitzen els errors humans.

A més, l'automatització s'aplica, sobretot, en aquells processos que requereixen una càrrega de treball físic i monòton pels treballadors. Per tant, no només millora la seva salut física i benestar, sinó que també permet que els humans destinin els esforços en tasques més creatives i úniques.

Els motius econòmics també són un factor clau quan es decideix automatitzar. Tot i que la seva implementació pot necessitar una inversió considerable, a la llarga resulta una reducció significativa de les despeses en recursos humans i en una major eficiència operativa.

D'altra banda, en un món cada vegada més digitalitzat, trobem la necessitat d'implementar facilitats i automatitzacions com a resposta davant la competència. La integració de tecnologies avançades pot permetre a les empreses mantenir-se competitives i créixer en un entorn canviant.

4.1.2 Ètica de l'automatització

Hem vist que automatitzar processos és sinònim d'eficiència a la producció i reducció de costos. Ara bé, és ètic prescindir del factor humà? Quina és la línia entre automatitzar un lloc de treball o contractar a una persona?

La substitució del treball va començar fa milers d'anys, accentuant-se amb l'era digital i la sortida de molts programaris que facilitaven la feina optimitzant temps i cost. A mesura que han anat passant els anys, tots hem implementat facilitats al nostre entorn que prescindien del factor humà.

La manera com personalment entenc l'automatització, és com a mecanisme d'evolució per la societat. Actualment, si a l'antiga Babilònia encara haguessin de tenir a treballadors per a pujar aigua als jardins, estaríem molt poc

avançats i les feines serien molt precàries. Per a reforçar aquesta explicació, m'agradaria citar a Robert Kiyosaki, qui al llibre de *Cashflow Quadrant* escriu que l'automatització no és una substitució del treball, ans al contrari, és un flux que obre les portes a les noves generacions. Permet que els joves treballin en feines més avançades i creatives. Per això evolucionem, i les feines actuals no són les mateixes que les de fa cent anys. (Kiyosaki, 2011).

Considero que automatitzar aquest projecte és ètic i lícit, per dues senzilles qüestions. La primera, perquè no implica deixar a ningú sense feina, ja que és un projecte nou i aquest lloc no existeix. La segona, perquè és una posició que comporta molta feina, però poca habilitat i creativitat. Això, en certa manera, es tradueix com a facilitat per a substituir-la.

La inversió inicial que comportaria tenir un treballador per a fer aquesta feina és considerable. Per aquest motiu, aquest projecte no es pot emprendre amb la contractació d'una persona. No és pas cap feina creativa i disposem de coneixements per desenvolupar un sistema que faci la mateixa feina automàticament. De totes maneres, he volgut calcular el cost econòmic si haguéssim decidit contractar una persona.

Partim del salari mínim interprofessional del moment, a dia 10 de gener de 2024, que és de 1.134€. Comptant la Seguretat Social i altres costos fiscals, passem als 21.000€ anuals en costos per un treballador. Dit amb altres paraules, hauríem d'aconseguir 25 vendes associades cada dia per assumir aquest cost, comptant una mitjana de 20€ de preu del producte i un 8% de comissió d'afiliat.

Aquest cost no és susceptible d'assumir per part de cap empresa ni inversor, pel projecte que hem plantejat. Per tant, només tenim una opció per executar-lo: automatitzar i reduir els costos inicials.

4.2 Explicació dels codis

És hora de saber quina tecnologia faig servir per desenvolupar l'aplicació encarregada d'automatitzar l'extracció d'ofertes a Amazon i enviar-les a Telegram. Pel tipus d'objectius que he establert de moment, faré servir Python. És el llenguatge de programació més popular i estès que actua al costat del servidor, per poder fer l'extracció d'informació i el posterior enviament. També faré servir altres llenguatges de programació, que establiré més endavant.

Per a poder desenvolupar el projecte correctament, he hagut de plantejar-me els objectius a llarg termini. Això ha desembocat en un repte, que ha tingut molta repercussió durant la creació. Es tracta de l'escalabilitat, ja que l'aplicació ha de ser capaç d'integrar altres plataformes de comerç electrònic i altres canals de comunicació. Com a qualsevol projecte tecnològic, el desenvolupament ha de contemplar la capacitat del sistema per créixer i adaptar-se a futures necessitats sense haver-ne de comprometre el funcionament.

El primer esquema per estructurar la complexitat del projecte ha estat el següent:



Figura 2. Esquema de l'objectiu del projecte. Elaboració pròpia (2023)

Primer, s'obté l'oferta a Amazon per, finalment, ser enviada a Telegram. Ara bé, aviat m'he adonat que tot plegat comporta un desenvolupament força més complex. Per començar, perquè s'han de respectar els requisits de cada canal. Com he comentat anteriorment, cada segmentació d'interessos necessita ofertes diferents. No s'enviaran els mateixos productes a un canal de futbol i a un de cotxes.

Per tant, s'han d'afegir els requisits abans d'obtenir l'oferta. També s'ha d'implementar un procés de neteja de dades, anterior a l'enviament. Hi ha canals que, per exemple, segmenten per idioma. Aleshores, s'ha de traduir el títol just després d'aconseguir-lo.

De moment, l'esquema actualitzat queda així:

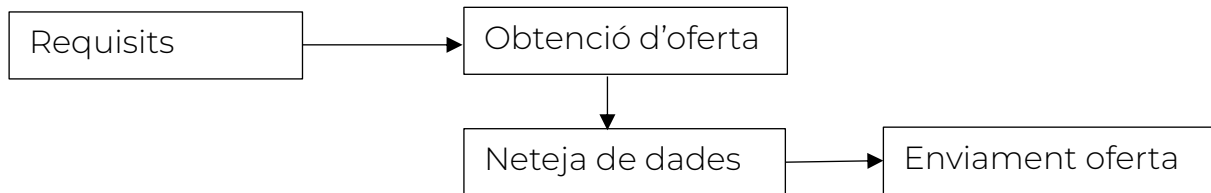


Figura 3. Esquema de l'objectiu del projecte. Elaboració pròpia (2023)

Sent conscient de la importància que tenen les dades als models digitals, no he volgut perdre l'oportunitat d'aprofitar-ho. Per això, he volgut enregistrar tots els productes que s'envien als canals de difusió, per a poder-ne fer una posterior analítica.

També he volgut saber quins productes generen més clics, quins preus i ofertes tenen, a quina hora es produeix la interacció... Tota aquesta informació s'ha d'introduir a la base de dades.

Per tant, he hagut d'incorporar el flux de dades per finalitzar el primer esquema.

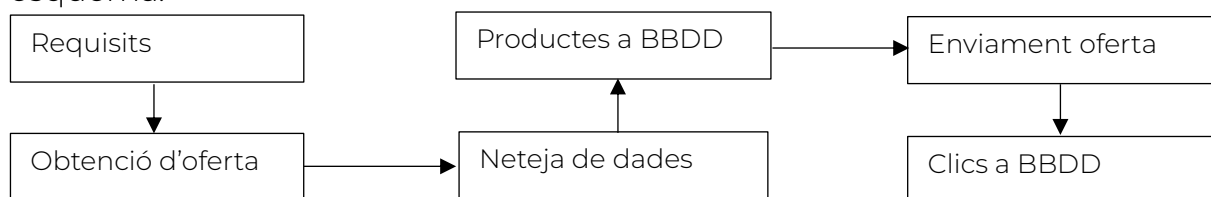


Figura 4. Esquema de l'objectiu del projecte. Elaboració pròpia (2023)

Per motius lògics i de practicitat, el sistema serà un conjunt de codis de programació que s'executaran per satisfer l'automatització. Python ofereix les prestacions necessàries, com per exemple llibreries per obtenir ofertes, inserir i seleccionar informació d'una base de dades, enviar missatges a Telegram...

Arribats a aquest punt, he hagut de considerar aquests apartats, i com els duré a terme.

4.2.1 Obtenció d'ofertes

Per obtenir l'oferta a Amazon, he hagut de considerar diferents mètodes. Tots tenen els seus avantatges i inconvenients, i la millor manera de decidir correctament ha sigut comparant-los.

Mètode 1. *Scrapping*

L'extracció d'informació mitjançant *scrapping* és un mètode habitual per aconseguir dades de determinades pàgines web, però no sempre és legal. Consisteix a obtenir el codi font amb HTML, i cercar-hi el que s'hi desitgi.

Avantatges:

- Personalització de l'extracció. D'aquesta manera, la recollecció de dades i la navegació són lliures, i es poden adequar als requisits de l'aplicació.
- Mètode gratuït. No s'ha d'efectuar cap pagament per a poder fer *scrapping*.

Inconvenients:

- Programació complexa. Programar un codi d'extracció de dades és complicat i necessita temps. Requereix coneixements avançats i de llibreries.
- Termes i condicions. Amazon, com la majoria de plataformes, no permetre rebre *scrapping* freqüent. El sistema s'ha de desenvolupar d'acord amb les normatives.

Mètode 2. API d'Amazon

Amazon disposa d'una API de productes d'accés restringit. Permet efectuar consultes fàcilment i de manera completament legal.

Avantatges:

- Programació simplificada. Amb poques línies de codi, es poden obtenir les ofertes que es necessitin.
- Termes i condicions. L'API és el mitjà més acceptat per extreure dades, ja que compleix amb totes les normatives.

Inconvenients:

- Accés restringit. D'entrada, els usuaris no hi tenen accés. Com a afiliat, hauria de generar 3 vendes perquè Amazon aprovi el compte i se'm permeti accedir a l'API.

Mètode 3. API Programari extern

Hi ha serveis que ofereixen una API similar a la d'Amazon, tot i ser externa i no oficial.

Avantatges:

- Programació simplificada. Igual que amb l'API oficial, la programació és fàcil i ràpida.
- Termes i condicions. Les API externes són legals i legítimes, tampoc incompleixen amb els termes i condicions d'Amazon.

Inconvenients:

- Mètode de pagament. L'accés a aquestes API és de pagament, normalment per subscripció. La més econòmica, a hores d'ara, és a partir de vint euros mensuals. (RapidAPI, s. f.)

Després de comparar les diferents opcions, m'he decantat per començar amb *scrapping* i, després d'haver generat 3 vendes i validar el compte d'afiliats, passar a fer servir l'API d'Amazon. D'aquesta manera, m'estalvio pagar subscripcions d'APIs externes.

A més, com s'ha comentat abans, el projecte s'anirà escalant a poc a poc, i les peticions per *scrapping* al principi seran poques i no afectaran a polítiques de termes i condicions d'Amazon.

M'he decantat per Python com a llenguatge de programació per obtenir i enviar les ofertes. Disposa de força llibreries i una gran comunitat de programadors que m'ha permès resoldre dubtes durant el procés.

