

---

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

Cruz Galvis, Jorge Ivan; Antens, Coen Jacobus , dir. Una App que ajuda a reciclar. 2023. (Enginyeria Informàtica)

---

This version is available at <https://ddd.uab.cat/record/280732>

under the terms of the  license

# Una App que ajuda a reciclar

Jorge Ivan Cruz Galvis

**Resum**— Aquest document descriu el procés de desenvolupament d'una aplicació per a dispositius mòbils que pretén ajudar a les persones a l'hora de reciclar. Aquest projecte té dos objectius principals, el primer és el de poder identificar un producte quotidià de supermercat fent una fotografia amb el mòbil i amb aquest, indicar a l'usuari a quin contenidor d'escombraries s'ha de disposar. El segon objectiu és el de recopilar anotacions de productes amb l'ajuda de l'usuari mitjançant l'aplicació. Aquest objectiu es basa en el cas que el producte fotografiat no es trobés en el sistema, l'aplicació pot iniciar un procés d'anotació en el qual mitjançant una simple descripció del producte i una sèrie de fotografies de distints angles, l'usuari haurà de dibuixar un rectangle identificant la ubicació i les dimensions de l'objecte. Per assolir aquests objectius es disposa d'un servidor que compta amb xarxa neuronal ja entrenada per identificar una sèrie de productes de supermercat i una base de dades on s'emmagatzemaran les anotacions per després afegir nous elements a la mostra d'entrenament. En aquest projecte es desenvoluparà la connexió i la transferència de dades entre aquests i l'aplicació, a més de crear una interfície simple i amigable per l'usuari.

**Paraules clau**—Android, Aplicació, Classificació, Reciclatge, Residus

**Abstract**— This document describes the development process of an app for smartphones that is aimed at helping people recycle. This project has two main objectives, the first is to be able to identify a normal supermarket product by taking a photo using the smartphone and after, tell the user on which bin has to recycle the product. The second objective is to add more labeling data using the app. This objective is based on the particular case that the object is not in the system, so the app will start a labeling process in which the user has to describe the product with a simple sentence and then, through taking a series of photos of various angles of the product, has to draw rectangles identifying the location and size of the object. In order to achieve these objectives, a web server is available on which a neuronal network is deployed and already trained to recognize a set of supermarket products. Also, a database is available to store the labeling data to be used as training data. This project will develop the connection and the data transfer between the server and database to the mobile application, and last but not least, create a simple and user-friendly interface.

**Index Terms**— Android, App, Labeling, Recycling, Waste



## 1 INTRODUCCIÓ - CONTEXT DEL TREBALL

EL reciclatge és un deure molt importat en la nostra societat, no només per cuidar el planeta sinó també per sostenir el nostre futur. La campanya publicitària impulsada per la Generalitat anomenada "Envàs on vas" [1] recentment va fer deu anys que es va crear i sembla que actualment reciclem més, però encara no és suficient i sobre tot encara no es classifiquen bé els residus. Un reportatge de l'any 2020 de l'agència EFE Verde va dir que 4 de cada 10 espanyols diuen no reciclar per por a equivocar-se, que només un 3% ho té clar i que un 65% admeten haver introduït objectes al contenidor erroni [2].

A més, partir d'aquesta campanya publicitària es va crear una aplicació [www.residuonvas.cat](http://www.residuonvas.cat) [3] per tal de donar resposta a les qüestions sobre a quin contenidor

correspon determinat producte, aquesta aplicació està disponible per a smartphones i web i la qual respon a la pregunta d'on llençar el producte mitjançant una sèrie de descripcions de l'objecte per part de l'usuari (veure Apèndix A). Aquesta aplicació, però, està limitada pel nombre de productes que existeixin en la seva base de dades i, per tant, pot ser que no pugui respondre a les necessitats dels usuaris.

Des del CVC (Centre de Visió per Computador) de la UAB proposen que per a aquesta tasca s'hauria d'utilitzar fotografies i mitjançant una xarxa neuronal ja desenvolupada, identificar el producte i donar la resposta a l'usuari (veure Fig. 1). A més, es suggereix que en el cas que el producte no sigui en el sistema, oferir l'opció d'afegir-lo a la base de dades per poder reconèixer més productes en el futur.

