



This is the **published version** of the bachelor thesis:

Roura Jiménez, Oriol; Benítez Fernández, Yolanda, dir. Seguidor i comparador de precios de diferentes productos con Copilot. 2023. (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/298946>

under the terms of the  license

Seguidor i comparador de preus de diferents productes amb Copilot

Oriol Roura Jiménez

Resum— L'objectiu principal del projecte és crear una pàgina web de seguiment de preu de productes. A diferència d'altres sistemes, el sistema proposat permet comparar aquest seguiment en diverses plataformes diferents al mateix temps.

Les principals funcionalitats del sistema seran tres, fer seguiment dels productes desitjats per l'usuari (el seguiment començarà en el moment que es demani i s'enregistraran com a màxim 13 mesos), mostrar un gràfic amb les variacions de preus i mostrar els preus actuals dels productes.

S'executarà a Google Cloud aprofitant la versió gratuïta d'un any que ofereix Google.

Paraules clau—token, maquina virtual, mv, Google Cloud, scrapping, llibreries, front-end, back-end, base de dades, API, Type Script, Angular, SQLite, web, CSS, HTML.

Abstract— The main objective of the project is to create a product price tracking website. Unlike other systems, the proposed system allows comparing this tracking on several different platforms at the same time.

The main functionalities of the system will be three, tracking the products desired by the user (the tracking will begin at the moment it is requested and a maximum of 13 months will be recorded), showing a graph with price variations and showing the current prices of the products.

It will be executed on Google Cloud taking advantage of the trial version of one year offered by Google.

Index Terms— token, virtual machine, VM, Google Cloud, scrapping, libraries, front-end, back-end, data base, API, Type Script, Angular, SQLite, web, CSS, HTML.



1 INTRODUCCIÓ

El projecte desenvolupa una aplicació web de seguiment de preus de productes electrònics a través de diferents botigues web. Ens recolzarem amb l'assistència de Copilot i l'entorn d'execució de Google Cloud.

Principals funcionalitats: El sistema proposat permet fer seguiment del preu dels productes que es desitgin en tres plataformes prèviament definides, mostrar un gràfic amb les variacions de preus dels últims 13 mesos i avisar en cas d'alarmes configurades per l'usuari. Per totes aquests funcionalitats caldrà que l'usuari estigui registrat a l'aplicació.

El paper de Google Cloud: Google Cloud és una plataforma que agrupa un conjunt de serveis de computació en núvol públic, incloent serveis allotjats de computació, emmagatzemament i desenvolupament d'aplicacions que son

executats en hardware de Google (serveis IaaS, PaaS o SaaS). Els seus serveis poden ser accedits per desenvolupadores de software, administradores de núvol i altres professionals de TI a través de la Internet o bé d'una connexió de xarxa dedicada.

Google ofereix una subscripció per proves amb un crèdit de 300\$ que caduca als 90 dies.

Farem servir aquesta subscripció per executar l'aplicació de seguiment de preus i que sigui accessible des de qualsevol lloc.

El paper del Copilot: El Copilot és un company d'intel·ligència artificial que col·labora amb els usuaris per proporcionar informació, respondre preguntes i participar en converses. Aquesta eina utilitza tecnologies com GPT-4 i Bing Search per oferir respostes rellevants i útils.

Hi ha diferents eines que ajuden amb intel·ligència artificial. La primera que va sortir és ChatGPT, posteriorment Microsoft va treure Copilot integrat amb el cercador de Bing però també tenim Copilot per Microsoft365, que ajuda amb les eines d'Office, Copilot Studio, que ajuda a crear bots o Copilot per GitHub, que ajuda en el desenvolupament de codi.

En concret el que volem provar és l'ajuda que pot

- E-mail de contacte: oriol.roura@autonoma.cat
- Menció realitzada: Enginyeria del Software
- Treball tutoritzat per: Yolanda Benítez (departament)
- Curs 2023/24

proporcionat Copilot per GitHub en el desenvolupament del projecte.

1.1 MOTIVACIÓ

El Black Friday, celebrat el divendres següent al Dia d'Acció de Gràcies, va aparèixer quan els comerços van començar a oferir grans descomptes i promocions per incentivar les compres nadalenques, buidar l'estoc per al canvi de temporada i, a més, allargar el període de compres.

Aquesta tradició continua, però pensem que hi ha molts productes als que se'ls puja els preus abans d'aquest esdeveniment. D'aquesta manera, quan arriba el dia, es poden anunciar grans descomptes que en realitat no ho són tant.

Amb aquest projecte volem ajudar a fer un seguiment al llarg de l'any dels canvis de preus en diferents webs per així poder veure si el descompte que estan oferint les promocions és realment el que diuen. Mostrarem estadístiques de 13 mesos enrere per poder comparar, també, amb els preus del mateix període de l'any anterior.

2 OBJECTIUS A ASSOLIR

En aquest apartat expliquem els diferents objectius que esperem assolir en aquest projecte.

L'objectiu principal és desenvolupar una aplicació web capaç de comparar de forma efectiva i precisa els preus d'un producte que s'ha posat en seguiment al llarg del temps a tres botigues diferents. Es podrà detectar si durant temporada de rebaixes els preus estan tant descomptats com anuncien.

A més tenim altres objectius com:

- Comprovar i avaluar l'assistència del Copilot per GitHub en el desenvolupament de codi.
- Millorar les competències en diferents àmbits de la informàtica. Com ara en el desenvolupament amb diferents llenguatges web com TypeScript, HTML5 i CSS, seguint les bones pràctiques i els estàndards de qualitat.
- Desplegar una aplicació web en un servei al núvol com ara GoogleCloud i que sigui accessible des de la xarxa.