Configurar l'*scrapping* ha sigut una feina força complicada, per la seguretat d'Amazon davant d'aquestes pràctiques. La llibreria de Python que m'ha permès fer-ho, ha sigut Selenium, que simula accions humanes en un navegador. Per tant, quan s'executa, s'obre el navegador seleccionat i comença a complir les ordres programades al codi.

Tenint en compte el repte de l'escalabilitat, he creat els requisits. Són variables que s'ajusten a les necessitats de cada canal de difusió. Per exemple, si en un canal de difusió s'han d'enviar ofertes per menys de 10 euros, aquest serà un dels requisits. Tenen la particularitat de declarar-se abans de l'*scrapping*.

Per tant, gràcies a un bucle d'extracció de dades, podem fer servir el mateix codi per obtenir ofertes amb els requisits necessaris. D'aquesta manera, crear nous canals de difusió serà molt més fàcil.

4.2.2 Processament de dades

Quan ja s'ha fet l'*scrapping* i s'han obtingut les ofertes, toca netejar-les i enregistrar-les a una base de dades. En aquesta fase, passem de tenir

requisits a tenir processos. En el primer canal, per exemple, s'han d'enviar ofertes en català. El procés consistirà a traduir el títol.

Amb l'*scrapping* he obtingut aquestes dades:

- Títol de l'article.
- ASIN (codi intern dels productes d'Amazon).
- Preu sense oferta.
- Preu amb oferta.

Les ofertes passen a validar-se, comprovant, per exemple, que el preu sense oferta sigui major al preu amb oferta, o que el títol no contingui caràcters especials. Si és així, s'enregistra a la base de dades, que explicaré més endavant.

4.2.3 Enviament d'ofertes

És, en aquest moment, quan s'extreuen les ofertes de la base de dades per enviar-les als canals de difusió en qüestió. En una primera instància, els canals s'ubiquen a Telegram, per la qual cosa he programat un "bot" dins la plataforma. Aquest, m'ofereix una API, que em permetrà connectar-lo des del codi perquè sigui ell qui envii les ofertes.

Això també implica instal·lar i manipular la llibreria de Python *Telegram*, i programar-ne els missatges.

4.2.4 Obtenció de dades

En l'actualitat, gairebé qualsevol negoci digital nota un increment de resultats si sap fer servir de manera adequada les dades que obté de la seva activitat. (Ntg, 2023). En el cas de l'enviament automàtic d'ofertes, s'ha de treballar en l'obtenció, la interpretació i l'aplicació. Serà útil per saber què agrada més al públic, quins preus són els més vistosos, què es compra més...

Per a obtenir aquestes dades, he creat un tràmit entre el clic que es produeix en un canal, i l'aterratge a la pàgina de destí. Aquest tràmit és un reencaminament automàtic que, per tant, l'usuari no ho detecta. S'enregistra informació sobre el clic, com la data, l'hora, el canal i el producte.

L'objectiu d'aquest registre serà, quan se'n tinguin molts, fer estadística i conèixer què és el que cliquen i compren més els usuaris. Com desenvoluparé a l'apartat de millores a futur, l'aplicació de la intel·ligència artificial en aquest procés permetria automatitzar encara més el funcionament de l'aplicació de manera òptima i eficient.

4.2.5 L'execució dels codis

Vistes les funcions que necessita l'aplicació, és hora d'entendre quants codis i quins llenguatges s'han de fer servir.

- Codi 1 [Python]: Obtenció d'ofertes.
- Codi 2 [Python]: Enviament d'ofertes.
- Codi 3 [PHP]: Registre de clic.

Els arxius Python es poden executar a qualsevol ordinador. Una idea seria tenir un portàtil sempre obert, tot i que no m'ha acabat de convèncer. Gràcies a una de les aportacions del tutor, però, he descobert un servei que ofereix l'empresa de serveis al núvol IONOS.

Es tracta dels servidors VPS, que actuen com un ordinador remot. Per a controlar-los, he d'accedir a través de la terminal del meu ordinador, i un cop a dins tot és molt similar a un sistema operatiu comú, amb la diferència que no hi ha interfície gràfica.

D'aquesta manera, he pogut passar els arxius al servidor a l'"ordinador llogat". La posada en marxa no ha sigut fàcil, ja que he hagut d'instalar i configurar les aplicacions, com a qualsevol ordinador. Quan tot funcionava

correctament, he hagut de programar que els codis s'executin de manera automàtica cada cert temps.

He fet servir el sistema *crontab* per programar l'execució automàtica. És un sistema innat dels sistemes operatius, on has d'indicar les instruccions de l'execució:

- Freqüència: combinació de números i símbols per establir cada quant s'executa.
- Aplicació: quin programa es farà servir per a l'execució.
- Ruta a l'arxiu: on s'ubica l'arxiu que es vol executar.
- Ruta de destí per problemes: en cas d'haver-hi una incidència a l'execució, el missatge d'error es desaria a l'arxiu especificat.

Per tant, ha quedat així:



Figura 5. Estructura *crontab*. Elaboració pròpia (2024)

La cadena "30 * * * *" significa que s'executarà al minut 30 de cada hora. A continuació, hi ha la ruta a Python, que actua com a entorn del codi. En aquest cas és "/usr/bin/python3". Aleshores trobem la ruta a l'arxiu que ha d'executar, que per aquest primer canal de difusió, és *preuscat-insert.py*, que es troba a "/home/app/preuscat/preuscat-insert.py". Finalment, trobem l'arxiu on desaria les incidències, si n'hi haguessin.

Aquesta és la metodologia per executar el primer codi de Python. Per executar el segon, però, no fa falta programar-ho d'aquesta manera. Quan acaba el primer, deriva a l'execució del segon, facilitant la feina i mantenint un ordre convenient.

Fins aquí, gairebé tot funciona correctament. Només hi ha un error, i el trobem al servidor. De tant en tant, atura les execucions i deixa de funcionar indefinidament. La manera de solucionar aquest problema l'he trobat reiniciant el servidor cada dia a les 00.00 hores:

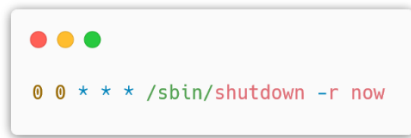


Figura 6. Estructura crontab. Elaboració pròpia (2024)

4.2.6 Bases de dades

He parlat força d'emmagatzemar informació a les bases de dades, però encara no he explicat de quina manera i a on. El primer contacte que vaig tenir amb les bases de dades va ser amb Alwaysdata, una empresa de serveis digitals. Ens ho van ensenyar en una assignatura de segon. Des d'aleshores, em resulta molt fàcil manipular aquest sistema. A més, és gratuït i segur.

L'emmagatzematge de dades es du a terme com si es tractés d'un Excel. Hi ha les columnes, que són les diferents dades que es volen introduir; i les files, tants com registres hi hagi. En una base de dades, s'ubiquen diferents taules, per poder emmagatzemar informació amb diferents estructures en un mateix lloc. En aquest cas, en tenim dues:

Taula 1: Informació de l'oferta.

Nom columna	Descripció columna
ID	Número únic a tota la columna, per poder identificar la fila.
ASIN	Codi alfanumèric per identificar cada producte d'Amazon. Tot i que tècnica permet que hi hagi el mateix ASIN a dues files, el codi que he programat, no.

TÍTOL	Títol del producte tal com s'enviarà a Telegram. En el primer cas, en català.
PREU ACTUAL	Preu que s'està aplicant en aquest moment. Inferior al preu anterior.
PREU ANTERIOR	Preu del producte quan no estava en oferta.
DESCOMPTE	Descompte en percentatge.
DATA I HORA	Data i hora de l'enviament a Telegram.
ENVIAMENT	Columna de "0" i "1" segons ja s'ha enviat a Telegram o no.

Taula 7. Columnes de la taula d'ofertes de la base de dades. Elaboració pròpia (2024)

Taula 2: Informació del clic.

ID	Número únic a tota la columna, per poder identificar la fila.
ASIN	Codi alfanumèric per identificar cada producte d'Amazon. Tot i que tècnica permet que hi hagi el mateix ASIN a dues files, el codi que he programat, no.
DATA I HORA	Data i hora del clic.
PLATAFORMA	Des de quina plataforma ha vingut, com per exemple, Telegram.
CANAL	Des de quin canal ha vingut, com per exemple, PREUS.

Taula 2. Columnes de la taula de clics de la base de dades. Elaboració pròpia (2024)

A la segona taula, no és necessari incorporar informació de l'ASIN que ja tinguem a la primera, com el preu o el descompte. Es poden relacionar les taules quan es fa el processament de dades, i combinar informació de les dues. Si emmagatzemés la informació de l'ASIN a la segona taula, l'únic que aconseguiria seria ocupar més espai a la base de dades.