Amb aquest projecte es pretén crear una aplicació per dispositius mòbils que respongui a aquestes propostes i necessitats per tal de complir l'objectiu final que és el d'ajudar a la ciutadania a saber com reciclar i impulsar encara més la filosofia del reciclatge que avui en dia més que una necessitat és una obligació.

- E-mail de contacte: [jcruzgalvis@gmail.com](mailto:jcruzgalvis@gmail.com)
- Menció realitzada: *Enginyeria del Software*
- Treball tutoritzat per: *Coen Atens(Ciències de la computació)*
- Curs 2022/23

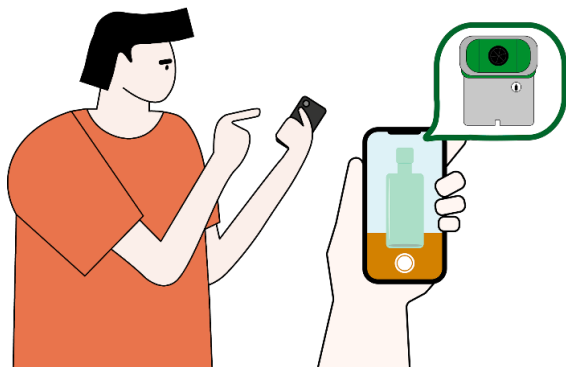


Fig. 1. Concepte del projecte

## 2 OBJECTIUS

En aquesta secció es descriuen els objectius del projecte, el seu propòsit i el procediment que es seguirà per aconseguir complir aquests objectius.

- El primer és el de desenvolupar una aplicació per a dispositius mòbils amb la qual l'usuari pugui entendre com reciclar determinat producte de manera fàcil amb una fotografia del producte.
- El segon objectiu i relacionat amb el primer, és el d'habilitar l'aplicació perquè es comuniqui amb el servidor del CVC per tal d'enviar les fotografies i rebre la resposta de la xarxa neuronal.
- El tercer objectiu és el de proporcionar noves dades a la IA per tal de poder reconèixer més productes, això s'aconseguirà fent diverses fotografies des de diferents angles, marcant amb un rectangle la ubicació i dimensions del producte, i per últim donant una descripció de l'objecte per tal d'augmentar la mostra. Tota aquesta informació s'enviarà a una base de dades que utilitzarà la xarxa neuronal.

## 3 ESTAT DE L'ART

Aquest projecte va estar inspirat a partir de dues iniciatives al voltant del reciclatge, la ja mencionada campanya publicitària Envàs on Vas, i a partir d'aquesta va sorgir una altra que cerca facilitar la classificació dels residus, l'aplicació web *residuonvas.cat*.

Aquesta última aplicació disposa d'un ampli catàleg de residus on l'usuari pot cercar de diverses maneres el seu producte. Un cop trobat, l'aplicatiu indica a quin contenidor s'ha de llençar, quin és el seu procés de reciclatge i en quin tipus d'infraestructura es processa.

### Aplicacions mòbils

Pel que fa a aplicacions mòbils, no n'hi ha moltes, però les trobades es poden classificar de dues maneres:

1. Iniciatives impulsades per organitzacions públiques que fomenten el reciclatge mitjançant recompenses en forma de crèdits.
2. Altres apps de reciclatge.

Entre les de la primera classe podem trobar varies com "RECICLOS" (veure Fig. 2), "Recicla y suma", "Recicla y Gana", etc. Les funcionalitats d'aquestes aplicacions segueixen un patró molt similar: primer es dona d'alta un compte, es mostren els productes que es poden reciclar classificats per material, per obtenir els crèdits s'ha d'escanejar el codi de barres de cada residu i a més s'han de llençar en contenidors especialitzats per aquest tipus d'activitat, ja que necessiten escanejar el telèfon des del contenidor.



Fig. 2. Funcionament de RECICLOS

La segona classe defineix a una aplicació en concret que s'anomena Aire [5], aquesta té una característica única que es diferencia amb les altres, es tracta d'un xatbot (veure Fig. 3). Aquesta aplicació simula l'aparença d'una app de xat i la seva funcionalitat consisteix a escriure pel xat preguntant a on es llença un producte en concret llavors aquest bot dona la resposta. Una altra funcionalitat molt relacionada amb aquest projecte és l'opció a pujar una imatge pel xat, aquesta va ser provada per mi, però no va retornar cap resposta.