3 ESTAT DE L'ART

En aquesta part expliquem les característiques principals del projecte respecte d'altres projectes ja existents amb el mateix propòsit.

Hem analitzat diverses aplicacions similars a la que es vol desenvolupar per a veure la situació en el mercat. Les característiques analitzades inclouen segmentació de preus, gràfics temporals, segmentació de productes, comparador de preus, característiques de cerca per URL, cerques per nom i permisos d'usuaris de la UE.

Les pàgines trobades inclouen la major part d'aquestes

característiques però cap d'elles les inclou totes. El que més es troba a faltar és la comparativa entre diferents botigues web. De fet, les que són més completes només fan seguiment de preus a Amazon.

Tenint en compte el que hem trobat, veiem que és molt interessant oferir la possibilitat de comparar el mateix producte a més d'una botiga. En el nostre cas, compararem el preu del producte a tres botigues.

En la secció A1 de l'apèndix es pot trobar la taula utilitzada per a fer l'anàlisi

4 METODOLOGIA

Una vegada repassats els objectius, expliquem quina metodologia hem dut a terme per a desenvolupar el projecte.

Per a l'execució d'aquest projecte, hem adoptat un enfocament de planificació adaptable o gestió de projectes híbrida. Aquest enfocament combina elements de la planificació detallada amb la flexibilitat per adaptar-se als canvis i desafiaments que poden sorgir durant l'execució del projecte.

Aquesta metodologia es basa en dos components principals:

Planificació Inicial: Al començament del projecte es realitza una planificació general que cobreix totes les fases i aspectes del projecte. Aquesta planificació inclou la definició dels objectius del projecte, la identificació de les tasques necessàries per assolir aquests objectius, l'assignació de responsabilitats i l'estimació dels terminis i recursos necessaris.

Flexibilitat i Adaptabilitat: Tot i que es compta amb una planificació general, es reconeix que els projectes poden enfrontar-se a canvis i incerteses. Per tant, la metodologia permet la flexibilitat per adaptar-se a aquestes circumstàncies. A mesura que es progressa en l'execució del projecte, es reavaluen i ajusten els plans en funció de la informació i els resultats que s'obtenen. Això pot implicar la modificació de les tasques, la reassignació de recursos o la revisió dels terminis, sempre amb l'objectiu de garantir que el projecte es mantingui en el camí correcte cap a l'assoliment dels seus objectius.

Pensem que aquest enfocament híbrid és l'adequat pel projecte perquè, per una banda, la planificació inicial està definida ja que té especificacions clares, data d'entrega definida i el projecte no és tant complex per requerir una metodologia dinàmica, per l'altra banda, pensem que cal un cert nivell de flexibilitat perquè mai hem treballat amb scrapping ni amb Google Cloud i no tenim experiència en treballar amb APIs.

5 PLANIFICACIÓ

La planificació és important per determinar la duració del projecte i comprovar que és assolible.

El projecte es divideix en 4 grans fases: posada en contacte, estudi del projecte, desenvolupament del projecte i documentació.

La primera fase de posada en contacte és per començar a definir la proposta i veure si és viable i tenir un primer contacte amb el/la tutor/a per fer-li la proposta.

La proposta sorgeix del dubte inicial de si les baixades de preu en el Black Friday son reals o no.

Amb el coneixement adquirit en programació d'aplicacions web i amb la informació que hem trobat sobre aquest tema a Internet hem fet una proposta que pensem que és assequible i viable.

La proposta s'envia al coordinador de la menció que és qui s'encarrega de buscar algú interessat en fer el seguiment del projecte. Amb el tutor/a assignat/a fem la primera reunió per explicar el projecte de forma més definida.

La segona fase, estudi del projecte, consisteix en aprofitar el feedback del tutor/a per re-definir el projecte i acabar d'especificar totes les incerteses prèvies. Durant aquesta etapa es comprova com d'únic és aquest projecte, es busca quines eines s'utilitzaran, quins coneixements seran necessaris i es defineix la durada total i de cada tasca del projecte.

La tercera fase, desenvolupament del projecte, consisteix en seguir la planificació presentada prèviament i desenvolupar el projecte mentre s'ajusten situacions inesperades. En aquesta fase incloem tots els jocs de proves de les diferents funcions. Aquesta és la fase més llarga del projecte ja que és en la que es desenvolupa tot i en la que poden sorgir imprevistos. Aquesta fase, a la seva vegada, es divideix en tasques més petites i senzilles (tests inicials, base de dades, back-end, front-end, scrapping, pujada al núvol).

L'última fase, documentació, consisteix en unificar, acabar i polir la documentació per crear la memòria y la presentació. La documentació es va desenvolupant al llarg del projecte i és en aquesta etapa on s'ajunta tota per a crear aquests dos documents i solucionar qualsevol error o inconsistència.

El diagrama de Gantt amb la planificació detallada es pot trobar a l'apèndix A2.

6 ANÀLISI

A continuació mostrem els requisits funcionals, no funcionals i quina estructura tindrà tant l'aplicació (diagrama de casos d'ús) com la base de dades.

6.1 REQUISITS FUNCIONALS

Pel desenvolupament del projecte hi ha un seguit de funcionalitats claus necessàries: seguiment de preus i llistes de seguiment, visualització gràfica de variacions de preus, alertes de canvis significatius en els preus, gestió d'usuaris.