4.3 Desplegament tècnic

4.3.1 Primer codi: recerca d'ofertes

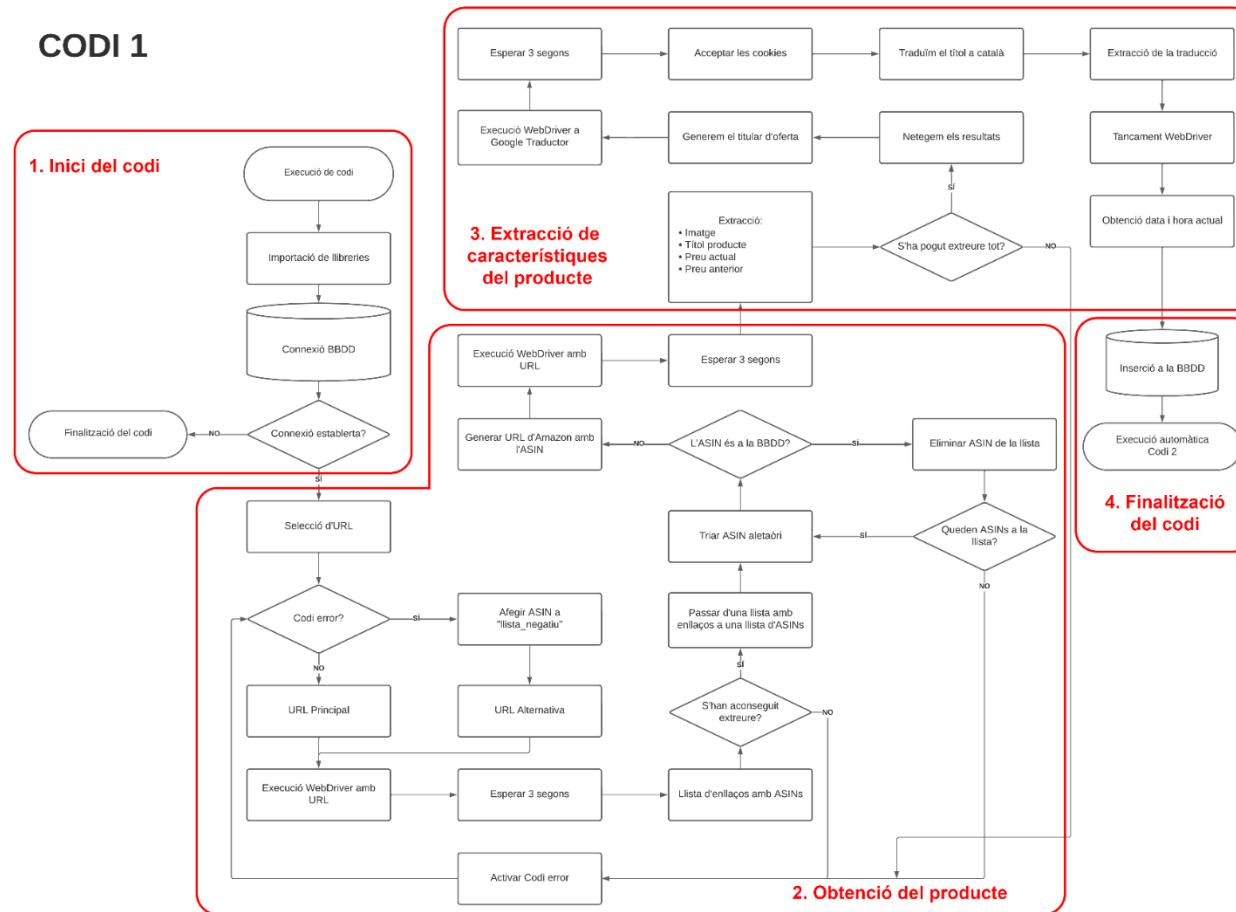


Figura 8. Esquema del primer codi de Python. Elaboració pròpia (2024)

a) Inici del codi

El llenguatge de programació Python permet realitzar moltes funcions, però en necessitarem més per poder dur a terme l'automatització. Per això, el primer que hi ha en un codi són les importacions de llibreries que ens ofereixen més funcionalitats.

A continuació, he fet la connexió amb la base de dades, que més endavant ens permetrà saber els productes enviats, i enregistrar el que s'enviarà. Igual que amb les llibreries, s'ha de connectar abans de fer-se servir. Si la connexió no s'ha pogut establir, el codi s'acaba. De totes maneres, és poc probable que això passi.

b) Obtenció del producte

Si, per altra banda, la connexió s'ha establert, el codi continua. Tot seguit, hem de decretar l'URL des d'on s'extrauran les ofertes. Aprofito per esmentar que he implementat un sistema de prevenció d'errors, que s'executa quan el codi entra en problemes. Per exemple, si no troba ofertes a l'URL proporcionat, canvia automàticament l'URL i ho torna a intentar.

Així, tenint un URL o un altre, el codi procedeix a obrir el navegador sense mostrar-se per pantalla. He de recordar que el programa s'executa a un servidor web, no a cap ordinador.

Amb el navegador Google Chrome obert, però no visible, el codi espera 3 segons perquè es pugui carregar la pàgina amb l'URL. Com he comentat en un principi, fent *scrapping* obtenim el codi HTML, un conjunt d'etiquetes i atributs per poder estructurar la part visual d'una pàgina web. El conjunt d'ofertes al web és aquest:

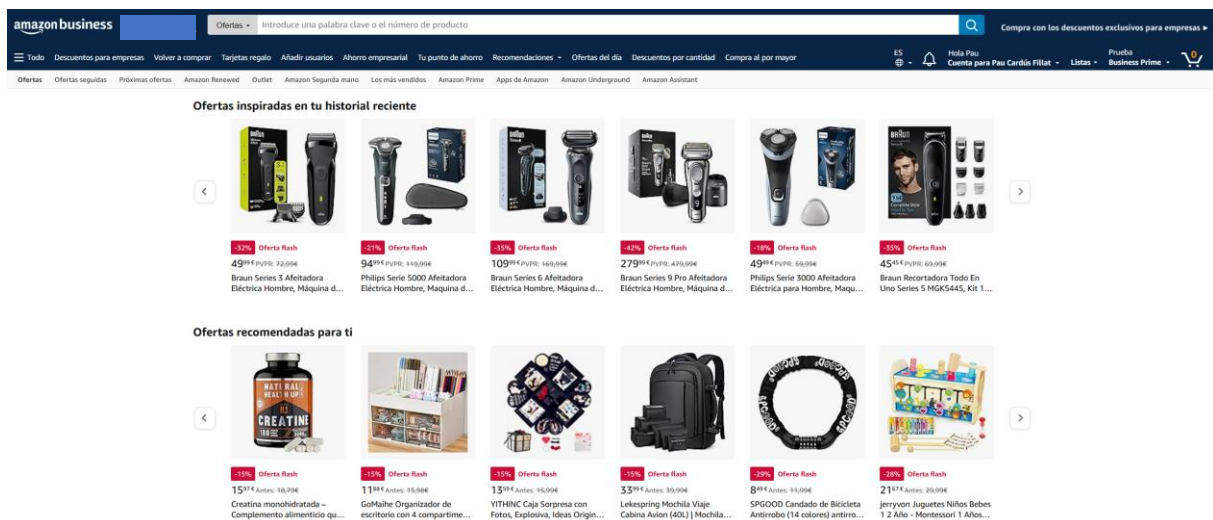


Figura 9. Captura de pantalla de la pàgina d'ofertes d'Amazon.es. Elaboració pròpia (2024)

És, en aquest punt, on he hagut de mirar com s'estructura amazon.es. La pàgina d'ofertes és una llista d'enllaços com aquest:



Figura 10. Paràmetres d'un enllaç de producte d'Amazon. Elaboració pròpia (2024)

Els enllaços a HTML es representen amb l'etiqueta `</a>`, i a dins s'ubica el paràmetre `href`, que conté l'adreça de destinació. En aquest exemple, si algú clica en aquest enllaç, anirà a parar a `amazon.es + /Braun-Afeitadora-Recortadora[...]/dp/ B0B3JC16GW`.

Tornant al codi, passats els 3 segons, fa una llista d'enllaços que continguin `"/dp/"`. Amb terminologia tècnica d'Amazon, qualsevol enllaç que contingui `"/dp/"` va acompanyat d'un ASIN, i això vol dir que és un producte.

L'ASIN és el que he fet servir com a referència en tot moment per a referir-me als productes. Per tant, quan el codi busca productes, està programat per obtenir una llista d'ASINs.

Una vegada ha obtingut tots els enllaços de la pàgina que contenen “/dp/”, fa una neteja, on només es queda amb els ASINs. En cas que no s'hagi pogut extreure cap enllaç, el codi selecciona un URL alternatiu i ho torna a intentar.

Tenint una llista d'ASINs, el codi en tria un aleatòriament i comprova que no està registrat a la base de dades. Si no és així, redirigeix el navegador a la pàgina “www.amazon.es/dp/+ASIN”.

L'execució en aquesta part és la mateixa que a l'anterior: identificar elements a través d'etiquetes HTML i extreure-les.

Ara ens trobem a la pàgina de detalls del producte:

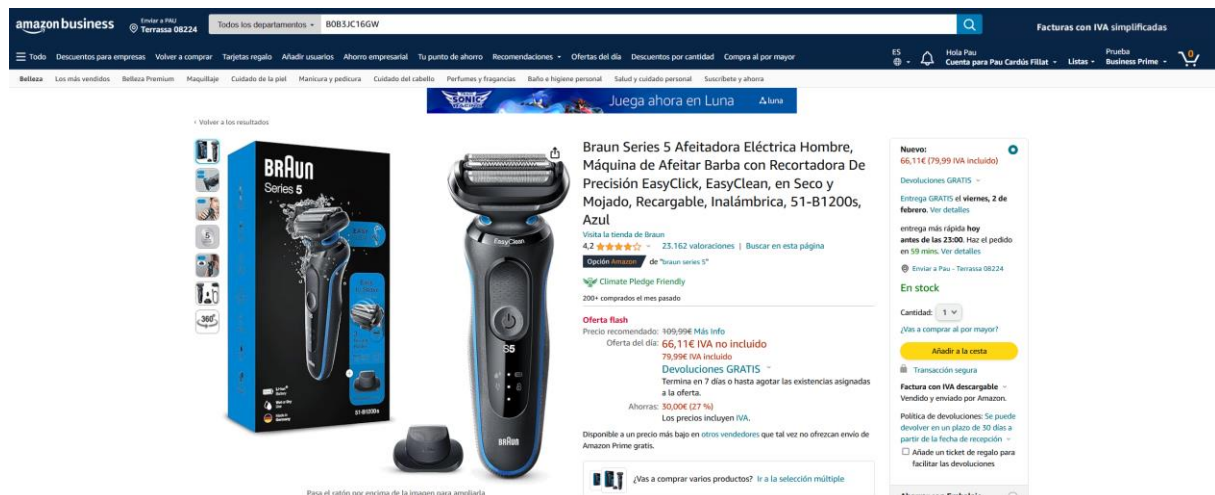


Figura 11. Captura de pantalla de la pàgina de detalls d'un producte d'Amazon. Elaboració pròpia (2024)

c) Extracció de característiques del producte

D'aquí, hem d'extreure la informació que enviarem a Telegram, és a dir, la imatge, el títol, el preu anterior i el preu amb oferta.

És probable que en alguns casos aquesta informació no es trobi fàcilment. Si el producte té variacions de colors, dimensions o preus sense IVA, es pot arribar a bloquejar el codi. Aquí també hi intervé el sistema de prevenció d'errors. Si no s'han pogut extreure dos preus, o si el preu en descompte és més alt que el preu habitual, salta el sistema. En aquest cas, elimina aquest

ASIN de la llista anterior, i ho torna a provar amb un altre. Si no en queden, saltarà el sistema novament per agafar un URL alternatiu.

A la majoria dels casos, l'extracció d'informació haurà sigut un èxit i tindrem a disposició tota la informació necessària. A continuació, tot i que ja podríem enviar aquesta informació als canals de difusió, haurem de passar per dos processos més.

El primer, per poder traduir el títol a català. Com s'ha comentat anteriorment, alguns canals tenen requisits que no consisteixen a segmentar els productes, sinó a transformar-los quan ja els tenim. Per tant, si l'oferta s'ha d'enviar al grup de "preus", el codi s'emporta el títol a Google Traductor, on segueix la següent metodologia.

Primer, carrega l'enllaç: "*translate.google.es/?hl=ca&sl=es&tl=ca&op=translate*". Això permet obrir la pàgina del Traductor amb els idiomes configurats (de castellà a català). Abans de començar, ha d'acceptar les galetes, requisit per accedir-hi. Aleshores, espera 3 segons perquè la pàgina s'acabi de carregar, enganxa el text al requadre de l'esquerra (idioma original) i copia el resultat del quadre de la dreta (idioma traduït).

