

E-Viewer: Seguiment en temps real per a Esports Electrònics

Adrià Martínez Vega

Resum—En aquest projecte, es busca crear una aplicació on els usuaris puguin consultar dades en directe de partits d'esports electrònics, evitant la necessitat d'entrar a la retransmissió per saber com li va al seu equip preferit. A més a més, aquesta aplicació també oferirà informació sobre equips, jugadors i competicions per a que l'usuari tingui una referència completa per consultar-les. Per aconseguir-ho, es desenvoluparà un sistema d'extracció de dades de les partides, recolzat per models d'aprenentatge automàtic, i una base de dades que contindrà tota la informació necessària per a l'aplicació. Es realitzarà un testing exhaustiu per identificar els punts dèbils de l'aplicació i solucionar-los. Finalment, es durà a terme un estudi de costos per determinar les despeses associades a la publicació de l'aplicació i es proposaran models de monetització per cobrir aquests costos i obtenir rendibilitat.

Paraules clau—Aplicació, Esports Electrònics, Equip, Competicions, Sèrie, Jugadors, Partit, League of Legends, Aprenentatge Automàtic, Logistic Regression, Base de dades, Firestore, Testing, Flutter, Interfaç d'usuari, Costos, Monetització.

Abstract—In this project, the aim is to create an application where users can access real-time data from esports matches without needing to enter the live stream to check on their favorite team's performance. Additionally, the application will provide information on teams, players, and competitions for users to reference. To achieve this, it will be necessary to develop a data extraction system for matches supported by machine learning models, and a database containing the data to be queried within the application. A thorough testing phase will be conducted to identify and address any weaknesses in the application. Finally, a cost study will be conducted to assess the expenses associated with potential publication, alongside exploring monetization strategies to cover these costs and achieve profitability.

Index Terms—Application, Esports Electronics, Team, Competitions, Series, Players, Match, League of Legends, Machine Learning, Logistic Regression, Database, Firestore, Testing, Flutter, User Interface, Costs, Monetization.



1 INTRODUCCIÓ - CONTEXT DEL TREBALL

AQUEST projecte consisteix en el desenvolupament d'una aplicació per a dispositius mòbils en la qual l'usuari pugui seguir en temps real la situació dels partits d'esports electrònics que estigui interessat.

Els esports electrònics són competicions professionals de videojocs. A dia d'avui, encara que les xifres d'espectadors estan lluny dels grans esdeveniments, hi ha milions d'espectadors que tenen interès en els partits i segueixen les competicions.

Al contrari d'altres esports tradicionals on amb una cerca ràpida a Google ja tens el resultat en directe, l'única manera que un aficionat pugui saber com va el seu equip sense la necessitat d'entrar a la retransmissió és a partir de les xarxes socials dels equips. La gran majoria de vegades això no és suficient per saber quina és la situació real de la partida, ja que molts equips simplement pugen petits fragments de la partida i el resultat final, cosa que fa que l'usuari mai tingui tota l'informació completa.

Per aquesta raó, amb aquest projecte s'intenta emular aplicacions com Besoccer [1], Sofascore [2] o Onefootball [3], que, centrades completament en el futbol i gràcies a un sistema d'extracció de dades en directe i una base de dades ben preparada, busquen que els usuaris tinguin un

lloc on, a part de simplement tenir el resultat, puguin veure dades més detallades dels partits en directe d'una manera senzilla, immediata i gratuïta. A causa de la seva naturalesa, aquest tipus d'aplicacions han aconseguit moltíssima popularitat en els últims anys, cosa que fa pensar que una aplicació que busqui donar aquesta informació als seguidors dels esports electrònics fagi que millori la seva experiència.

Durant els mateixos partits, l'usuari podrà consultar totes les estadístiques rellevants de la situació de partida des de l'inici fins al moment actual, per a que es pugui veure com ha anat evolucionant la partida, fins i tot oferint gràfiques de les mètriques més importants (segons l'esport electrònic al qual pertanyi la partida consultada) per tenir una representació més visual del progrés de la partida.

A més, més enllà dels partits, l'aplicació oferirà consultar altres dades o estadístiques de jugadors, equips o competicions en concret, com per exemple el palmarès d'un equip o la classificació actual d'una competició, servint com una espècie d'enciclopèdia perquè els usuaris tinguin un lloc on consultar ràpidament el que estiguin interessats. Actualment, existeixen pàgines web que ja compleixen aquesta funció i que són la referència per

saber classificacions, estadístiques... però moltes vegades les dades no s'actualitzen fins a l'endemà. En oferir les dades en directe i amb actualitzacions automàtiques, sumat a tenir-ho tot a una aplicació més fàcil d'utilitzar en dispositius mòbils que una pàgina web, farà que l'aplicació tingui un avantatge sobre aquest tipus de pàgines.

Per qüestions de temps material, s'ha decidit centrar l'aplicació exclusivament en l'escena competitiva del videojoc League of Legends [4], utilitzant com a base de dades de prova la Lliga Europea de 2023 [5]. Aquesta decisió es va prendre perquè incloure més videojocs hauria requerit un sistema d'extracció de dades i models d'aprenentatge automàtic diferents per a cada joc, fet que hauria allargat massa el temps de desenvolupament i no hauria permès obtenir un resultat satisfactori dins dels terminis previstos. L'escena competitiva de League of Legends és considerada la major dintre dels esports electrònics, oferint una gran quantitat de dades per analitzar i utilitzar en el desenvolupament de l'aplicació. La Lliga Europea de 2023, en particular, proporciona un conjunt de dades robust i divers que és ideal per provar i perfeccionar els models d'aprenentatge automàtic implementats. Aquesta focalització permet assegurar que l'aplicació pugui gestionar eficaçment les dades i produir resultats significatius i útils.



Figura 1: Captura d'una retransmissió de la Lliga Europea de League of Legends durant una partida

1.1 Objectius

Segons la prioritat dels objectius, s'ha separat en dues categories: Principals i Secundaris.

Els objectius Principals per al desenvolupament del treball són els següents:

- Muntar un sistema d'extracció de dades que sigui capaç de connectar-se a una retransmissió en directe d'una partida de League of Legends, reconèixer la partida que s'està jugant i extreure les dades a partir de captures i crear una connexió a la base de dades per a guardar les dades en directe. Amb aquest sistema, apareixen 2 subobjectius:

- + Dissenyar un programa que, a partir de l'ús d'OCR (Reconeixement Òptic de Caràcters), sigui capaç de reconèixer i interpretar totes les dades claus d'una partida. Les dades poden estar

conformades tant per caràcters alfabètics (noms dels equips que estan jugant la partida o els noms d'usuari dels jugadors) com per caràcters numèrics (marcador o temps de partida).