(Enumeracions)

Tenint en compte aquestes funcionalitats hem obtingut els següents requisits funcionals:

- El sistema ha de permetre als usuaris fer seguiment del preu dels productes electrònics en tres plataformes o botigues específiques.
- Els usuaris poden afegir productes a la seva llista de seguiment especificant la URL del producte i la plataforma.
- El sistema ha de mantenir un històric dels preus dels productes al llarg del temps.
- L'aplicació ha de mostrar un gràfic que representi les variacions de preus dels productes seleccionats durant els últims 13 mesos.
- El sistema ha d'enviar alertes als usuaris quan es produeixen canvis significatius en els preus dels productes.
- Els usuaris poden configurar els seus propis límits d'alarma per a cada producte.
- Els usuaris han de poder registrar-se a l'aplicació.
- Només els usuaris registrats poden accedir a les funcionalitats de seguiment i configuració d'alarmes.

6.2 REQUISITS NO FUNCIONALS

Els requisits no funcionals que delimiten i acaben de definir el projecte son els següents:

- L'aplicació s'ha d'executar utilitzant Google Cloud.
- L'aplicació ha de garantir la seguretat de les dades dels usuaris i la confidencialitat dels preus dels productes.
- Cal implementar autenticació i autorització adequades per protegir les dades.
- L'aplicació ha de ser eficient i proporcionar una resposta ràpida als usuaris.
- Cal optimitzar les consultes a la base de dades per preveure gestionar grans quantitats de dades de preus.
- L'aplicació ha de ser intuïtiva i fàcil d'utilitzar per als usuaris.
- Cal proporcionar una interfície d'usuari clara i ben dissenyada.
- El gràfic ha de ser fàcilment llegible i proporcionar una visió clara de les tendències de preus.

6.3 DIAGRAMA DE CASOS D'ÚS

El diagrama mostra les accions que podrà realitzar l'actor.

Podrà registrar-se, si ja té registre podrà iniciar sessió, afegir o seleccionar productes, veure els preus actuals dels seus productes i veure estadístiques.

A l'apèndix A3 s'inclou el diagrama d'aquests casos d'ús.

6.4 ESTRUCTURA DE LA BASE DE DADES

La base de dades està formada per 6 taules: usuaris, productes, botigues, usuaris_productes, productes_botigues i històric_preus.

Usuaris: Conte la informació de cada usuari registrat, amb el seu nom, correu electrònic, identificador i contrasenya encriptada.

Productes. Petita taula amb el nom i identificador de cada producte en seguiment.

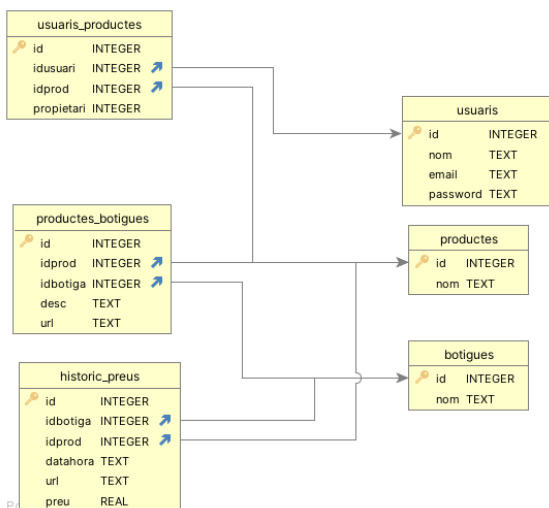
Botigues. Inclou les botigues que s'estan seguint. Per a aquest projecte només hi haurà tres (MediaMarkt, Amazon i NewEgg).

Usuaris_productes. Taula que combina els usuaris i els productes, hi haurà un entrada per a cada producte que té cada usuari, permet connectar els productes en seguiment amb els usuaris que el segueixen.

Productes_botigues. Un producte es segueix en cada botiga i té diferent informació en cada cas. Aquesta taula ens serveix per mostrar les dades d'un producte a cada botiga.

Històric_preus. Taula encarregada d'emmagatzemar els preus que anem veient de cada producte a cada botiga i la data en que s'han vist.

L'esquema de la base de dades és el següent:



Il·lustració 1: esquema de la Base de Dades

7 EINES

Tenint l'anàlisi i els requeriments dels projectes, a continuació expliquem quines eines s'han utilitzat per al seu desenvolupament.

Aquestes eines es podrien dividir en els següents grups: llenguatges de programació, entorns de desenvolupament, motors d'execució, suport, documentació i seguiment, llibreries i base de dades.

7.1 LLenguatges de Programació

Els llenguatges de programació requerits per aquest projecte són diversos degut a que han de cobrir diferents aspectes del desenvolupament.

Per al servidor i les APIs, hem fet servir TypeScript i JavaScript, amb SQL per a les consultes a la base de dades.

La gestió de la base de dades es fa amb SQL.

En quant al front-end, l'hem desenvolupat amb el framework Angular, emprant HTML per estructurar la pàgina web, TypeScript per les classes i funcions, i CSS pel disseny i estil visual.

7.2 ENTORNS DE DESENVOLUPAMENT

Hem utilitzat dos entorns per a desenvolupar aquest projecte: Visual studio code i DB browser (SQLite).

Visual studio code per al desenvolupament del servidor, de les APIs i de tot el front-end.

DB browser per al disseny, desenvolupament i seguiment de la BD.