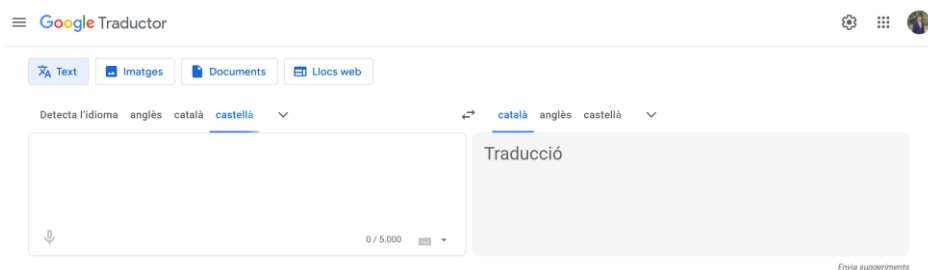


Figura 12. Captura de pantalla de Google Traductor. Elaboració pròpia (2024)

d) Finalització del codi

Finalment, amb tota la informació necessària extreta i transformada, es procedeix a l'enregistrament d'aquesta a la base de dades, no sense abans haver tancat el navegador. Per tant, obtenim la data i hora actuals i ho registrem tot. Aleshores, es crida a l'execució del codi 2.

4.3.2 Segon codi: enviament d'ofertes

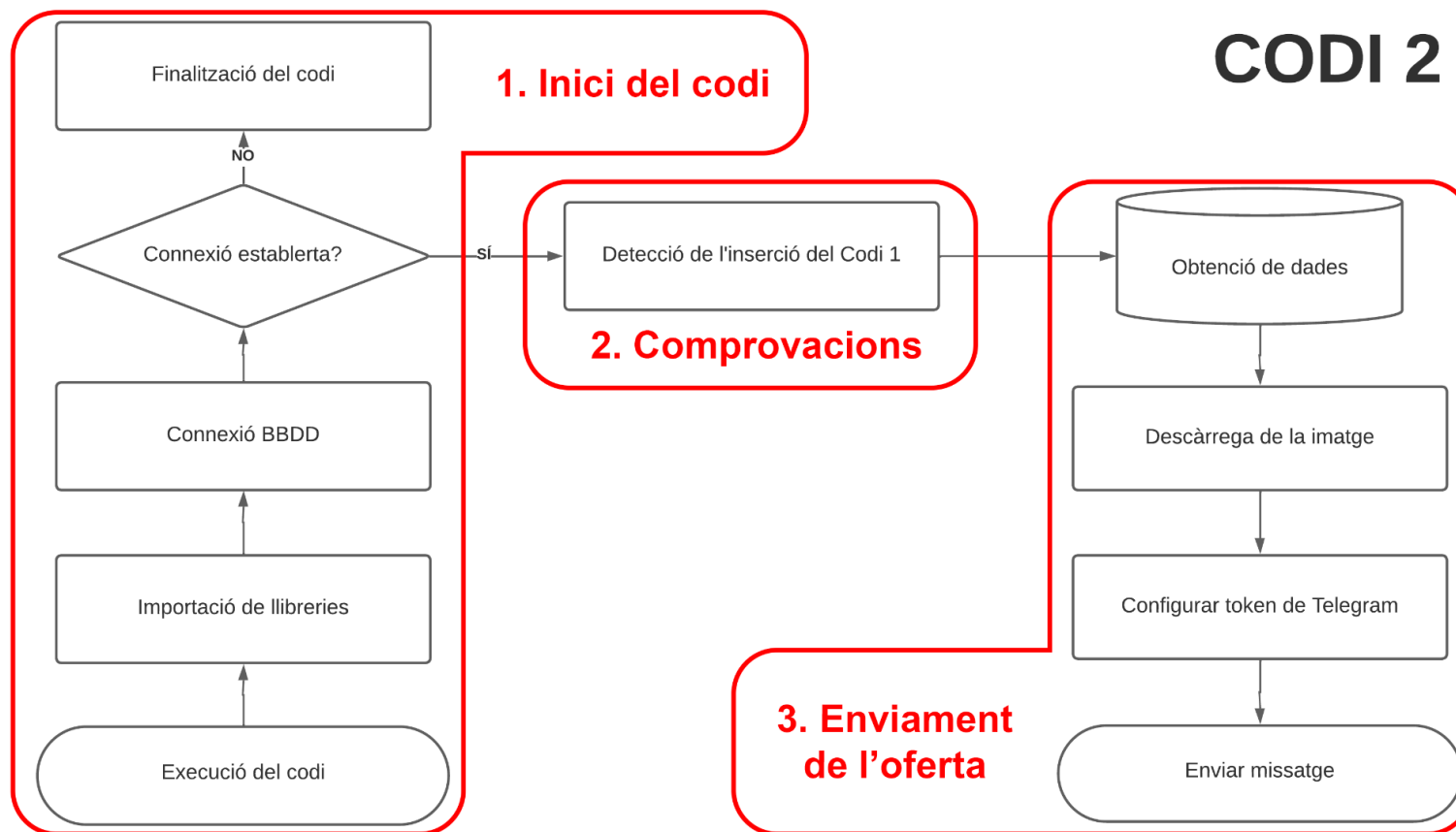


Figura 13. Esquema del segon codi de Python. Elaboració pròpia (2024)

a) Inici del codi

L'execució del segon codi es produeix just després d'acabar el primer, de manera automatitzada i immediata. Igual que amb el primer codi, s'importen les llibreries necessàries i s'intenta connectar amb la base dades. Si no ho aconsegueix, finalitza el codi.

Si la connexió s'estableix exitosament, es procedeix al punt 2.

b) Comprovacions

En aquest punt, l'aplicació recórrer la base de dades amb l'objectiu de trobar la informació que s'ha inserit al final del codi 1. El procediment per aconseguir-ho és el següent.

Abans hem vist que a la primera taula de la base de dades, on s'enregistrà informació del producte, hi ha la columna ENVIAMENT. Quan el codi 1 ha inserit la fila, hi ha enregistrat un "0". Per tant, el codi 2 busca la fila que conté aquest número. Un cop el troba, guarda tota la informació de la fila i canvia el "0" per un "1", indicant que el producte ja s'ha enviat i que no es torni a agafar més.

c) Enviament de l'oferta

Aquesta és la part final del segon codi. Quan tenim l'ASIN que hem afegit abans, el codi obté la informació relacionada (nom, preus, descompte...) i procedeix a descarregar temporalment la imatge principal. Amb tota aquesta informació, el següent pas consisteix a configurar l'enviament de missatges a Telegram, amb una de les llibreries instal·lades al pas 1.

Com és habitual amb aquestes metodologies, Telegram ofereix públicament la creació de "bots", que es poden controlar amb codis de programació. Per fer-ho possible, es proporciona un "token", que és una contrasenya de caràcters alfanumèrics. Finalment, es proporciona la contrasenya i l'oferta a la llibreria de Telegram, que l'acaba enviant al canal indicat.

4.3.3 Tercer codi: recopilació de dades

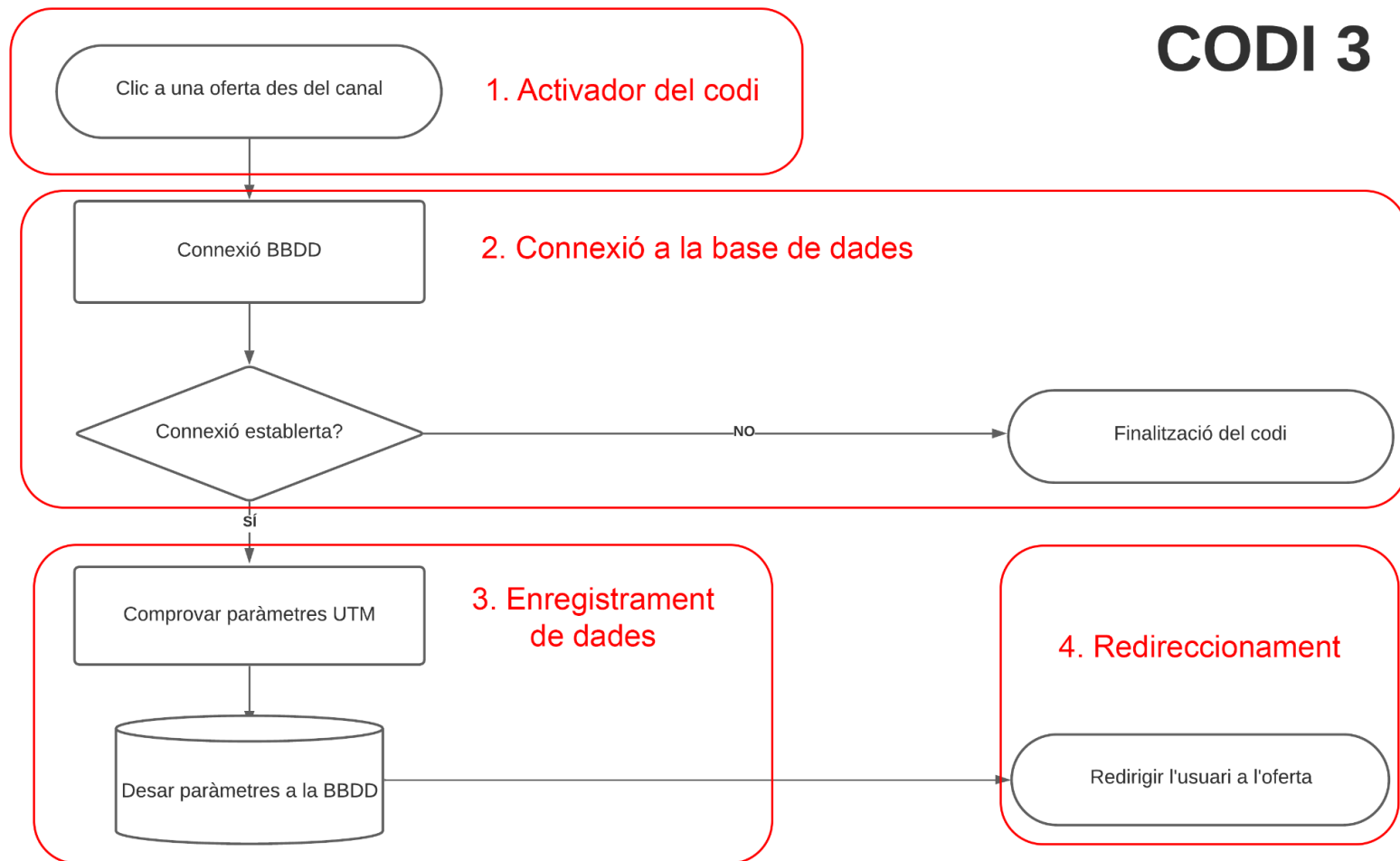


Figura 14. Esquema del tercer codi de Python. Elaboració pròpia (2024)

L'objectiu d'aquest codi és enregistrar informació del clic per analitzar els comportaments de cada un dels públics depenent del canal. Informació com la data i hora del clic, des de quin canal s'ha fet i, sobretot, quin producte s'ha clicat. Amb tot això, es podrà millorar la comprensió de cada canal i entendre'n les preferències dels usuaris.