- + Crear i entrenar diversos models d'aprenentatge automàtic perquè es puguin reconèixer les dades que estan en imatges/símbols [6][7], com per exemple amb quin personatge està jugant cada jugador, i que el sistema d'extracció de dades sigui capaç de saber si en el moment de l'extracció de les dades s'està dintre o fora de la partida.

- A partir de decisions de disseny tenint en compte requeriments com l'escalabilitat, aplicar els coneixements aconseguits durant la carrera per la creació de la base de dades i omplir-la amb les dades suficients per poder demostrar totes les capacitats de l'aplicació.
- Crear una aplicació que permeti consultar les dades extretes en directe i un recull de dades històriques des de 0 seguint les decisions de disseny preses per trobar un llenguatge i SDK òptim tant per la connexió amb la base de dades com per a l'experiència de l'usuari.

Els objectius Secundaris per al desenvolupament del treball són els següents:

- Fer que el sistema d'extracció de dades sigui robust a errors que puguin existir en la retransmissió, com, per exemple, que s'endarrerixi l'inici de la partida o que hi hagi una pausa; i a errors del mateix sistema d'extracció de dades, com quan detecta algun caràcter erròniament.
- Optimitzar el rendiment de l'aplicació, aconseguint que el temps de càrrega de les pantalles de l'aplicació siguin raonablement baixos i que la diferència de temps entre el directe i l'aplicació sigui prou acceptable per evitar estar massa desfasat.
- Tenir una interfàç d'usuari a l'aplicació que contingui les dades relacionades amb la pantalla del jugador, equip o competició d'una manera que sigui fàcil i còmode per l'usuari trobar l'estadística que estigui interessat. A més, afegir gràfiques interactives per millorar l'experiència de l'usuari quan vulgui veure l'evolució de la partida i la situació d'aquesta en un punt concret.
- Buscar possibles sistemes de monetització per a l'aplicació, a partir d'exemples d'altres aplicacions similars.

2 ANÀLISI DE REQUERIMENTS I DISSENY DE L'APLICACIÓ

En aquest apartat, s'explicarà el perquè de les decisions de disseny sobre tot el que rodeja l'aplicació i el funcionament de tot l'esmentat, des del sistema d'extracció de dades fins a les pròpies pantalles de l'aplicació.

2.1 Sistema d'extracció de dades

El sistema d'extracció de dades és l'encarregat d'aconseguir les dades a partir d'un codi es connecta a la plataforma de directes Twitch [8] (que és la plataforma utilitzada per a la gran majoria d'esports electrònics) i

retorna una captura de pantalla de mida 1920x1080, que li direm “situació de partida”.

Al rebre un captura del codi anterior, el programa de detecció de textos retalla els textos d’interès amb unes coordenades preestablertes i extreu els caràcters per retornar la paraula.

Depenent del tipus de text, es fan diverses transformacions al la zona retallada per facilitar la detecció del text, com passar l’imatge a blanc i negre o utilitzar un threshold de color, canviar el tamany de la imatge, o moure la zona de retallada a una altra posició segons el que passa a la pròpia partida.

Per les dades numèriques, si hi ha un error de detecció i detecta un nombre menor al de la situació de partida anterior (cosa que és impossible), el sistema no guarda les dades d’aquesta situació. Si es dona el cas que detecta un nombre major, es comprova si la diferència entre les mètriques és massa dràstica, de manera que si hi ha un error on la mètrica errònia és major, no es queda en una situació incorrecta.

2.2 Models de Machine Learning

Per reconèixer les dades en forma d’imatges, el sistema d’extracció de dades es reforça amb l’ús de diversos models d’aprenentatge automàtic:



Figura 2: A partir de la Figura 1, exemple de les dades que s'han d'extreure en una partida. En vermell, les dades extretes amb OCR. En blau, les dades que necessiten un model d'aprenentatge automàtic.

- **Model de detecció de personatges:** Igual que amb els textos, utilitzant la llibreria cv2 es retalla la part integral per a la detecció. Amb la retallada, es passa per una Logistic Regression [9] entrenada amb l'estratègia One-Shot (un sola mostra d'entrenament per classe) i aquesta retorna correctament quin personatge és.

- **Model de detecció de Dracs:** En "League of Legends" existeixen 4 tipus de "Dracs" que els equips poden aconseguir per millorar les seves estadístiques. Com és una estadística rellevant, s'ha de guardar per a que l'usuari tingui aquesta informació. Per aconseguir-ho, es retalla la part del Overlay on estan indicats, utilitzant dos models, ja que un model sol no detecta suficientment bé el tipus de drac. El primer, una Logistic Regression One-Shot, detecta si existeix un símbol o no, i, si existeix, el segon, entrenat amb més mostres, detecta el tipus de drac a partir de la forma i color del símbol.

- **Model de detecció de Partida:** Durant les primeres proves amb partides senceres, va sorgir el problema que no es podia detectar l'inici o final de la partida, cosa que era essencial si es volia fer la prova per tota la jornada sencera. Per això, és necessari un model SVM (Support Vector Machines) [10] entrenat amb múltiples captures d'una retransmissió per a que pugui saber quan està en partida i quan no. Quan no està en partida, el programa busca un text que es mostra a les retransmissions i que indica el guanyador. Això també ens permet diferenciar si la partida ha acabat o està en pausa.

2.3 Base de Dades

L'aplicació necessitava una base de dades que fos capaç de gestionar una gran quantitat d'informació de manera eficient, permetent una ràpida recuperació de dades i suportant múltiples consultes simultànies. A més, buscàvem una solució que fos escalable, que oferís una bona integració amb el nostre entorn de desenvolupament i que pogués funcionar correctament en un entorn de núvol per facilitar l'accés des de qualsevol lloc. També era important que la base de dades escollida fos capaç de sincronitzar els canvis en temps real, per tal de millorar l'experiència d'usuari en l'aplicació.

Inicialment, es va considerar l'ús d'una base de dades relacional, però optar per una no relacional, a més de que complia tots els requisits amb facilitat, va oferir una major flexibilitat en l'emmagatzematge de les dades, adaptant-se millor al sistema d'extracció al no tenir que adaptar-se a una estructura concreta. Això ha simplificat i concretat les consultes, millorant notablement l'eficiència i el rendiment del sistema.