7.3 MOTORS D'EXECUCIÓ

Per al desenvolupament de la part del servidor, hem utilitzat Nodemon. Nodemon gestiona l'execució de Node.js i permet mantenir un servidor actiu i garantir que s'actualitzi automàticament quan es modifiqui qualsevol dels seus fitxers. D'aquesta manera s'optimitza el procés de desenvolupament i es facilita la tasca de programació.

Per a producció hem utilitzat pm2 com a gestor de l'execució de Node.js ja que facilita l'inici, reinici o aturada dels programes.

Per a llençar l'script que fa l'scrapping de les botigues i obté els preus diàriament fem servir el cron de la màquina, d'aquesta manera es pot canviar fàcilment la data d'execució o el número de vegades que volem que s'executi cada dia i aprofitem el servei que ja ens ofereix el propi sistema operatiu.

Finalment, gestionem el client amb un servidor Apache, donant accés a l'aplicació a qualsevol usuari d'Internet que estigui registrat a l'aplicació.

7.4 SUPORT

Com a suport per a aquest projecte, a més del propi ordinador personal, hem utilitzat les eines de Google Cloud i el suport de Copilot per GitHub.

Copilot per GitHub s'ha utilitzat per a que assisteixi en el desenvolupament del projecte i de tot el seu codi.

GoogleCloud ha proporcionat les eines necessàries per a poder pujar el servidor al núvol i que pugui funcionar correctament com una pàgina web un cop pujat.

7.5 DOCUMENTACIÓ I SEGUIMENT

Per documentar i fer un seguiment del projecte, hem fet us dels següents programaris: Project, Word, PowerPoint, Excel i GitHub.

Project (Office project). Per la planificació del TFG i per fer el seguiment de l'estat actual respecte a aquesta planificació.

Word i PowerPoint. Per a la documentació, la memòria i la presentació del TFG.

Excel. Per a crear taules, gràfiques i diferents càlculs.

GitHub. Per desenvolupar, guardar el desenvolupament del projecte de forma segura en diferents versions i per poder fer servir Copilot.

7.6 LLIBRERIES

Per implementar el projecte utilitzem diverses llibreries. Les principals són les següents:

- Axios: una llibreria basada en promeses per a realitzar peticions HTTP, que permet treballar tant en el navegador com en node.js.

- Cheerio: una llibreria ràpida, flexible i lleugera per a analitzar HTML i treballar amb el DOM.

- Express: un framework per a aplicacions web de node.js, dissenyat per a construir aplicacions web i APIs de manera ràpida i senzilla.

- bootstrap: potent eina de frontend que utilitza sass per a permetre dissenyar la teva pàgina de forma ràpida i simple

- ng2-charts: llibreria que proporciona directives per a diversos tipus de gràfics, com ara línia, barra, pastís, radar

7.7 BASE DE DADES

Aquest projecte utilitza la base de dades SQLite. Aquesta base de dades proporciona una solució eficient per a muntar una base de dades simple incloent tots els sistemes essencials requerits. La seva simplicitat i facilitat d'ús són ideals per als entorns de desenvolupament ràpid i prototipatge.

En aquest cas, hem optat per aquesta base de dades perquè creiem que el projecte no necessita una de més complexa.

8 DESENVOLUPAMENT

Després de totes les definicions i consideracions prèvies, descriurem tot el període de desenvolupament del projecte i els seus resultats.

El desenvolupament es divideix en les següents parts: proves de concepte inicials, desenvolupament de la BD, desenvolupament del Back-end, desenvolupament del Front-end, Desenvolupament del scrapping i pujada el sistema al Google Cloud.

8.1 PROVES DE CONCEPTE INICIALS

Degut al desconeixement sobre alguns aspectes de la proposta vam decidir fer unes proves de concepte per verificar que el projecte era raonable. Es van fer proves sobre la part d'scrapping perquè no havíem treballat simulant un navegador web, sobre APIs perquè tot i que ho havíem vist a classe, no teníem experiència i proves amb Angular per agafar familiaritat amb el llenguatge de programació.

Scrapping: En primer lloc vam crear un petit script de scrapping ja que era l'entorn més desconegut. Aquest script obtenia el nom, preu i descripció d'un producte d'Amazon fent un get a la url amb la llibreria Axios i obtenint els camps específics dins l'html amb la llibreria Cheerio. Va funcionar sense complicacions.

API: També vam crear un petit script per provar el desenvolupament amb APIs. Vam crear un procés que escolava pel port 3000 del localhost i retornava un codi 200 quan rebia una petició get (get <http://localhost/api>).

Angular: Finalment vam crear un projecte senzill d'Angular que al clicar un botó fes una crida a l'api anterior.

8.2 DESENVOLUPAMENT DE LA BD

Per a desenvolupar la BD vam decidir utilitzar una llibreria anomenada SQLite. Aquesta llibreria permet crear bases de dades bàsiques de forma ràpida i simple.

Amb aquesta llibreria només vam necessitar passar al Copilot l'estructura plantejada, que ens retornés les comandes necessàries per a crear les taules i relacions i fer una última revisió per acabar d'arreglar algunes incoherències.

Crear la base de dades va ser un dels processos a on vam poder veure la utilitat del Copilot.

8.3 DESENVOLUPAMENT DEL BACK-END

Amb la BD creada, el següent pas va ser crear el Back-end, encarregat de nodrir la base de dades (BD) a partir de crides externes rebudes per diverses Api.

En primer lloc vam crear la connexió a la BD i vam obrir un port (el 3000) per escoltar crides.