Per poder arribar a aquest codi, hem de tornar al segon, precisament en el punt d'enviar l'oferta. Un dels paràmetres que es proporciona a Telegram quan es vol enviar el missatge és l'enllaç de l'URL. És a dir, a on aterrarà l'usuari quan faci clic a una oferta.

Tot i que el més obvi és que el porti directament a Amazon.es amb l'enllaç d'afiliació, això no ens permetria saber tota la informació comentada anteriorment. Per això he creat aquest codi, que actua com a intermediari entre el canal de Telegram i Amazon. Aquest procés intermedi és instantani, un simple reencaminament. Per tant, l'usuari no ho nota, ja que no implica més temps de càrrega.

És molt important destacar que aquest codi s'ha creat amb PHP i no Python. La raó és la següent: el programa s'ha de poder executar al dispositiu dels usuaris, en un lloc web. Això ho permet fer PHP, que té una lògica de programació molt similar a Python.

a) Activador del codi

L'execució del codi es produeix quan algú fa clic a un producte. L'URL associat reencamina a aquest codi, que s'allotja a una pàgina web. Per tant, l'usuari no s'envia directament a Amazon, sinó que ho fa a preus.cat. Aquest pas pot provocar fricció i rebuig entre els usuaris que no es fiïn d'un enllaç que no sigui amazon.es. Encara no se sap com reaccionaran els usuaris a aquest procés, però sempre es pot obviar i reencaminar directament cap a Amazon.

b) Connexió a la base de dades

A continuació, connectem el codi PHP amb la mateixa base de dades que estem fent servir durant tot el projecte, que no canvia pel fet de fer servir un altre llenguatge.

Ara bé, el que sí que canvia és la taula, ja que la informació que procedirem a enregistrar no segueix la mateixa estructura que les ofertes. No es desaran dades dels productes, sinó característiques del clic.

c) Enregistrament de dades

Quan la connexió s'estableix correctament, es procedeix a desar la data i hora del clic, la plataforma (Telegram en aquest cas), el canal (Preuscat), el preu del producte i el descompte.

Tot això serà suficient per analitzar els resultats i millorar els canals segons les necessitats dels públics.

d) Reencaminament

Un dels paràmetres que rep el codi 3 és l'ASIN del producte, que no només es fa servir per enregistrar-lo a la base de dades, ja que també servirà per redirigir l'usuari al producte que ha clicat.

En aquest punt es genera l'enllaç d'afiliació, que es veu així:



Figura 15. Enllaç d'afiliació. Elaboració pròpia (2024)

La particularitat d'aquest enllaç és el paràmetre "tag", que permet a Amazon saber el meu codi personal, en aquest cas "www23-22".

4.4 Explotació de dades i analítica

El desenvolupament d'aquesta aplicació s'ha realitzat, des d'un principi, prioritant la màxima recollecció de dades. L'aplicació podria ser tan senzilla com practicar *scrapping* a Amazon i enviar les ofertes sense enregistrar-les enlloc. També es podria reencaminar a l'usuari a amazon.es sense passar pel tercer codi, que guarda la informació relacionada amb el clic.

Sense aquests registres, el d'informació de producte i clic, el funcionament seria exactament el mateix, amb la diferència que desconixeríem quines ofertes s'envien i quines es cliquen. Aleshores, no podríem saber l'interès de l'audiència del canal, i ens impediria millorar les ofertes enviades.

Sabent que la diferència entre recollir o no les dades és força gran, he decidit implementar-ho des d'un principi. L'analítica posterior es pot du a terme de moltes maneres. Sense anar més lluny, a cop d'ull si els clics són pocs. Si el volum comença a créixer, es pot optar per introduir-les en un programa de visualització de dades, com Tableau o Looker Studio Overview. Una altra opció, encara més professional, seria desenvolupar un quart codi de Python que processi la informació automàticament, i n'extregui conclusions gràcies a la implementació de la intel·ligència artificial.

5 POSADA EN PRODUCCIÓ

5.1 Primers mesos en producció

En aquest moment, ja tenia el compte d'Amazon Afiliats aprovat i els codis per enviar les ofertes. Abans de continuar, però, m'havia d'assegurar que funcionaven correctament. Per això, vaig preparar la següent estratègia per seguir amb el projecte.

L'objectiu del projecte és, per una banda, aprendre a crear una aplicació des de zero que sigui capaç d'automatitzar l'extracció de dades d'una pàgina web per processar-les i enviar-les.

Per altra banda, l'objectiu no deixa de ser aconseguir generar ingressos entenent i gestionant l'audiència. Per això, necessito a gent interessada a rebre i aprofitar ofertes. He estructurat la següent estratègia d'audiència:

Per aconseguir obtenir vendes associades, es necessiten usuaris que accedeixin als enllaços productes, i per això es requereix tenir usuaris subscrits a Telegram. L'accés al canal de PREUS es fa a través d'Instagram, on es compartirà l'enllaç juntament amb ofertes.

En resum, tot aquest sistema és un embut, i com més s'aconsegueixi ampliar per la part superior, més ho farà per la part inferior.

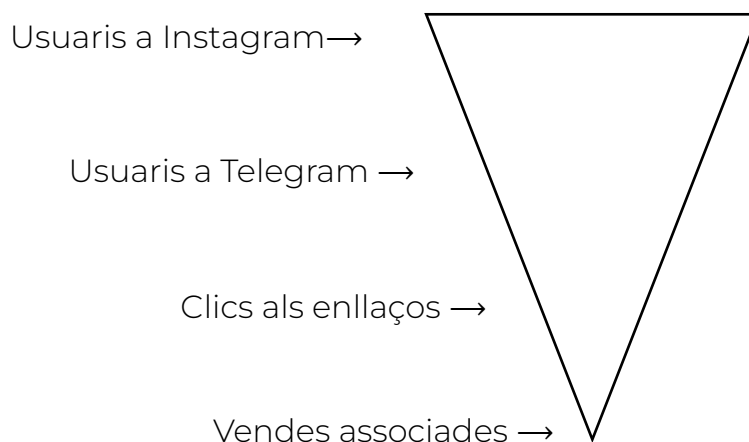


Figura 16. Embut de conversió del model de negoci. Elaboració pròpia (2024)

Per tant, com més usuaris tingui a les xarxes socials, més vendes associades obtindré. Ara bé, començar un compte des de zero no és fàcil i comporta setmanes, o fins i tot mesos de feina. Per això, paral·lelament faré les proves oportunes al canal provisional. Quan això funcioni bé, compto tenir usuaris suficients perquè comencin a entrar a Telegram.

La primera xarxa social que he decidit tenir és Instagram, pel tipus de contingut i usuaris que té. El procediment serà publicar sovint les millors ofertes de cada dia i seguir a gent que hi tingui interès. Com he comentat anteriorment, el primer segment serà el català. Ofertes de tota mena, almenys en un principi, traduïdes a català.

En aquest moment, han passat dues setmanes des del llançament. He optimitzat i millorat les ofertes de Telegram, reduint la possibilitat que apareguin errors. De totes maneres, encara n'hi ha dos que afecten el funcionament de tant en tant:

- Primer error: quan un producte d'Amazon està en oferta i té variants, el codi pot arribar a agafar preus que no són els correctes. És un error fruit de les complicacions de l'*scrapping*.
- Segon error: tècnicament, molt similar al primer. Els productes de determinades categories tenen títols amb etiquetes HTML diferents dels altres. Això vol dir que el codi no és capaç d'obtenir el títol correctament.

Pel que fa a Instagram, he aconseguit 200 seguidors penjant publicacions i històries. L'estratègia per aconseguir-los ha consistit a seguir a molta gent i fer que retornin el "seguir".

Ara mateix, ens ubiquem a la part alta de l'embut, els "usuaris a les xarxes socials". Per continuar amb els "usuaris a Telegram", insereixo l'enllaç al canal a les històries, quan s'anuncia una oferta. Per tant, qui vulgui aprofitar l'oferta, haurà d'entrar al grup.

Continuant amb els “clics als enllaços”, ha sigut un dels punts dèbils. Com a màxim, n’hi ha un o dos cada dia. Això vol dir que l’embut es tanca massa aviat, i s’ha d’aconseguir obrir per la part superior. Com pot ser obvi, no he obtingut cap venda associada.

5.2 Procés d’optimització

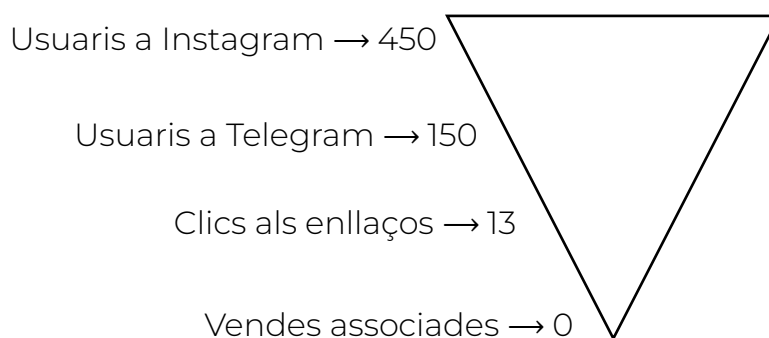
Un mes després del llançament, continuava força igual, tot i que els seguidors d’Instagram ja eren 450 i els usuaris de Telegram, 150.

De moment, entre 5 i 8 persones veuen cada oferta, que s’envia cada hora. Només he aconseguit una venda indirecta. És a dir, un usuari ha arribat a Amazon amb el meu enllaç afiliat, i ha acabat comprant un altre producte. Això també s’atribueix econòmicament amb una petita comissió.

Per tant, he d’aconseguir incrementar les vistes per cada oferta enviada. Per això he decidit implementar experiments a tot el model per entendre l’audiència i maximitzar els resultats.

El primer experiment ha consistit a incrementar el temps entre oferta i oferta. D’aquesta manera, n’hi haurà menys i tindran més visualitzacions. S’ha incrementat d’1 a 4 hores, amb la intenció d’obrir l’embut.

Al cap d’una setmana, les visualitzacions han incrementat de 5-8 a 12-15. És a dir, gairebé el triple. Ara, les probabilitats que s’obtinguin vendes han incrementat. L’embut queda així:



5.3 Dificultats en la posada en producció

Amazon, sent una de les empreses més grans del món, té els recursos per actuar davant *l'scrapping* i aconseguir-lo evitar. A principis del 2024, l'estructura HTML de la pàgina s'ha actualitzat, canviant les etiquetes i classes que configuraven la pàgina fins ara.