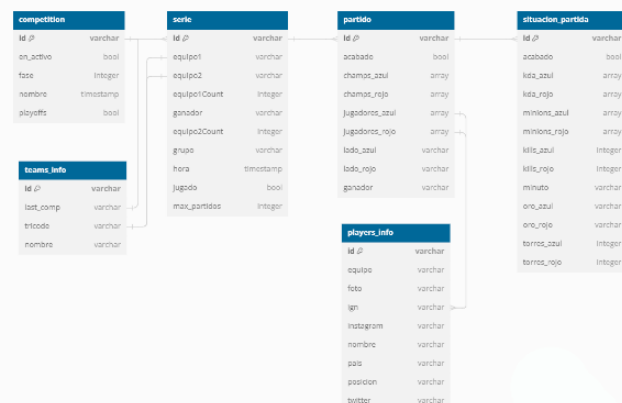


Figura 3: Organització de la base de dades No Relacional.

Amb totes les dades necessàries, l'estructura dintre de la base de dades és aquesta: Competicions: Es guarden les diverses competicions jugades, guardant el nom, si està en actiu, la fase on està (lliga/playoffs) i:

- **Sèries:** Sèrie jugada en una competició. Es guarda els equips, la quantitat de partides guanyades dintre de la sèrie, partits necessaris per acabar i la hora que s'ha jugat. A més conté:

- Partits: Partit dintre d'una sèrie. Es guarden les dades que duren tot el partit, com els equips, jugadors i personatges i si s'ha jugat. Dintre té:

+ Situació de partida: Aquí es guarden les dades que van canviant durant el partit i el minut on passen.

- Equips: Equips que participen en una competició amb la seva posició. A més, es guarda la plantilla de l'equip amb la posició de cada jugador.

- Informació d'equips: Guarda dades com l'última competició que estan jugant o el nom sencer de l'equip.

- Informació de jugadors: Es guarda l'equip on està jugant, la seva posició actual, i altres dades com el seu nom o les seves xarxes socials.

2.4 Connexió entre BD i el Sistema d'extracció

Per guardar les dades extretes a la base de dades, necessitem un codi que estigui executant-se contínuament que connecti la base de dades amb el sistema d'extracció d'aquestes. A continuació es descriuen els passos necessaris per aconseguir aquest objectiu:

1. Inici de la sèrie: Quan arriba l'hora d'inici de la jornada, es registra l'ID de la sèrie a la base de dades i es marca com a partit en directe. Això assegura que el sistema estigui preparat per al seguiment en temps real.

2. Inici del seguiment del partit: Mentre cap equip hagi arribat al nombre màxim de partits guanyats, es comença el seguiment del partit. Això implica estar alerta per a la recepció d'imatges que indiquin que el partit ha començat. Si es dona el cas que hi ha un equip guanyador, es torna al pas 1 amb la següent sèrie de la jornada. Si no hi ha més sèries, el codi esperarà al dia següent per tornar al pas 1.

3. Recepció de la primera imatge de partida: Quan es rep una primera imatge de la partida, es guarden les dades persistents, com els jugadors titulars, els personatges seleccionats i el costat dels equips. En aquest punt, es considera que el partit ha començat oficialment, marcant-lo com a en directe a la base de dades i inicialitzant una situació de partida 0.

4. Seguiment continu de la partida: Mentre la partida no hagi finalitzat, el sistema entra en un bucle en el qual es crida la funció que captura pantalles, extreu les dades i les guarda a la base de dades. Com s'ha comentat abans, aquesta funció ha de tenir en compte que hi ha certs valors (com el minut de partida) que, si per error en el reconeixement del text tenen un valor menor que l'anterior, no s'actualitzen, ja que no tindria sentit. Realitzar actualitzacions cada pocs segons assegura que aquesta solució no provoca problemes en el seguiment de les partides.

5. Finalització de la partida: Quan la partida acaba, es suma 1 al comptador de l'equip guanyador i es torna al condicional del pas 2, preparant el sistema per al següent partit.

Aquest procés automatitzat és clau per garantir que el seguiment dels partits es realitzi de manera precisa i eficient, minimitzant els errors i assegurant que les dades

es registrin correctament en temps real.

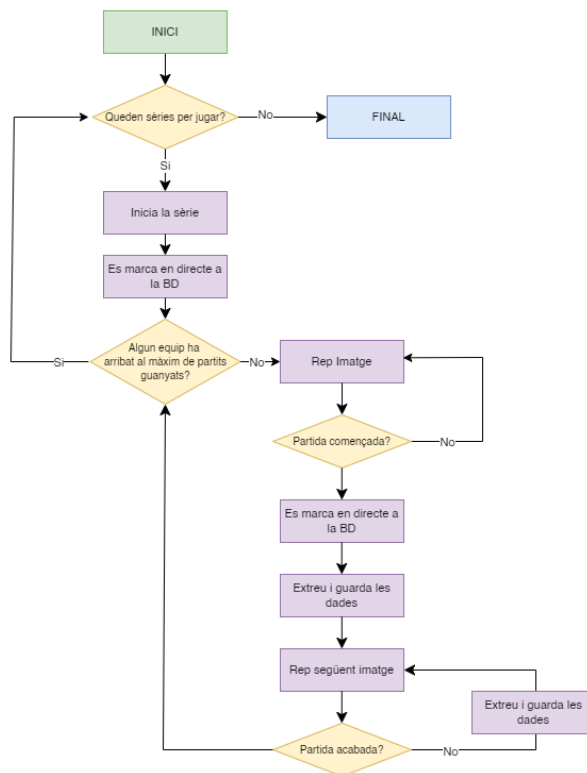


Figura 4: Diagrama de flux del codi que connecta la Base de dades i el sistema d'extracció

2.5 Funcionalitats de l'aplicació

A continuació, es detallen els desenvolupaments específics per cada pantalla:

- Pantalla Inicial: Aquesta pantalla és la primera que veurà l'usuari en obrir l'aplicació. Es divideix en tres pestanyes que mostren les sèries d'ahir, avui i demà. L'aplicació ha de tenir en compte si es juguen sèries per a cada dia i si la sèrie ja s'ha jugat, encara s'està jugant, o encara no s'ha jugat. Si la sèrie ja s'ha jugat o encara està en directe, el botó d'informació de la sèrie mostrarà el resultat (i un indicador si està en directe); si encara no s'ha jugat, mostrarà l'hora prevista del partit. A més, aquests botons també permeten accedir a la pantalla de la sèrie.

- Sèrie: En aquesta pantalla, inicialment es mostra el resultat de la sèrie en curs (o l'hora prevista del partit), els equips participants (interactius per anar a la pantalla de l'Equip) i la competició (també interactuable). Té 3 pestanyes: Partits, Prèvia i Classificació. La primera, similar a la pantalla inicial, mostra els partits de la sèrie que ja s'han jugat o s'estan jugant, amb botons interactius que mostren el resultat i porten a la pantalla de Partit. Si la sèrie ja s'ha jugat o està en curs, aquesta serà la pestanya que apareixerà quan s'inicia l'aplicació; si no, la primera serà la de Prèvia. En aquesta pestanya de Prèvia, es mostra una taula amb la ratxa de cada equip, perquè l'usuari pugui veure l'estat de forma de cadascun d'ells.