Posteriorment, vam fer us del Copilot per a que creés de forma automàtica les crides que considerés bàsiques per a

cada taula. També vam crear, amb ajuda de Copilot, un sistema de testing de cada API utilitzant una BD en memòria.

A continuació vam crear unes poques APIs específiques necessàries per situacions puntuals del Front-end i finalment vam crear un sistema de verificació d'usuaris, retornant un token després de cada crida. Vam definir la validesa del token en una hora.

8.4 DESENVOLUPAMENT DEL FRONT-END

Per a desenvolupar el Front-end vam crear un projecte d'Angular en TypeScript. Vam estructurar tot l'esquelet del projecte, creant tots els components necessaris per a les pàgines, els mòduls per agrupar els components i enrutant tots els components en el router d'Angular.

També vam dissenyar una estructura d'APIs per simplificar les crides amb un servei general i un servei per cada component. El servei d'API general (GeneralApi.service) és l'encarregat de fer les crides i hi ha un servei d'APIs per cada component (getItemApi.service, getListApi.service...) que crida al servei general passant-li tota la informació processada.

Una vegada creat l'esquelet del Front-end vam desenvolupar els diferents components, afegint les funcionalitats necessàries a cada pàgina i creant l'html bàsic (en aquest moment no tenim en compte el disseny).

En aquest punt, vam considerar l'opció de canviar el Front-end a Vue ja que s'estava explicant al temari d'una de les assignatures del curs, però després de començar a traduir el codi existent d'Angular a Vue vam arribar a la conclusió que el millor era continuar amb Angular ja que migrar a Vue era més complex del que esperàvem.

Seguidament vam començar a afegir les crides a les APIs tipus get i a polir el Front-end segons les dades que anàvem rebent. Vam desenvolupar els següents components:

- El component general que serveix per passar paràmetres a la resta de components.
- El component de la pàgina de la llista de productes de l'usuari.
- Les crides al component de la pàgina de login (les crides no eren funcionals perquè encara no s'havia afegit el sistema d'autenticació).
- El component d'items afegint les crides a l'API que recupera els productes seleccionats.
- El component amb la gràfica dels productes.

Seguidament vam desenvolupar el sistema d'autenticació. Consisteix en la pàgina de login per a la identificació de l'usuari i el sistema de token que s'envia a les APIs per verificar en tot moment que l'usuari està autoritzat a fer les diferents crides a l'API.

A continuació vam acabar de desenvolupar la pàgina d'afegir productes en seguiment, cridant a les APIs

necessàries i afegint la característica de poder eliminar productes des de la llista.

Finalment vam aplicar el disseny (CSS) al client.

8.5 DESENVOLUPAMENT DEL SCRAPPING

En aquest punt ja teníem una pàgina web funcional en local i havíem de desenvolupar el sistema per a recopilar les dades, un script que ha de recorre diàriament els elements (productes en seguiment) de la base de dades i obtenir el seu preu actual.

Al començar la implementació d'aquesta part vam veure que el petit script (amb Axios) de la prova de concepte ja no funcionava, Amazon el considerava un bot i li denegava l'accés. Després de buscar més documentació sobre l'scrapping i provar diferents mètodes per fer sol·licituds que semblessin el més "reals" possibles, finalment vam optar per afegir, a la crida d'Axios, unes capçaleres html que simulessin un navegador (user-Agent, Accept, Accept-Language, Accept-Encoding).

Amb el scrapping d'Amazon ja funcionant, quedaven les dues altres botigues web (MediaMarkt i PC Componentes), però vam comprovar que aquest sistema, que valia per Amazon, no funcionava a cap dels dos. MediaMarkt seguia considerant que era un bot i denegant l'accés i PC Componentes sempre retornava un error abans de carregar la pàgina i això provocava que Axios només retornés l'error i no el contingut de la pàgina.

Per solucionar aquests dos problemes vam decidir utilitzar Selenium, una llibreria que, en lloc de fer les crides directes, obre un navegador per tal que la sol·licitud fos realment d'un navegador.

Utilitzant aquesta llibreria ja teníem els tres scripts per obtenir els preus dels productes de les tres botigues.

A continuació vam crear un programa que s'executava automàticament per tal que, cada dia, obtingui tots els productes de la BD, cridi als tres scripts per cada producte, i guardi el resultat a la BD.

En aquest moment, el projecte ja funcionava completament en local.

8.6 PUJAR EL SISTEMA A GOOGLE CLOUD

Tenint el sistema funcionant en local, havíem de pujar-ho al núvol i fer-ho accessible des de Internet.

Per poder treballar amb Google Cloud ens vam crear un compte a aquesta plataforma amb un correu creat pel projecte.

Amb el compte de Google Cloud vam crear una màquina virtual (MV) a on es fa córrer tot el projecte i li vam assignar una IP fixa per a que fos accessible des de l'Internet.

Amb la MV muntada i executant-se, mitjançant la connexió ssh que ofereix Google Cloud, vam pujar tots els fitxers comprimits, els vam descomprimir i els vam

organitzar dins la MV.

Per posar en funcionament el Front-end vam instal·lar Apache i vam fer el build del Front-end. Els fitxers d'execució obtinguts van ser col·locats a /var/www/html, també vam afegir un fitxer .htaccess per redirigir les peticions a la pàgina d'inici. Vam reiniciar l'Apache i vam poder accedir a la pàgina de login des de la IP pública.