Les conseqüències per aquesta aplicació han sigut greus, ja que ha deixat de funcionar immediatament. El codi no detecta la mateixa informació que abans, i això fa que no la pugui extreure. Per tant, s'han deixat d'enviar ofertes a Telegram.

El nivell de l'actualització de l'HTML ha sigut alt, ja que no només s'han canviat les classes, sinó que també ho ha fet l'estructura. D'aquesta manera, és molt complicat trobar un patró per tots els preus, per exemple. Res identifica el títol com a títol, o el preu com a preu, per la qual cosa dificulta l'extracció.

En aquest punt, i veient les anteriors dificultats a l'hora d'emprendre aquest projecte, hi ha dos possibles camins a seguir. Per una banda, reescriure el codi adaptat a la nova estructura HTML. Per l'altra, concloure el projecte aquí havent provat que no és tan fàcil com fan veure.

Davant la dedicació de temps que suposa reescriure el codi, i veient que no hi ha més alternatives per continuar, he decidit concloure amb aquest experiment. Definitivament, Amazon no facilita la incorporació de nous afiliats, i acaba trobant la manera de denegar l'accés si no es compleixen tots els estrictes requisits i polítiques.

He arribat, unes quantes vegades, a aconseguir les tres vendes associades necessàries per poder sotmetre'm a l'aprovació com a afiliat, i sempre se m'ha denegat. En algunes ocasions, els motius han sigut clars, però en altres simplement se'm denegava sense raó.

6 MILLORES A FUTUR

Tot i haver decidit donar per finalitzat aquest experiment, el seu desenvolupament m'ha permès entendre'n el model i saber quines millores implementaria en un futur. Les tindrè en compte per altres projectes que pugui emprendre més endavant de magnituds similars.

Per començar, tornaria a plantejar l'organització de desenvolupament. M'he centrat, sobretot, en programar un codi *d'scrapping* per extreure la informació dels productes, però aquesta tècnica presenta certes dificultats que no el permeten ser rendible. Per contra, centraria els esforços a aconseguir l'API d'Amazon i que l'aplicació creada pogués funcionar correctament. Això, però, ho he intentat de diferents maneres, i hauria de comprovar com aconseguir l'accés més fàcilment.

A continuació, implementaria l'ús de la intel·ligència artificial a l'hora de processar les dades generades pels usuaris i triar les ofertes per enviar. És a dir, tenint una base de dades amb la informació dels clics, la IA ho faria servir per determinar quines ofertes són més interessants per enviar, tenint en compte característiques com la categoria, el preu o el descompte.

També es podria aplicar a l'hora de processar el text que s'envia juntament amb el producte. El sistema actual permetria adjuntar el títol del producte tal com és, però a Amazon aquest sol ser massa tècnic i poc clar. La IA transformaria un títol llarg en un text resumint les especificacions del producte més importants, com la marca, el tamany o les dimensions.

7 CONCLUSIONS

En primer lloc, m'agradaria destacar que l'objectiu d'aquest treball ha sigut emprendre el projecte a escala real per conèixer-ne la viabilitat. Amb satisfacció, puc afirmar que he complert aquest propòsit principal, ja que he desenvolupat, des de zero, una aplicació capaç d'automatitzar l'enviament d'ofertes des d'Amazon a Telegram, fent servir la tècnica *d'scrapping*.

Ara bé, la conclusió d'aquest experiment és que el model de negoci del màrqueting d'afiliació no és tan fàcil com ho volen fer veure alguns. Requereix, per una banda, molta dedicació, ja que s'ha de trobar la manera de mostrar els productes recomanats i que es comprin. O bé s'automatitza i es destina el temps a desenvolupar el sistema, o bé s'envien manualment. Per altra banda, es necessita cert coneixement sobre audiències i com treballar-les. Un dels punts que m'ha sorprès més, per exemple, és que centenars de persones veiessin les meves recomanacions, i ningú en comprés cap.

També és oportú mencionar que la corba d'aprenentatge d'aquest projecte ha suposat un desafiament considerable perquè que he utilitzat sistemes, llibreries i programes amb els quals no tenia experiència prèvia. Això m'ha comportat una inversió considerable de temps en recerca i aprenentatge de programació per a tasques específiques.

Convé destacar que he adquirit coneixements importants en l'àmbit de l'*scrapping* per poder-lo aplicar a una plataforma de comerç electrònic tan rellevant com és Amazon. Aquest procés ha permès extreure informació de manera eficient i ràpida, entenent les estructures del codi font de la pàgina web. També he configurat un servidor VPS per a programar els executables perquè funcionin de manera autònoma, aspecte crucial per aquest projecte.

Per altra banda, he desenvolupat un codi PHP per enregistrar els clics, que valoro, sobretot, per la planificació en recollectar més dades. Això ha permès monitorar les interaccions dels usuaris i analitzar el comportament de

l'audiència. De totes maneres, aquesta és una millora que valdria la pena analitzar amb deteniment, atès que pot provocar confusió si un usuari vol anar a una determinada pàgina, i l'enllaç el porta a una altra. També he utilitzat l'API i els *bots* de Telegram per automatitzar l'enviament d'ofertes, ampliant l'eficiència i rapidesa del codi.

En resum, la meva valoració personal sobre aquest treball és positiva. M'he proposat desenvolupar el projecte per saber la viabilitat del model, i ho he aconseguit. No he pogut fer-lo funcionar, però això era un objectiu secundari. He après que no és fàcil crear un negoci al voltant d'aquest model per diferents motius que n'han dificultat la viabilitat.

De totes maneres, retindré tot aquest coneixement i no descarto tornar-lo a emprendre més endavant, partint en un punt més avançat del que ho he fet aquesta vegada.

8 WEBGRAFIA I BIBLIOGRAFIA

Anders, V. (s. d.). *AUTOMATIZAR, radicación*. Etimologías de Chile - Diccionario Que Explica el Origen de las Palabras. Disponible a <https://etimologias.dechile.net/?automatizar>

About Selenium. (s. d.). *Selenium*. Disponible a <https://www.selenium.dev/about/>

Amazon afiliados. *Anexo Lista de Comisiones de la UE*. (s. d.-a). Disponible a <https://afiliados.amazon.es/help/operating/scheduleoct1>

Amazon Afiliados — *El programa de marketing de Afiliados de Amazon*. (s. d.-b). Disponible a <https://afiliados.amazon.es/>

Amazon API Gateway | API Management | Amazon Web Services. (s. d.). Amazon Web Services, Inc. Disponible a <https://aws.amazon.com/es/api-gateway/>

Brunete, San Segundo, Herrero (s. d.). *Introducción a la automatización industrial*. Disponible a https://bookdown.org/alberto_brunete/intro_automatica/

Condiciones de Uso y Venta - Servicio de atención al cliente de Amazon. (s. d.). Disponible a <https://www.amazon.es/gp/help/customer/display.html?nodeId=GLSBYFE9MGKKQXXM>

Cookie Information. (2022, 19 octubre). *¿Qué es una cookie? - Cookie Information*. Disponible a <https://cookieinformation.com/es/que-es-una-cookie/>

Forrest, B. (2023, 6 noviembre). *Make money with Amazon affiliate marketing: Expert Tips*. AMZPREP. Disponible a <https://amzprep-com.translate.goog/become-best-at-amazon-affiliate-marketing>

Is It Legal to Scrape Amazon and Never Getting Blocked | Octoparse. (2023, 18 agosto). Disponible a <https://www.octoparse.com/blog/is-it-legal-to-scrape-amazon-data>

Jurídicas, N. (2024, 26 enero). *¿Se acerca la desaparición de las cookies de terceros?* Noticias Jurídicas. Disponible a

<https://noticias.juridicas.com/conocimiento/tribunas/18720-iquest;se-acerca-la-desaparicion-de-las-cookies-de-terceros/>

Kiyosaki, R. T. (2011). *Rich Dad's cashflow quadrant: Guide to Financial Freedom*. Editorial Rich Dad.

Marquez, J. (2022, 19 junio). *El primer ordenador personal del mundo nació en un garaje en 1970: así era el Kenbak-1*. Xataka. Disponible a <https://www.xataka.com/ordenadores/primer-ordenador-personal-mundo-nacio-garaje-1970-asi-era-kenbak-1>

Moreno, A. (2023, 23 noviembre). *¿Qué es una URL y para qué sirve? Estructura y tipos existentes*. Atecnis. Disponible a <https://www.atecnis.com/url-que-es/>

Ntg. (2023, 29 marzo). «*Big Data, Big Results: How Analytics is Delivering Business Value*». Disponible a <https://www.linkedin.com/pulse/big-data-results-how-analytics-delivering-business-value-ntg-oman>

Plieshakov, A. (2023, 15 diciembre). *¿Cuándo se inventó el primer ordenador?* Book Down. Disponible a <https://www.info-computer.com/blog/cuando-se-invento-el-primer-ordenador.html>

Python Institute. (s. d.). *About Python*. Disponible a <https://pythoninstitute.org/about-python>

RapidAPI. (s. d.). *API Hub - Free public & Open rest APIs | Rapid*. Disponible a <https://rapidapi.com/hub>

RapidAPI. (s. d.). *Real-Time Amazon Data API Documentation (Letscrape-6bRBa3QGUO5) | RapidAPI*. Disponible a <https://rapidapi.com/letscape-6bRBa3QguO5/api/real-time-amazon-data/>

reCAPTCHA. (s. d.). *reCAPTCHA*. Disponible a <https://www.google.com/recaptcha/about/>

Sánchez, C. (2023, 2 agosto). *¿Qué es el marketing de afiliación y cómo funciona?* Awin. Disponible a <https://www.awin.com/es/marketing-de-afiliacion/que-es-el-marketing-de-afiliacion>

VPS | Servidores Virtuales a partir de 1€/mes | IONOS. (s. f.). 2001-2024 IONOS SE. Disponible a <https://www.ionos.es/servidores/vps>

What is CAPTCHA? - Google Workspace Admin Help. (s. f.). Disponible a <https://support.google.com/a/answer/1217728>

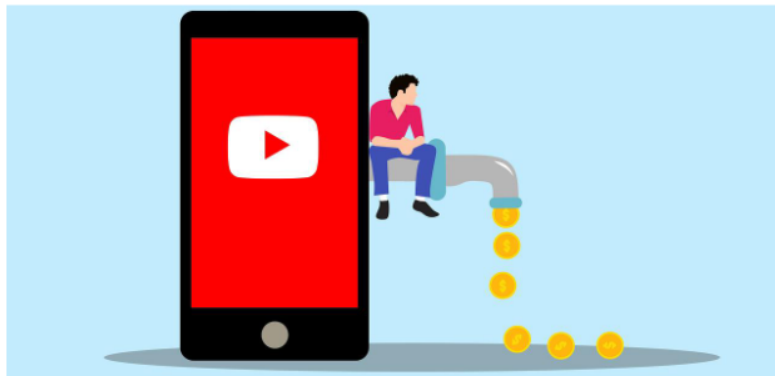
9 ANNEXOS

- Annex 1 prescendent del treball de l'assignatura Analítica Digital (4rt de Comunicació Interactiva, UAB)

Pau Cardús i Míriam Ardèvol

Els riscos ocults dels consells financers a YouTube

Què hi ha darrera dels vídeos sobre com guanyar diners? Són objectius educatius, o merament interessats? Anàlisi sobre els patrons d'aquesta tipologia de vídeos.