També es mostren els últims enfrontaments entre els equips, amb enllaços interactius a la pantalla d'aquesta sèrie específica, i una taula amb les plantilles, amb enllaços a la pantalla del Jugador. Finalment, a la pestanya de Classificació s'ensenya l'estat de la classificació de la competició en el moment de la sèrie, proporcionant a l'usuari informació sobre la situació actual de la competició.

- **Equip:** A la part superior, apareix el nom de l'equip i la regió (interactuable per anar a la competició). La primera pestanya mostra informació sobre l'equip, com ara les xarxes socials, i la darrera i pròxima sèrie que jugarà (interactuable). Les següents pestanyes mostren la informació de les últimes sèries jugades i la plantilla actual de l'equip, amb enllaços interactius a les pantalles corresponents. La pestanya del Palmarès està conformada per diversos widgets desplegable amb els resultats de les competicions en què ha participat l'equip. En interactuar-hi, es desplega i mostra la plantilla amb la qual s'ha jugat (cada jugador és interactuable). Finalment, també es mostra la classificació actual de la competició en què participa l'equip, amb el mateix format que a la pantalla de Sèrie.

- **Jugador:** En aquesta pantalla, primer es mostra el nom del jugador i l'equip pel qual juga (interactuable). Així com amb l'Equip, la primera pestanya conté informació general sobre el jugador i la següent sèrie en què jugarà. La segona pestanya mostra els partits que ha jugat el jugador, ja que, en aquest cas, la informació és específica de cada partit i no de la sèrie en general. La darrera pestanya mostra la plantilla de l'equip actual del jugador, tot sent interactuable. De manera similar a la pestanya del Palmarès d'Equip, a la pestanya de Trajectòria s'ensenyen desplegables amb els equips amb els quals ha jugat el jugador. En interactuar amb ells, es mostren les competicions en què ha participat amb aquest equip i els resultats, amb enllaços a la pantalla d'aquella competició.

- **Competició:** En tenir una pantalla separada per a cada "split" (la subdivisió de la competició, dintre de cada temporada hi ha 3 splits), l'usuari té la possibilitat de canviar l'any i el "split" dins de la mateixa pantalla per facilitar la cerca d'informació. Inicialment, es mostra la pestanya de Sèries, que bàsicament és una llista de totes les sèries que s'han jugat i es jugaran, començant per les sèries més properes a la data actual. Si es llisca cap amunt, apareixeran les que ja s'han jugat, i si es toca el botó d'una sèrie, anirà a la pantalla corresponent. A les següents pestanyes també es conserva la classificació i els equips que participen a la competició (com és lògic, en interactuar amb ells portarà a la pantalla de l'Equip).

Per últim, també hi ha una pantalla amb estadístiques de la competició, que inclou el top 3 jugadors per les estadístiques individuals més valorades.

- **Partida:** Aquesta és la pantalla més important de l'aplicació, ja que mostra la situació del partit amb totes les dades importants perquè un usuari pugui saber com va el partit. A l'informació inicial es mostren els equips

que estan jugant, el minut de partida (o el resultat si ja ha acabat), la diferència de or (la mètrica més important per saber com va la partida) i les "kills" (la puntuació individual més important) de cada un dels equips. A sota, es guarda una taula amb tota altra informació que s'extreu i que està relacionada amb el jugador, com altres mètriques individuals, el personatge que està jugant o la diferència de puntuació amb la contrapart de l'altre equip.

En interactuar amb una cel·la de la taula, apareixerà una nova pestanya amb els mateixos detalls i les últimes partides jugades amb el mateix personatge a la competició i pel jugador. Hi ha dos tipus de partits:

- **Partit Recent:** En aquests partits es guarden totes les situacions de la partida, oferint una gràfica amb la qual l'usuari pot interactuar per saber quina és la diferència de or en aquell

moment de la partida. A més, la taula també s'actualitzarà perquè l'usuari tingui la informació de cada jugador en aquella situació de la partida.

- **Partit Antic:** Aquest tipus de partits no tenen la informació per cada situació de partida, ja que es considera que al haver passat temps des del partit, la informació de cada situació no és rellevant per a l'usuari. Es guarda només la situació final per tenir espai a la base de dades.

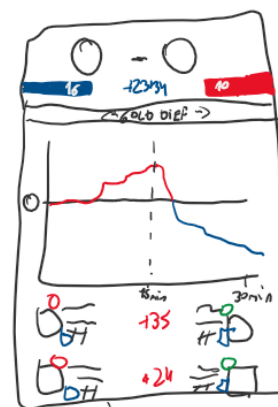


Figura 5: Mock-Up inicial de la pantalla de partida en directe

3 IMPLEMENTACIÓ DE L'APLICACIÓ

Durant la fase d'implementació, s'ha dut a terme la traducció dels dissenys en codi funcional. Això ha implicat la creació de diverses capes de l'aplicació, incloent la interfície d'usuari i el Software Development Kit utilitzat. Cada component ha estat meticulosament integrat per garantir un funcionament coherent i eficaç del sistema.

3.1 Sistema d'extracció

Per a la implementació d'aquest sistema, es farà servir la llibreria Selenium [11] per a automatitzar la connexió i navegació per la plataforma Twitch, així com per a capturar les pantalles necessàries. Un cop obtinguda la captura de pantalla, la llibreria OpenCV (cv2) [12] s'utilitzarà per a retallar les àrees específiques on es troben els textos d'interès mitjançant les coordenades predefinides. Finalment, s'emprarà la llibreria pytesseract [13] per a extreure els caràcters i convertir-los en text que el sistema pugui processar.