Per a posar en funcionament el Back-end calia modificar l'API d'entrada per substituir el localhost per la MV (problema de CORS) i executar "npm run start" en la carpeta correcta. El problema era que només funcionava mentre el terminal estava actiu. Per solucionar-ho vam utilitzar pm2, un sistema de d'execució de tasques. Un cop executant-se amb pm2 ja es podia tancar el terminal sense problema.

També va caldre modificar, al Front-end, la URL base de les APIs per substituir el localhost per la nova adreça.

Per posar en funcionament el scrapping calia afegir un altre procés a pm2 però el scrapper tornava a no funcionar. En el cas d'Amazon, les respostes no eren consistents, a vegades donava error i a vegades no. En el cas de PC Components, considerava que les crides eren fetes per un bot. MediaMarkt funcionava correctament.

En el cas d'Amazon vam canviar l'script original pel que utilitzava la llibreria Selenium (la mateixa llibreria que es feia servir per els altres dos scripts). En el cas de PC Components, després d'un munt de proves, vam arribar a la conclusió que estaven detectant que la IP procedia de Google Cloud i denegaven l'accés i vam decidir deixar de fer servir PC Components i utilitzar la botiga NewEgg.

També vam canviar l'execució de l'script d'scrapper, en lloc d'executar-ho amb pm2, ho vam afegir al cron de la màquina.

8.7 RESULTATS

Després de presentar tot el desenvolupament del projecte, farem referència als resultats del projecte.

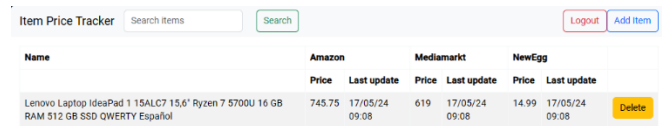
S'ha completat el desenvolupament de l'aplicació web i s'ha obtingut un servei de seguiment de preus totalment funcional que permet:

- Crear usuaris protegits amb contrasenya
- Introduir productes en seguiment per usuari
- Fer el seguiment dels preus diaris dels productes a les tres botigues configurades
- Mostrar l'últim preu dels productes en seguiment
- Mostrar el gràfic de seguiment de productes i preus

El sistema és modular i permet fàcilment substituir o afegir botigues al sistema.

Respecte a la planificació inicial, ens ha faltat desenvolupar el mòdul d'avisos. Hem vist que era més complex del que pensàvem inicialment i requeria més temps que el que teníem planificat.

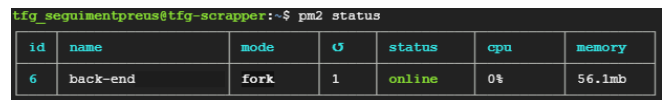
Exemple de la pàgina de la llista de productes en seguiment amb l'últim preu mesurat:



Name	Amazon		Mediamarkt		NewEgg		
	Price	Last update	Price	Last update	Price	Last update	
Lenovo Laptop IdeaPad 1 15ALC7 15.6" Ryzen 7 5700U 16 GB	745.75	17/05/24 09:08	619	17/05/24 09:08	14.99	17/05/24 09:08	Delete
RAM 512 GB SSD QWERTY Español							

Il·lustració 2: llista de productes en seguiment d'un usuari

Taula amb el procés d'execució de Back-end amb pm2:



id	name	mode	♿	status	cpu	memory
6	back-end	fork	1	online	0%	56.1mb

Il·lustració 3: llista d'execució de pm2, mostrant l'execució del Back-end

Crontab per l'execució de l'scrapper:

```
13 5 * * * cd /home/tfg_seguimentpreus/tfg-back-end/api/src/scrapping && /usr/bin/npm run start >> /home/tfg_seguimentpreus/tfg-back-end/api/src/scrapping/logs.txt
```

8.8 TESTING

Tots els test els hem fet al servidor, concretament per als objectes, per a les API i per a les sol·licituds a la base de dades. Per a crear-los hem utilitzat l'assistència de Copilot que, gràcies a les seves capacitats per a llegir codi, permetia crear de forma senzilla jocs de proves de cada part del programa.

Les proves als objectes eren proves bàsiques per testear que estiguin ben estructurats.

Els jocs de proves de les API comprovaven cada possible valor de retorn de les crides.

Els tests per a la base de dades utilitzava una BD creada en memòria per a poder fer els testos de cada taula de forma viable i ràpida.

Aquesta part ha estat molt ràpida amb l'ajuda de Copilot, això demostra que Copilot pot ajudar en gran mesura a millorar la qualitat del codi.

L'ús de Copilot per a l'automatització de jocs de prova ha demostrat ser una eina valuosa en el desenvolupament de software. La capacitat de generar tests a partir de l'anàlisi del codi facilita la creació de proves robustes i eficients, optimitzant el temps i millorant la qualitat del codi. A més, l'ús d'una base de dades en memòria per als tests permet una execució més ràpida i menys costosa de recursos. Aquesta metodologia de testos ha permès validar les funcionalitats de les APIs, la integritat dels objectes bàsics i la correcta interacció amb la base de dades, assegurant així

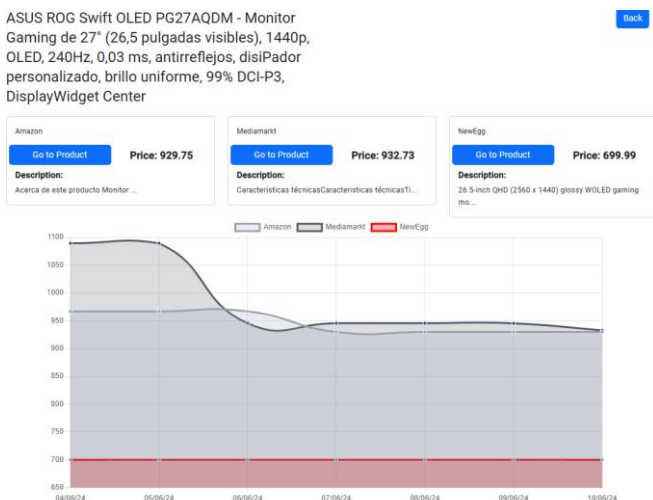
un desenvolupament més fiable i consistent.