Gràfic 1: YouTube, una màquina de fer diners.

Amb el sorgiment de les noves plataformes digitals s'ha renovat el model de negoci que les acompanya. Estem parlant d'un model gratuït, on les dades i la presència de l'usuari són més importants que les subscripcions.

Per poder-ho entendre amb tota la seva complexitat, ens hem de preguntar com generen ingressos les plataformes com YouTube, Instagram, Facebook o X, que són gratuïtes. La majoria dels ingressos els aconsegueixen deixant fer publicitat a tothom qui ho vulgui. Com més interès hi hagi per part de les empreses a anunciar-se en alguna d'aquestes plataformes, més ingressos obtindrà.

Per aconseguir aquest interès esmentat, la plataforma necessita desenvolupar un parell d'aspectes, que són essencials. Per una banda, ha de conèixer l'usuari i saber tot el que pugui d'ell. L'exactitud a la segmentació se sol traduir com a bons resultats per les empreses anunciantes. Per l'altra banda, necessita usuaris que facin servir la plataforma. Sense ells, no es podrien mostrar els anuncis.

És, en aquest punt, quan entrem a la guerra entre plataformes per aconseguir que els usuaris hi passin temps. Aquestes estratègies, lògiques i polítiques entren dins del marc anomenat "Economia de l'atenció". En resum, és una guerra per obtenir la màxima retenció dels usuaris. I, com a tota guerra, sempre hi ha perjudicats. En aquest cas, els usuaris.

Partim que les plataformes tenen molta força sobre els hàbits dels consumidors. Són capaces d'enganxar, manipular, liderar opinions i moltes altres accions que es duen a terme sense que l'usuari se n'adoni.

Ara bé, realment emprenen aquestes accions? I si ho fan, fins a quin punt? Tenen límits ètics, o merament econòmics? Tot i que saber-ho és molt complicat, podem investigar aquestes plataformes per analitzar-ne el contingut que hi ha, i les estratègies que duen a terme.

Pau Cardús i Míriam Ardèvol

Ens centrem amb YouTube, el segon lloc web més visitat del món, després d'un altre lloc de la mateixa empresa, Google. És la plataforma de contingut audiovisual més important a nivell d'audiència. Tant és així, que gràcies o per desgràcia d'ella, van aparèixer els anomenats *youtubers*.

Tot i que s'humanitza molt el concepte, no deixen de ser figures líders d'audiència. La credibilitat amb el seu públic, fa que aquest li sigui lleial i un fidel creient amb gairebé tot el que digui. El consumidor no només se sent representat amb aquesta figura, sinó que s'adapta per assemblar-s'hi tot el possible.

Una de les maneres que tenim per comprovar que YouTube emprèn accions que afecten els usuaris, en conseqüència de la "guerra" per l'economia de l'atenció, és analitzar si permeten contingut perjudicial.

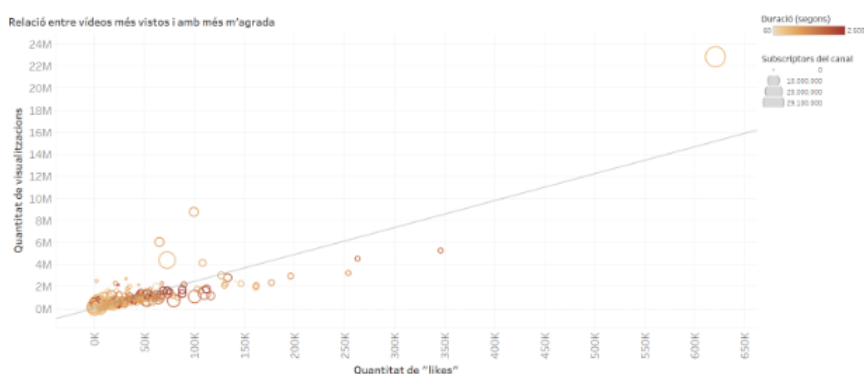
Es podrien estudiar molts continguts, tot i que n'hi ha un en especial que pot donar molt joc. Parlem dels vídeos que expliquen com guanyar diners.

Com guanyar diners, a YouTube

Aquest tema d'interès genera un dilema per Youtube i la protecció de l'usuari. No es pot prohibir un contingut d'aquest tipus, llevat que es consideri enganyós. I això és molt complicat, perquè no es pot du a terme una detecció automàtica. De totes maneres, serà el tema d'estudi que ens servirà per afirmar o contradir que a YouTube hi ha contingut disponible perjudicial.

Per això, farem servir les dades de vídeos (2019-2023) que continguin "como ganar dinero" en alguna part de la seva estructura, com pot ser el títol, descripció, subtítols... acompanyats d'altres mètriques referents al vídeo o al canal.

Per començar, hem procedit a fer una anàlisi general, estudiant els vídeos amb més repercussió.



Gràfic 2: Dispersió entre la quantitat de "likes" i visualitzacions dels vídeos analitzats.

Amb les visualitzacions a l'eix x, i els "likes" a l'eix y, podem trobar, ja d'entrada, una relació entre aquestes dues mètriques. Com més visites té un vídeo, més "m'agrada" aconsegueix. Això no és exclusiu d'aquesta temàtica, i és del tot normal amb la majoria dels vídeos de YouTube.

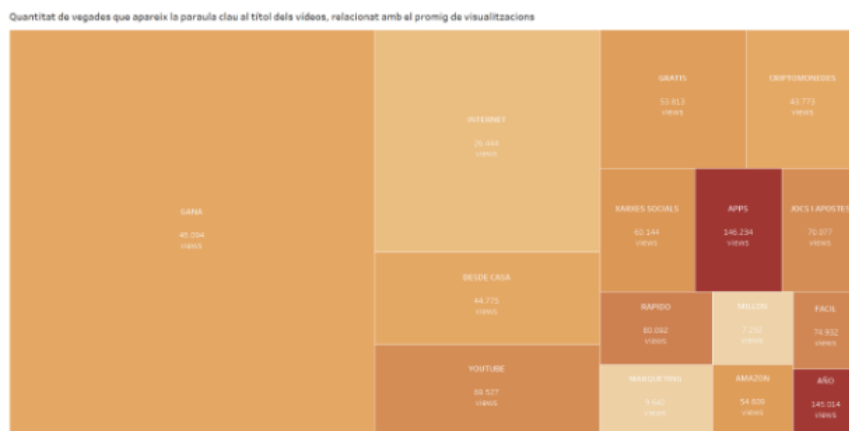
Pau Cardús i Míriam Ardèvol

Ara bé, analitzant el vídeo més vist, amb 22 milions de visites, tracta d'un concurs per aconseguir guanyar el màxim de diners possible durant 24 hores. Forma part del canal "Alejo Igoa". No té la intencionalitat d'ensenyar a guanyar diners, ni molt menys de manera perjudicial, però incita a invertir i emprendre accions per guanyar-ne. Això, ja ho hem comentat abans, no és pas dolent.

Tot i que, a continuació, trobem algun vídeo més amb to humorístic o d'entreteniment, ja comencen a aparèixer vídeos seriosos. El més vist d'aquesta tipologia és del canal "Emprende con Raul". Explica a guanyar diners des de YouTube sense fer vídeos amb un mètode secret. Només el títol ens porta a la següent conclusió: provoca que els usuaris entrin a veure el vídeo, ja que promet que no cal dedicar-hi temps, i que és un mètode secret.

És perjudicial per als usuaris, aquest contingut? Com que és un dels vídeos amb més visualitzacions i "likes" d'aquest estudi, hem anat a YouTube a comprovar-ho. Trobem que, ja d'entrada, destina cinc minuts a convèncer als usuaris que poden arribar a guanyar molts diners. Ho fa amb un missatge esperançador ensenyant casos d'èxit, on s'arriben als milions de guanys. El mètode que promet ser secret, és una estratègia força instaurada dins del món de l'automatització. Pel que fa als comentaris, trobem que molts agraeixen la sinceritat del *youtuber*, fet que ens porta a pensar que la quantitat de "likes" que té es deu, en gran mesura, a la proximitat que mostra.

La resta de vídeos d'aquest primer gràfic són molt variats, tot i que la majoria explica a guanyar diners d'alguna manera o altra. Per poder analitzar aquesta massa de vídeos i saber de què tracten, hem procedit a la següent anàlisi.



Gràfic 3: Quantitat de vegades que apareixen les paraules en qüestió als títols dels vídeos analitzats.

Havent inspeccionat les temàtiques dels vídeos, hem generat aquestes subcategories. Són elements que apareixen al títol, i ens donen a entendre de què va. Més del 75% dels vídeos analitzats estan dins d'aquestes categories, per la qual cosa són resultats fiables.

Representat amb tamany, tenim la quantitat de vídeos que contenen la paraula clau. Amb color, el promig de visualitzacions que han obtingut. Quan un vídeo té dues o més paraules clau al títol, es representa a la categoria més específica. S'ha procedit a una classificació de genèric a específic.

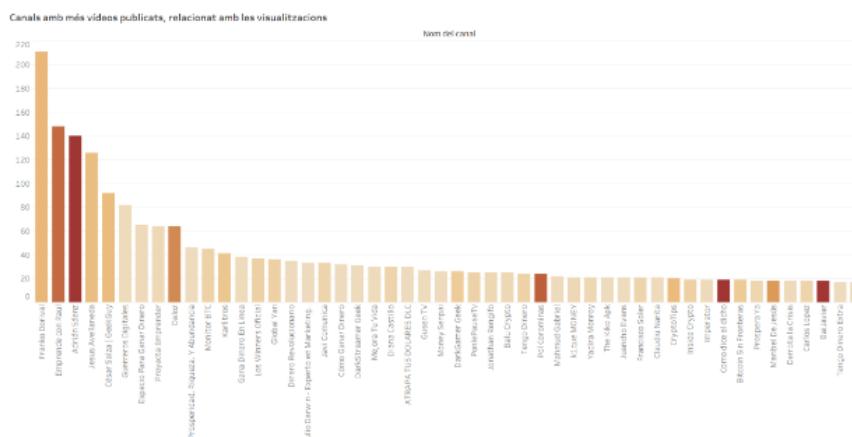
Pau Cardús i Míriam Ardèvol

Predomina, amb gran distància, els vídeos que contenen “gana” al títol. Això també inclou els que porten “ganar”. Continua el rànquing “internet”, “desde casa” i “youtube”. Les que tenen més promig de visualitzacions, però, no són pas aquestes. “Apps” (sent també “aplicaciones”) i “año” són les categories que ho lideren.