3.2 SDK

Es va optar per Flutter [14], el llenguatge per a dispositius mòbils propi de Google per aquestes raons:

- **Llenguatge propi:** Amb la necessitat d'aprendre des de zero qualsevol de les opcions disponibles, Dart [15] (el llenguatge de programació de Flutter) s'ha revelat com una opció atractiva per la seva simplicitat i corba d'aprenentatge amigable.
- **Únic codi per totes les plataformes:** La capacitat de Flutter d'utilitzar el mateix codi per crear aplicacions en Android, iOS i Web ofereix una eficiència notable i assegura una coherència en el desenvolupament multiplataforma.
- **Eines d'ajuda al desenvolupament:** Les facilitats proporcionades per Flutter, com el "Hot Reload" (recarregar l'aplicació sense reiniciar-la per veure els canvis) i la integració de widgets predeterminats, simplifiquen el procés de desenvolupament i acceleren la iteració.
- **Part del Framework de Google:** La integració de Flutter amb el framework de Google ofereix avantatges significatius, especialment en la connexió amb Firestore, la base de dades seleccionada. Aquesta integració estreta millora la eficàcia del desenvolupament i la interacció amb les dades.
- **Integració a Android Studio i gran comunitat:** Encara que aquestes no són característiques exclusives de Flutter, és important destacar que la seva integració amb Android Studio i el suport d'una gran comunitat de desenvolupadors són avantatges significatius. La integració facilita l'ús d'eines de desenvolupament ja conegudes, mentre que una gran comunitat significa més recursos, tutorials i suport per a la resolució de problemes.

3.3 Base de dades

Després d'avaluar diverses opcions, es va decidir utilitzar Firestore [16], una base de dades no relacional, ja que complia tots els requisits necessaris:

- **Espai:** La versió gratuïta ofereix el suficient espai per guardar les dades necessàries per una versió funcional de l'aplicació.
- **Consultes:** Ofereix la possibilitat de fer consultes en diverses plataformes d'una manera senzilla, fins i tot creant índexs automàticament quan ho requereix. A més, amb la versió gratuïta es permeten suficients consultes de lectura i escriptura.
- **Cloud:** Al ja estar implementada directament en el núvol de Google, no és necessària una implementació en el cloud i totes les betes es podran provar des de qualsevol lloc.
- **Escalabilitat:** Igual que Azure o AWS, ens ofereix uns preus competitius si és necessari augmentar les capacitats de la base de dades en número de consultes i en espai.
- **Framework:** Al ser propietat de Google, té una implementació molt més ràpida dintre de Flutter, el SDK de codi obert (també propietat de Google) per crear aplicacions de mòbil i que era l'opció principal per crear l'aplicació beta.
- **Actualització:** Firestore és capaç d'enviar els canvis a la

base de dades en directe a l'aplicació per a que la pantalla canviï sense la necessitat de refrescar manualment.

A més, gràcies a l'ús de Firestore com a base de dades, amb una simple configuració dintre del codi de l'aplicació les dades de la partida en directe s'actualitzen automàticament, de manera que l'usuari pot estar atent a la seva pantalla per veure com evoluciona la partida sense la necessitat de recarregar-la.

3.4 UI

L'objectiu d'aquests canvis és no només millorar l'aspecte visual de l'aplicació, sinó també fer que sigui més funcional i agradable d'utilitzar. Els canvis més importants són:

- **Paleta de colors:** S'ha passat del color blanc bàsic de la plantilla inicial de Flutter a una combinació de blanc i lila neó. A més, s'han afegit interaccions especials per remarcar certes situacions concretes, com per exemple, si la partida està en directe, el contenidor del resultat canvia completament de color, ressaltant per sobre dels altres. Aquesta nova paleta no només fa que l'aplicació sigui visualment més atractiva, sinó que també millora la usabilitat destacant les parts més rellevants de la informació.
- **Font:** Per captar millor l'interès de l'usuari, s'han utilitzat fonts més modernes que harmonitzen amb la paleta de colors esmentada anteriorment. L'ús de fonts modernes no només fa que el text sigui més fàcil de llegir, sinó que també contribueix a una aparença més professional i polida de l'aplicació.
- **Ús d'imatges:** S'han afegit imatges per proporcionar referències visuals a textos importants, com els logos i escuts dels equips (carregats localment) i altres imatges, com els personatges que s'estan utilitzant en la partida (carregats a través de la connexió a Internet). En el cas dels logos, les imatges s'han guardat en múltiples colors per adaptar-se a les interaccions especials explicades anteriorment. Això assegura que les imatges siguin sempre rellevants i coherents amb el context en què es mostren.

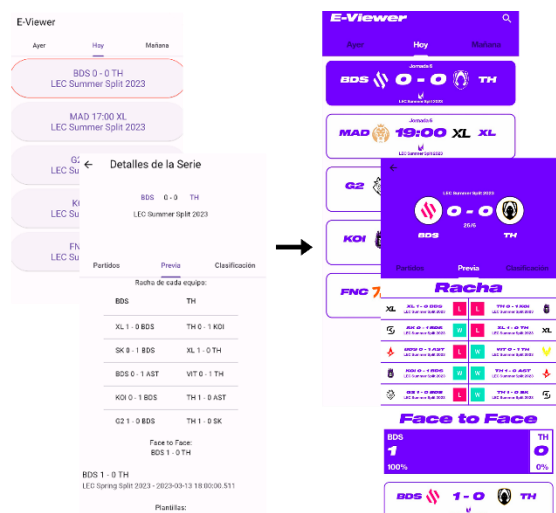


Figura 6: Canvi de la UI de l'aplicació a partir dels canvis esmentats

4 TESTING I RENDIMENT

Durant el desenvolupament de la app i al final d'aquest, s'han anat fent diversos tipus de test per extreure conclusions sobre com funciona l'aplicació i què es pot fer per millorar-la.

4.1 Funcionalitats Bàsiques

Abans de fer proves més avançades, el primer test va consistir a comprovar que totes les funcionalitats de l'aplicació no presentaven errors. Es va accedir a tots els tipus de pantalles, utilitzant cadascun dels elements interactius per assegurar que funcionaven correctament i estaven enllaçats amb la pantalla corresponent.

Després d'aquesta verificació inicial, es va passar a la pantalla de Partida en Directe per saber cada quant s'actualitzava l'aplicació i si detectava correctament l'inici i el final de la partida. Es va observar que les dades de la partida s'actualitzaven cada 30 segons, que és el temps que el codi triga a fer la captura, processar les dades i enviar-les a la base de dades (ja que l'actualització des de la base de dades a l'aplicació és instantània gràcies a Firestore). Fora de la partida, el codi fa captures cada 2 segons, ja que no ha de processar informació al detectar que està fora de partida. Aquesta rapidesa permet que l'usuari sàpiga en el moment quan comença la partida i tingui accés a la informació només 30 segons després de que ocorri a la partida, un retard perfectament acceptable per als objectius de l'aplicació.

4.2 Rendiment general de l'aplicació

En aquest testing, s'ha buscat provar totes les funcionalitats de l'aplicació per veure si durant una jornada de lliga regular de 5 hores, aquesta era capaç de suportar l'ús continuat sense grans temps de càrrega i de manera lleugera. Per fer-ho, 2 usuaris simultàniament han entrat a l'aplicació 2 vegades per partida, han anat a la pantalla de partida en directe i esperat a l'actualització de les dades. A més, també s'ha provat a entrar a una pantalla d'equip, competició i jugador cada vegada que s'obria l'aplicació.