9 CONCLUSIONS

De tot el projecte, l'script d'scrapping és la part més delicada ja que no només depèn de l'script sinó de les mesures anti-scrapper que estiguin fent servir cadascuna de les botigues, que a més pot canviar al llarg del temps i implica anar adaptant l'script a les mesures anti-scrapper que es vagin aplicant.

Tenint en compte tot el desenvolupament i que l'scrapper segueixi funcionant, creiem que el projecte es podrà acabar per la data de l'entrega.

Fa poc que s'estan recollint preus però ja hi ha hagut resultats amb canvis de preu. Com es pot veure en la següent imatge, ja podem veure variacions de preus. Es pot veure que per al ASUS ROG Swift s'ha modificat el preu de forma significativa a Amazon i de manera moderada en Mediamarkt, mentre que NewEgg no ha modificat el seu preu en absolut.



Il·lustració 4: Producte en seguiment mostrant una gràfica amb tots els preus des de l'inici del seguiment

10 LÍNIES OBERTES

En aquesta darrera secció, plantejarem diverses propostes de millora per al projecte. Son idees que poden millorar l'aplicació però que no entren dins de l'abast d'aquest projecte.

10.1 CERTIFICAT HTTPS

Caldria afegir un certificat HTTPS vàlid per millorar la seguretat i la confidencialitat de les comunicacions entre els usuaris i la plataforma.

10.2 AUTOMATITZACIÓ DE LA CERCA DE PRODUCTES

Una millora futura podria ser la creació d'un sistema d'automatitzat que cerqui productes en les botigues en línia mitjançant el seu nom. Això estalviaria temps als usuaris i milloraria l'experiència de cerca. Ja que ara, per fer seguiment d'un producte, cal introduir la URL del producte de cada una de les botigues.

10.3 PERSONALITZACIÓ DEL GRAF

Els gràfics de l'aplicació son molt bàsics, es podrien millorar afegint més opcions de personalització per als usuaris, com canviar el disseny, permetre seleccionar el període de temps, fer gràfics comparant els diferents preus per percentatge en lloc de preu absolut...

10.4 ELIMINACIÓ D'USUARIS

L'aplicació només permet afegir usuaris, caldria una gestió completa que permeti també modificacions i baixes, canvis de contrasenya i possibilitat de definir un doble factor d'autenticació.

10.5 SISTEMA D'AVISOS

Tot i que estava previst en el projecte, no hem arribat a implementar una gestió d'avisos.

Seria interessant afegir la possibilitat de generar avisos als usuaris quan hi hagi un canvi important als preus o permetre a l'usuari configurar llindars per rebre avisos quan el preu baixi o superi el llindar. Els avisos es podrien rebre per correu electrònic, Telegram, Whatsapp...

També es podria aprofitar per enviar notificacions sobre canvis en la versió o altres esdeveniments rellevants.

10.6 FER UN RANG MAJOR DE BOTIGUES

Actualment només hi ha 3 botigues i no es dona l'opció de triar si es vol fer el seguiment a unes i no a les altres. Una millora per al projecte seria augmentar el nombre de possibles botigues i permetre que l'usuari seleccioni en quines es vol fer el seguiment.

AGRAÏMENTS

Vull agrair a la Yolanda Benitez per a l'assistència i la guia proporcionada al llarg del projecte i al meu pare l'Aureli Roura per les recomanacions que m'ha anat fent al llarg del projecte.

BIBLIOGRAFIA

(2023, November 17). *What's the Real History of Black Friday?* History. Retrieved March 1, 2024, from <https://www.history.com/news/black-friday-thanksgiving-origins-history>

(n.d.). *Camelcamelcamel*. Camelcamelcamel. Retrieved March 4, 2024, from <https://es.camelcamelcamel.com/>

(n.d.). *Histórico de precios y seguimiento de precios para amazon*. PriceTracker. Retrieved March 4, 2024, from <https://price-tracker.es/>

(n.d.). *Free Amazon Price Tracker and European Comparison Tool*. Bolabob. Retrieved March 4, 2024, from <https://es.bobalob.com/>

(n.d.). *The Best Deals, Coupons, Promo Codes & Discounts*. Slickdeals. Retrieved March 4, 2024, from <https://slickdeals.net/>

(n.d.). *Get Coupons & Deals at Your Favorite Stores*. CapitalOne Shopping. Retrieved March 4, 2024, from <https://capitaloneshopping.com/s/all>

(2023, July 24). *Top +20: Las mejores herramientas de monitorización de precios (2023)*. Marketing 4 Ecommerce. Retrieved March 4, 2024, from <https://marketing4ecommerce.net/las-mejores-herramientas-de-monitorizacion-de-precios/>

(2023, October 24). *These Are the Best Tools for Tracking Online Sale Prices*. Lifehacker. Retrieved March 4, 2024, from <https://lifehacker.com/>