En resum, gràcies a aquest gràfic podem extreure la següent conclusió. Molts vídeos opten per oferir facilitats com a forma d'atracció, prometent guanyar diners des d'Internet, des de casa, amb YouTube, de manera gratuïta, amb xarxes socials, criptomonedes, aplicacions... Tots aquests conceptes enganxen i esdevenen títols pescaclics.

Tot i que considerem aquesta metodologia com a enganyosa, som conscients que s'aplica a tots els àmbits i categories, i no és exclusiu d'aquesta temàtica.

Ara ens centrarem a analitzar els canals amb més repercussió. Primer, hem generat un gràfic amb els 50 canals que han publicat més vídeos amb la paraula clau “como ganar dinero”. Amb el color, hem ordenat les visualitzacions que han obtingut. El resultat ha sigut el següent.



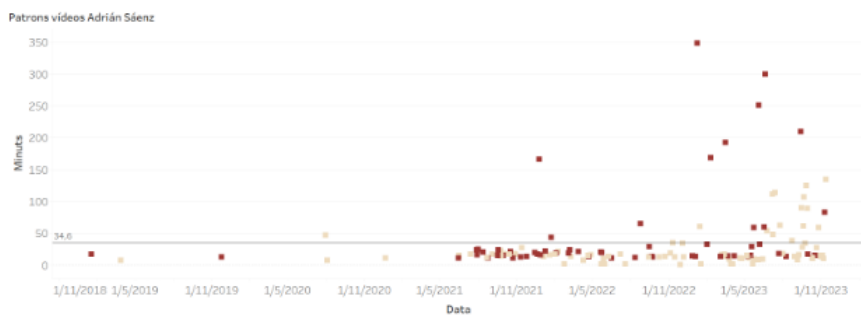
Pau Cardús i Míriam Ardèvol

promig se situa vora els 15 minuts. Això és així per poder monetitzar el vídeo amb totes les condicions possibles a YouTube.

Per tant, tenim un primer patró. Sol fer vídeos més llargs de 10 minuts per a poder-los monetitzar. Per altra banda, hem trobat estructures similars als títols. Focalitza l'èmfasi amb el guany econòmic ("ganar", "dinero", "invertir..."), especifica xifres exactes ("gana 47.350€ al mes") que generen expectativa i credibilitat, fa servir l'experiència pròpia ("he probado", "he ganado") i, finalment, ofereix instruccions i tutorials ("cómo ganar", "cómo invertir").

Finalment, volem comprovar si els vídeos contenen enllaços a la descripció, que puguin redirigir l'usuari cap a una plataforma i així generar trànsit d'afiliació. Hem decidit aplicar aquest criteri, perquè l'Adrián no posa enllaços a les seves xarxes socials, cosa que distorsionaria els patrons.

Els hem combinat i, representats, ens ofereixen el següent resultat. Els punts de color fosc són els que compleixen amb el patró assenyalat. Els clars, no. Per començar, podem afirmar que ho segueix una bona part dels seus vídeos.



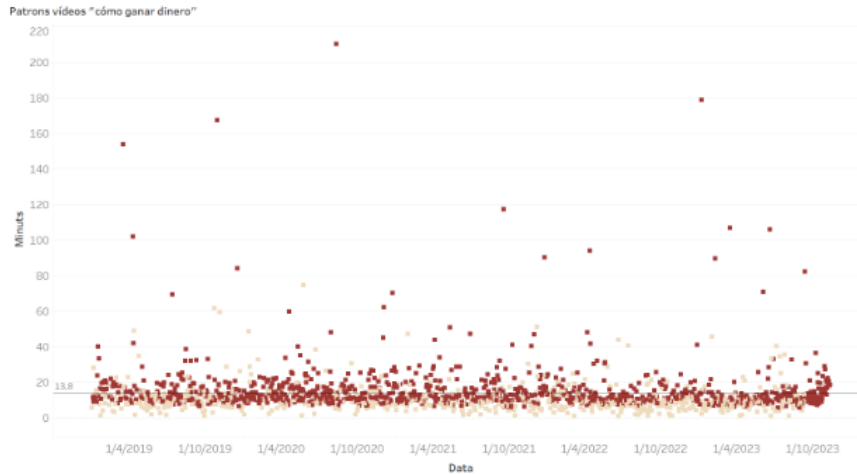
Gràfic 5: Evolució temporal dels vídeos publicats per Adrián Sáenz relacionat amb la duració. Amb color, si els vídeos segueixen el patró analitzat.

Per tant, podem afirmar que sol seguir un patró. Vídeos que capten l'atenció, generen trànsit, duren més de 10 minuts i contenen enllaços a la descripció. Això ens porta a pensar que l'interès d'aquest canal és generar ingressos des de YouTube i amb programes d'afiliació.

La pregunta és, a hores d'ara, si això és perjudicial per a l'usuari. Segurament no ho és tant com ho poden ser continguts de violència o sexe. Ara bé, hem de tenir en compte que, si el seu objectiu no se centra en informar correctament els usuaris, aquests poden realitzar inversions arriscades o desinformades. També és perjudicial si el canal recomana fer unes inversions interessades als programes d'afiliació.

Tot i que ens hem centrat en l'Adrián, aquests patrons són força habituals amb tots els vídeos analitzats. A continuació, presentem el mateix gràfic, però sense restringir-ho a cap canal.

Pau Cardús i Míriam Ardèvol



Gràfic 6: Evolució temporal de tots els vídeos publicats relacionats amb la duració. Amb color, si els vídeos segueixen el patró analitzat

Fins a quin punt YouTube ha de prohibir el contingut que segueixi patrons similars? Considerem que hi ha d'haver un control que permeti protegir a l'usuari d'aquest tipus de contingut. Es podria definir com a estafa o contingut brossa, generat per augmentar les visualitzacions sense tenir com a objectiu informar a l'espectador.

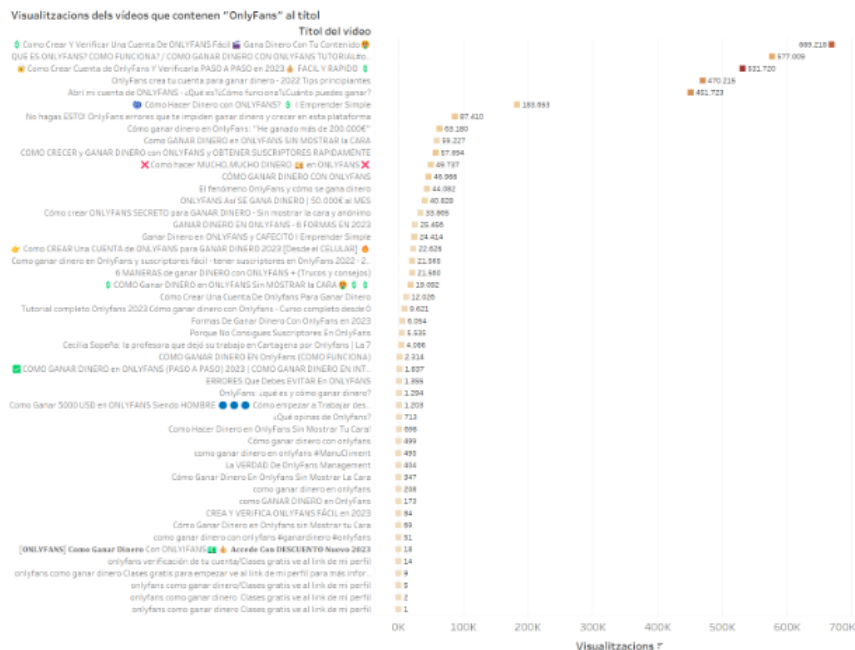
No tot acaba aquí, però. Si bé aquest tipus de contingut pot generar diferents opinions al respecte, hem volgut profunditzar una mica més, buscant contingut disponible a la plataforma i que sigui clarament perjudicial per a l'usuari. La primera mostra és la següent, on s'ensenya a guanyar diners de manera fàcil i ràpida enganyant els subscriptors de la mateixa plataforma.



Gràfic 7: Títols dels vídeos considerats com a perjudicials pels usuaris.

Aquests no són casos puntuals. Amb un títol més o menys explícit, el contingut perjudicial està normalitzat i no es bloqueja a Youtube. Entrem al cas particular d'OnlyFans, una plataforma d'imatges compromeses a canvi de subscripcions. Hi ha molt contingut basat en explicar com crear un compte, guanyar diners "fàcilment i ràpidament" i "sense ensenyar la cara".

Pau Cardús i Míriam Ardèvol



Gràfic 8: Títols dels vídeos que contenen "OnlyFans" o "Only Fans" al títol, relacionats amb les visualitzacions. Amb color, la quantitat de "likes".

Gràcies a aquesta visualització, podem veure que els primers resultats prometen guanyar diners, i fins i tot expliquen com crear un compte i començar. Considerem que és un greu problema, ja que no es tracta d'incitar a efectuar una falsa inversió. Va més enllà d'una qüestió econòmica, i supera el límit de la dignitat individual. Un cop s'accedeix a penjar contingut explícit d'un mateix a internet, ja queda exposat per sempre.

Conclusions

Com a mode de conclusió, i gràcies a l'anàlisi efectuada, podem afirmar que hi ha dos tipus de contingut relacionat amb la paraula clau "com guanyar diners". El primer tipus és un contingut totalment honest i amb intencionalitats informatives. S'explica com generar riquesa, i no ho considerem perjudicial.

Ara bé, el segon tipus de contingut atempta completament contra la seguretat dels usuaris. Estem parlant d'enganyifes, apostes, falses inversions i incentius a accedir al món pornogràfic amb fins econòmics.

YouTube hauria de controlar tot aquest contingut, ja que sobrepassa la línia vermella que protegeix els usuaris.

Pau Cardús i Míriam Ardèvol

Webgrafia i bibliografia

- Los vídeos de los adolescentes en YouTube: Características y vulnerabilidades digitales. (s. f.). Comunicar. https://e-aules.uab.cat/2023-24/pluginfile.php/522409/mod_resource/content/0/1988-3293-comunicar-26-54-61.pdf
- Johnson, S. (2009). *Si és dolent t'ho recomano*.
- Adián Sáenz. YouTube. <https://www.youtube.com/@AdrianSaenz>
- Emprende con Raul. YouTube. <https://www.youtube.com/@EmprendeconRaul>
- Alejo Igoa. (2022, 22 julio). QUIEN PUEDE CONSEGUIR MAS DINERO EN 24 HORAS?!! [Vídeo]. Youtube. <https://www.youtube.com/watch?v=cLDODT5iZY>