En general, no s'ha detectat quasi cap problema de rendiment. Tot i que gairebé tota la informació s'ha d'extreure de la base de dades, les respostes a les consultes són ràpides i el temps d'espera mitjà per veure les dades d'una nova pantalla no supera els 2 segons, cosa que proporciona una experiència d'usuari fluida. A més, gràcies a l'ús de Firestore, al actualitzar les dades en directe, l'aplicació no té cap mena de càrrega ni "lag", canviant les dades directament sense que l'usuari noti res.

Pel que fa a les imatges locals, l'aplicació les carrega immediatament, mentre que les imatges provinents d'Internet poden trigar uns segons a aparèixer. No obstant això, un cop carregades per primera vegada, queden guardades a la memòria cau de l'aplicació. Així, si l'aplicació es deixa en segon pla durant la jornada, no cal tornar a carregar-les.

L'únic problema de rendiment real es troba en els

punts on s'han d'extreure moltes dades amb més de dues consultes enllaçades (en la pantalla de Partit en directe, al entrar als detalls, en les últimes partides jugades amb el mateix personatge a la competició i pel jugador o en equips/competicions/jugadors quan carrega tots els partits de l'equip/competició), que poden arribar a trigar entre 10 i 20 segons a carregar. Per mitigar aquest problema, s'ha dissenyat que aquests widgets carreguin de manera independent a la resta de la pantalla. Això permet que l'usuari pugui consultar tota la resta d'informació sense estar compromès per un widget que trigui més a carregar.

4.3 Rendiment dels models d'aprenentatge automàtic

Per assegurar el bon funcionament, s'han fet molts testos dels models amb moments de la partida en pausa i un test general a la vegada que el test de Rendiment General, on es va mirar en les 5 partides d'una jornada, per veure si el model detectava bé Personatges, Dracs, Runes i si estava en partida. Es va observar que els models presenten un 100% d'efectivitat en les seves prediccions, és a dir, tot el que prediu és correcte. Això es deu al fet que són models senzills dissenyats per reconèixer una imatge o un símbol concret, tasca que aconseguen amb facilitat. En betes inicials de l'aplicació, quan tots els models utilitzaven SVM (Support Vector Machines), independentment del tipus de dataset, no presentaven errors visibles. No obstant això, en analitzar les dades intrínseques, es va observar que en el cas de models amb datasets multiclasse, com en la detecció de personatges, la *predict_proba* (és a dir, la probabilitat de l'imatge que pertanyi a la classe correcta) de les proves eren molt baixes, oscil·lant entre un 2% i un 4%. Tot i que això no generava problemes visibles, no era una situació acceptable per garantir la fiabilitat del sistema, ja que es podia arribar a atribuir a la sort que el model donés correctament el resultat.

Després de realitzar diverses proves i substituir els models multiclasse per models de regressió logística, que són més adequats per aquest tipus de datasets, els resultats van millorar molt. Les *predict_proba* van incrementar a més del 90% en tots els models multiclasse, obtenint així resultats molt més precisos i fiables, ja que vol dir que el model realment sap classificar les imatges.

Aquesta millora significativa en els resultats es deu a la naturalesa intrínseca dels models de regressió logística, que són capaços de gestionar millor la variabilitat i complexitat associada amb els datasets multiclasse. A més, la regressió logística permet una millor interpretació de les probabilitats associades a cada classe, la qual cosa proporciona una major confiança en les prediccions.

4.4 Cost de la base de dades

Com s'ha explicat anteriorment, per a aquesta aplicació s'ha utilitzat el pla gratuït de Firestore [17]. Aquest pla consisteix en 20.000 operacions d'escriptura per dia i 50.000 operacions de lectura per dia, suficients per a un ús personal del projecte. En el cas de voler fer

pública l'aplicació, és necessari calcular el cost real de mantenir l'aplicació per a un ús extensiu a partir de les proves de rendiment general i del testing amb usuaris externs.

Durant el testing de rendiment durant una jornada completa, es va calcular que per una jornada completa d'una lliga són necessàries unes 9.940 escriptures (180 situacions de partida mitjanes * 5 partits per jornada * 11 escriptures per situació + 5 partits * 8 escriptures per partit). Suposant que totes les lligues tenen el mateix format i que, de moment, només continuariem amb competicions de League of Legends, amb aquesta quantitat d'escriptures es podrien cobrir 2 lligues. Per a les jornades regulars, no hi hauria d'haver problema, ja que normalment no coincideixen més de 2 lligues alhora. No obstant això, hi hauria moments on seria necessari donar suport a més lligues, sent el cas màxim que coincideixin 4 lligues, el que implicaria 39.760 escriptures, amb un cost màxim d'1\$ al mes en el pla de pagament de Firestore. Com que és un pla on es paga per ús diari, en el cas d'escriptures ens sortiria a compte passar-nos al pla de pagament, ja que els costos són nuls la majoria dels dies i per un cost extremadament baix tindriem suport quan coincidissin més de dues competicions.

Pel que fa a les lectures, es va calcular que, de mitjana, els usuaris fan unes 700 lectures a la base de dades cada vegada que obren l'aplicació (ja que, en cas que l'aplicació es quedi oberta, les dades consultades es queden a la memòria cau), a partir d'entrar a 2/3 pantalles d'equips, revisar una partida en directe, una partida antiga i 2/3 jugadors/competicions. Si suposem (a partir del testing) que una persona entra unes 6 vegades a l'aplicació durant una jornada completa, veiem que només per un usuari tindria una mitjana de 4.200 lectures. Suposarem que els usuaris mantindran aquesta mitjana diària, ja que encara que ara només es segueixi una lliga, la situació ideal de l'aplicació seria tenir a la base de dades informació de totes les lligues importants que es segueixen, tal com s'ha comentat a l'apartat anterior, amb partits cada dia.

Amb el pla gratuït, tindriem suport per uns 12 usuaris de mitjana, augmentant 6 cèntims el cost per usuari adicional. A partir de les últimes dades, veiem que l'audiència mitja per les 5 grans competicions (sense comptar l'audiència xinesa) són unes 150.000 persones [18]. Per posar un exemple de cost total, si suposem que d'aquesta "audiència fidelitzada" un 30% dels usuaris podrien estar interessats en veure les dades de les competicions (45000 usuaris), ens quedaria un cost total 2700 euros al mes.