(n.d.). *Finding web elements*. Selenium. Retrieved April 12, 2024, from <https://www.selenium.dev/documentation/webdriver/elements/fin- ders/>

(n.d.). *Selenium*. Selenium. Retrieved April 12, 2024, from <https://www.selenium.dev/>

(n.d.). *Welcome to Cheerio!* Cheerio. Retrieved April 19, 2024, from <https://cheerio.js.org/docs/intro>

(n.d.). *Getting Started*. Axios. Retrieved May 1, 2024, from <https://axios-http.com/docs/intro>

Google (n.d.). *Documentación de Google Cloud*. GoogleCloud. Retrieved May 16, 2024, from <https://cloud.google.com/docs?hl=es-419>

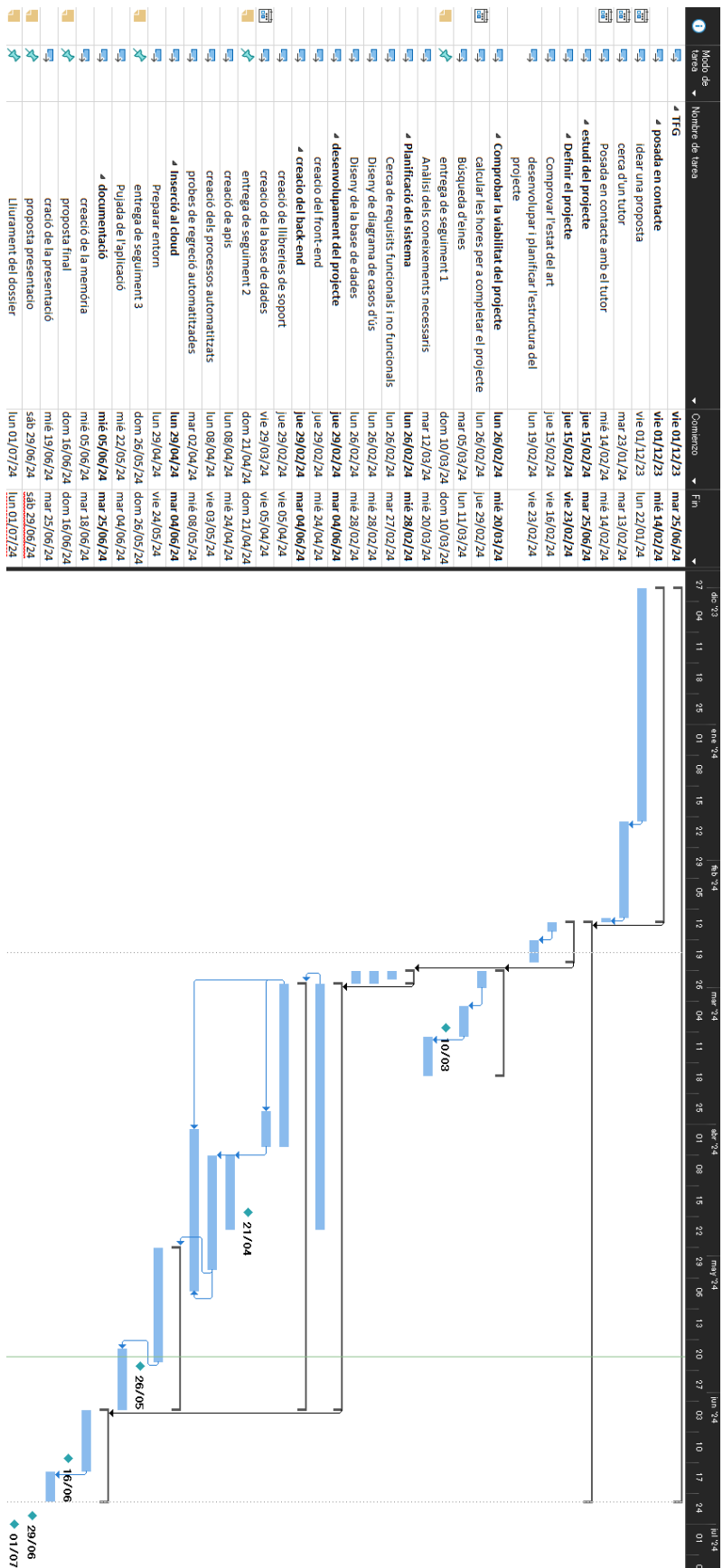
(n.d.). *PM2 Plus Documentation*. Pm2. Retrieved May 20, 2024, from <https://pm2.io/docs/plus/overview/>

APÈNDIX

A1. TAULA D’ANÀLISI D’APLICACIONS SIMILARS

Nom de la web	característica							
	seguiment de preus	graf temporal	seguiment d'un producte	compara-dor de preus	usuaris	cerca per URL	cerca per nom	usuaris de la UE permesos
camelcamelcamel	si	si	si	no	si	si	si	si
Keepa	si	si	si	no	si	si	si	si
Honey	si	si	si	no	si	si	si	si
Slickdeals	si	no	si	no	si	si	si	no
CapitalOne Shopping	si	no	no	si	error en la creació	no	imprecisa	si
Bobalob	si	si	si	no	si	si	si	si
price-Tracker	si	no	si	no	si	si	si	si
Aliprice	si	si	si	no	si	si	si	si
Twenga	si	no	no	no	no	no	imprecisa	si

A2. DIAGRAMA DE GANTT AMB LA PLANIFICACIÓ



A3. DIAGRAMA DE CASOS D'ÚS