Es pot veure que, a partir d'un nombre gran d'usuaris, el cost deixa de ser assumible i s'haurien de buscar formes de monetització alternatives per mantenir l'aplicació. Això podria incloure la publicitat, subscripcions de pagament o col·laboracions amb entitats esportives que es beneficiïn de l'ús de l'aplicació. D'aquesta manera, es garantiria la sostenibilitat del projecte i la seva expansió per a un públic més ampli.

4.4 Testing amb usuaris externs

S'han dut a terme dos tests de l'aplicació amb usuaris externs per recollir feedback valuós. Aquest procés va implicar oferir l'aplicació a 10 persones diferents, algunes amb coneixements d'esports electrònics i altres que no havien interactuat mai amb aquest àmbit. Es van realitzar dos tests: el primer amb una beta inicial que incloïa les funcionalitats implementades, i el segon amb tota la interfície d'usuari (UI) completada. L'objectiu dels tests era que els usuaris provessin d'utilitzar l'aplicació pel seu compte i després se'ls demanava arribar a la pàgina d'un equip o jugador concret.

Durant el primer test, el comentari generalitzat com a resposta a la pregunta "Què milloraries de l'aplicació?" va ser que l'aplicació necessitava una renovació completa de la UI. L'ús de plantilles bàsiques i la manca d'imatges feien que molts usuaris no poguessin identificar clarament en quina pantalla es trobaven. A més, tot i que no va ser un gran problema, alguns usuaris van esmentar que esperar alguns segons per carregar pantalles concretes amb widgets feia que l'experiència no fos del tot satisfactòria.

En el segon test, que es va dur a terme amb una nova UI i amb els widgets de consultes anidades carregant independentment a la pantalla, gairebé no hi va haver queixes i l'experiència dels usuaris va millorar notablement. Els usuaris van tenir menys problemes per arribar a les pàgines de jugadors o equips concrets. Els comentaris en aquesta fase es van centrar més en algunes imatges que trigaven a carregar, un problema associat a la connexió a internet, i en suggeriments per a noves funcionalitats que es podrien afegir en el futur.

Aquests dos tests van ser fonamentals per entendre les necessitats i preferències dels usuaris finals. El feedback rebut va ser essencial per millorar la UI i l'experiència general de l'aplicació, assegurant-se que sigui intuïtiva i eficient. A més, els suggeriments per a futures funcionalitats proporcionen una direcció clara per a la millora contínua de l'aplicació, fent-la més robusta i atractiva per a una audiència més àmplia.

5. MONETITZACIÓ

Com s'ha comentat abans amb el cost, si es donés el cas que l'aplicació tingués èxit entre els usuaris, seria necessari buscar un model de monetització òptim per poder mantenir els costos operatius i, a la vegada, obtenir un rèdit econòmic de l'aplicació.

La primera opció, i probablement la més lògica, seria incloure anuncis dins de l'aplicació. Aquests anuncis podrien ser tant relacionats amb els esports electrònics (ordinadors, perifèrics, merchandising dels clubs) com d'altres tipus. Si mirem aplicacions similars en el món dels esports tradicionals (Besoccer, Sofascore, etc.), es pot veure com predominen els anuncis de cases d'apostes, ideals en aquest context perquè aprofiten la situació del partit en directe i dades addicionals sobre els equips,

jugadors o competicions per facilitar les apostes. Ara que, en els últims anys, ha augmentat l'interès en els esports electrònics per part d'aquestes empreses, no seria sorprenent que estiguessin interessades en col·locar anuncis similars en una aplicació que ofereix el mateix tipus de contingut.

Aquests anuncis podrien estar col·locats en banners a la part inferior de la pantalla, sense molestar excessivament l'usuari. Es podrien combinar amb anuncis de pantalla completa amb els quals l'usuari hauria d'interactuar, que només apareixerien després d'un ús prolongat de l'aplicació o cada cert nombre de pantalles carregades. Tot i així, cal tenir en compte que aquests últims poden resultar massa invasius en l'experiència de l'usuari.

Una altra manera típica de monetitzar l'aplicació seria a través de la recollida de dades d'ús dels usuaris dins de l'aplicació. Per una banda, es podrien aconseguir dades pròpies dels usuaris com localització, temps d'ús, etc. Per altra banda, es podrien obtenir dades sobre el contingut de l'aplicació: quin és l'equip amb més visites, temps mitjà a cada pantalla, clics als enllaços de les xarxes socials d'un jugador en concret, etc. Aquestes dades són extremadament útils tant pels propis clubs com per potencials anunciants o marques relacionades amb la competició, aconseguint així una rendibilitat superior a la dels simples anuncis i sense modificar l'experiència de l'usuari. Igualment, cal tenir en compte les múltiples regulacions existents en aquesta matèria per no tenir futurs problemes en la recollida, ús i venda de les dades.

Una altra manera menys intrusiva seria arribar a acords de col·laboració amb equips concrets per promocionar el seu contingut (posts a les xarxes socials, vídeos, etc.) dins de la seva pantalla a l'aplicació, augmentant així l'exposició de l'equip. Aquests acords podrien incloure fins i tot una pestanya pròpia a la pantalla de l'equip on aparegués la botiga oficial, permetent comprar merchandising directament des de l'aplicació o oferint promocions exclusives per als usuaris.

Per últim, encara que no és l'ideal, es podria implementar un model *freemium* [19], on part de l'aplicació és gratuïta i una altra part és de pagament, ja sigui amb un pagament únic o amb un servei de subscripció mensual. Hi ha diverses opcions per a aquest model, com ara tenir un límit de pantalles diàries per usuari, la qual cosa, per una banda, reduiria els costos de la base de dades, ja que es podria controlar millor les lectures per usuari, i els que paguessin cobririen el seu cost amb la subscripció. Una altra opció seria limitar l'accés a certes funcionalitats, com ara no poder veure dades en concret fins que la partida en directe no hagi acabat.

Hi ha diverses estratègies de monetització que es poden explorar per mantenir i rendibilitzar l'aplicació. La clau està en trobar un equilibri entre obtenir ingressos i mantenir una bona experiència d'usuari. Un enfocament

combinat que utilitzi una selecció acurada d'aquestes estratègies podria ser la solució òptima per assegurar la sostenibilitat i l'èxit continuat de l'aplicació.

6 CONCLUSIONS I TREBALL FUTUR

Després de tot, s'ha aconseguit desenvolupar una aplicació completament funcional que compleix amb tots els objectius proposats. Aquesta aplicació disposa d'un sistema d'extracció de dades suficientment robust per gestionar errors, acompanyat de models d'aprenentatge automàtic capaços de detectar i processar aquestes dades de manera eficient. La base de dades és prou completa per permetre realitzar proves exhaustives, i el codi de l'aplicació està correctament integrat amb el sistema d'extracció de dades.

A més, l'aplicació inclou totes les funcionalitats previstes inicialment i presenta una interfície atractiva per als usuaris. El rendiment de l'aplicació és adequat per a un ús habitual, amb temps de càrrega mínims que assegurin una experiència d'usuari fluida i satisfactòria.

Un altre aspecte important ha estat la realització d'un estudi sobre els costos i les possibles vies de monetització de l'aplicació en cas que es decideixi publicar-la i aquesta tingui èxit. L'estudi inclou diverses estratègies de monetització, com la integració d'anuncis publicitaris, la recollida i anàlisi de dades d'ús, col·laboracions amb equips i la implementació d'un model *freemium*. Aquestes opcions permetrien mantenir els costos operatius i obtenir rendiments econòmics, tot i garantint una bona experiència d'usuari.

En conclusió, el projecte ha aconseguit desenvolupar una aplicació que no només és tècnicament sòlida sinó que també té potencial per ser comercialment viable. La combinació d'un sistema d'extracció de dades robust, models d'aprenentatge automàtic eficients, una base de dades completa i una interfície d'usuari ben dissenyada asseguren que l'aplicació pugui oferir una experiència de qualitat als usuaris. A més, l'estudi de costos i monetització proporciona una base sòlida per a la futura expansió i sostenibilitat de l'aplicació en el mercat.

Amb tot això, si es continua amb el projecte de l'aplicació, encara es poden millorar diversos aspectes per assegurar una experiència d'usuari òptima i una operació més eficient:

- Millorar el rendiment de les consultes anidades: Fins ara, l'aplicació "esquiva" aquest problema fent que les consultes carreguin independentment. Tot i això, s'haurien de veure si aplicar mètodes d'indexació per reduir el temps de càrrega i millorar l'eficiència general del sistema.

- Millorar la interfície d'usuari (UI): Actualment, la UI ha estat dissenyada sense una capacitat professional específica per al disseny gràfic, cosa que pot resultar en una experiència d'usuari menys polida. És crucial col·laborar amb dissenyadors professionals per assegurar que la UI sigui atractiva, intuïtiva i competitiu en el

mercat de les aplicacions. Aquestes millores podrien incloure la revisió de la paleta de colors, la tipografia, i la disposició dels elements visuals per assegurar una experiència més coherent i atractiva.

- Expandir la base de dades: En l'actualitat, la base de dades de prova només conté informació d'una competició específica d'un any concret. Perquè l'aplicació sigui realment útil i atractiva per a un públic més ampli, és necessari expandir aquesta base de dades per incloure les dades de totes les grans lligues i competicions, tant històriques com actuals. A llarg termini, també s'haurien d'afegir dades de nous jocs per reflectir la diversitat de l'escena dels esports electrònics.

BIBLIOGRAFIA

- [1] "Artificial intelligence (AI) applied to informative documentation and journalistic sports writing. The case of BeSoccer", Jesús Segarra-Saavedra, F. Javier Cristòfol, Alba-Maria Martínez-Sala, 12 pàg., 14-01-2019, Última Entrada: 03/05/24
<https://core.ac.uk/download/pdf/275661911.pdf>
- [2] "Sofascore", Sofascore team, Última Entrada: 9/03/2024
<https://www.sofascore.com/>
- [3] "Onefootball", Sofascore team, Última Entrada: 03/05/2024
<https://onefootball.com/es/inicio>
- [4] "League of Legends", Riot Games, Última entrada: 9/03/2024
<https://www.leagueoflegends.com/es-es/>
- [5] "LEC 2023", Leaguedpedia Team, Última entrada: 26/05/2024
https://lol.fandom.com/wiki/LEC/2023_Season
- [6] "scikit-learn API Library", scikit-learn programming team, Última entrada: 26/05/2024
<https://scikit-learn.org/stable/modules/classes.html#>
- [7] "Digital image processing and recognition using Python", Sri Shreya, IJEAST, 02-2021, 4 pàg., Última entrada: 20/04/2024
<https://www.ijeast.com/papers/319-322,Tesma510,IJEAST.pdf>
- [8] "Behind the Game: Exploring the Twitch Streaming Platform", Jie Deng, Félix Cuadrado et al., 7 pag, Última entrada: 9/03/2024
https://www.researchgate.net/publication/285055799_Behind_the_Game_Exploring_the_Twitch_Streaming_Platform
- [9] "Understanding logistic regression analysis", Sandro Sperandei, 09-2013, Última entrada: 6/05/2024
https://www.biochemia-medica.com/assets/images/upload/xml_tif/Sperandei_S_Logistic_regression_analysis.pdf
- [10] "A Comprehensive Survey on Support Vector Machine in Data Mining Tasks: Applications & Challenges", H. S. Behera et Al., Nadia, 18 pàg., 2015, Última entrada: 9/03/2024
http://article.nadiapub.com/IJDTA/vol8_no1/18.pdf
- [11] "Selenium with Python", Baiju Muthukadan,
<https://selenium-python.readthedocs.io>
- [12] "OpenCV", OpenCV programming team, Última entrada: 20/04/2024
<https://docs.opencv.org/4.x/>
- [13] "Python Tesseract", Samuel Hoffstaetter, Juarez Bochi, Matthias Lee, Lars Kistner, Ryan Mitchell, Emilio Cecchini, Última Entrada: 09/04/24
<https://github.com/h/pytesseract>
- [14] "Flutter for cross-platform App and SDK development", Lukas Dagne, 05-2019, Última entrada: 3/05/2024
<https://www.theseus.fi/bitstream/handle/10024/172866/Lukas%20Dagne%20Thesis.pdf>
- [15] "Flutter documentation", Flutter Development Team, Última entrada: 25/06/2024
<https://docs.flutter.dev/>
- [16] "Firestore: The NoSQL Serverless Database for the Application Developer", Ram Kesavan et al., IEEE, 13 pàg., 2023., Última entrada: 5/05/2024
<https://ieeexplore.ieee.org/abstract/document/10184529#citation>
- [17] "Firestore Pricing", Firestore Development Team, Última entrada: 25/06/2024
<https://firebase.google.com/pricing?hl=es-419>
- [18] "ESCharts", ECharts Team, Última entrada: 25/06/2024
<https://echarts.com/tournaments/lol/>
- [19] "Mobile App Monetization: App Business Models in the Digital Era", Ailie K. Y. Tang, IJIMT, 2016, 4 pàg., Última entrada: 25/06/2024
<https://www.ijimt.org/vol7/677-MB00017.pdf>