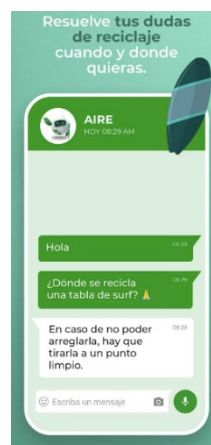


Fig. 3. App Aire

Segons la descripció de l'aplicació a la PlayStore aquesta utilitza Machine Learning (ML) per tal d'augmentar la seva base de dades quan més quèries rep. Encara que no es menciona el model específic de ML, un

dels possibles models podria ser CLIP (Con-contrastive Language-Image Pre-Training). Aquest model va ser desenvolupat per la companyia OpenAI, i consisteix en una xarxa neuronal que aprèn conceptes visuals a partir del llenguatge natural supervisat [6]. Aquest té la capacitat de classificar el text buscant paraules claus i així determinar, en aquest cas, el producte a reciclar.

### Projectes de detecció d'objectes

Durant la cerca d'altres projectes amb temàtica similar de detecció de productes es va trobar un parell, els quals fan ús d'algorismes de detecció d'objectes i intel·ligència artificial (IA) per ubicar i classificar els articles de supermercat.

El primer es tracta de la supervisió de l'estoc de productes en els prestatges de la botiga. Fa ús de càmeres i un sistema amb IA (Shelfwatch) en el núvol per tal de processar les imatges i retornar una resposta als dispositius del personal amb la ubicació, la classificació i l'alerta per si cal més estoc [7].

El segon fa ús de YOLOv5 (xarxa neuronal) per la detecció de múltiples objectes en els prestatges i així determinar l'estoc disponible [8].

## 4 METODOLOGIA

La metodologia que es va seguir pel desenvolupament del projecte va ser una *Agile*, la qual es va basar en iteracions i reunions setmanals amb el client per donar i rebre feedback sobre l'avenç del projecte.

### 4.1 Eina de seguiment

Per poder fer un seguiment exhaustiu del desenvolupament del projecte es va utilitzar l'eina Jira, amb la qual es van registrar i gestionar els sprints i les tasques agrupades en el *product backlog* o cartera de productes. Aquesta eina compta d'un servei de control de cada tasca amb la qual es canviarà l'estat d'aquesta. Aquests estats poden ser:

- **Tasques per fer:** Estat inicial en el qual es troba la tasca en ser creada, es fixen els detalls com el nom, data i personal assignat a aquesta.
- **En curs:** El moment que la persona assignada comença a fer la tasca
- **Finalitzat:** Un cop s'han completat les activitats associades a la tasca

Cadascuna d'aquestes tasques es troben agrupades en un *Sprint* o iteració, cada iteració està limitat a una duració d'entre dues o tres setmanes per tal d'agilitzar el projecte i fer entregues contínues. La iteració finalitza

un cop s'han completat les tasques corresponents i en el temps fixat.

## 5 PLANIFICACIÓ

Per fer el seguiment i gestionar la planificació es va fer ús de l'eina de Microsoft Project. Al ser una eina ja coneguda es va poder crear i gestionar de forma àgil el progrés de les tasques i recursos assignats en tot el recorregut que va durar el projecte.

El desenvolupament del projecte es va dividir en quatre fases:

- **Anàlisi de requisits**

Es recopilen els requisits a partir de reunions amb el client, analitzant productes similars i projectes ja desenvolupats (com la ja citada 'residuonvas')

- **Disseny i planificació**

A partir dels requisits es fa el disseny i l'arquitectura inicial del projecte, a més es fa la planificació de les activitats i les iteracions.

- **Desenvolupament en iteracions**

S'apliquen els mètodes i mitjançant les eines disponibles i el coneixement s'executaran les tasques designades a cada iteració. Es fa el control de versions i proves. Al final de l'última iteració es fa un test d'acceptació per confirmar la qualitat del producte amb el client.

- **Tancament**

Un cop totes les tasques han sigut completades i les iteracions tancades, es procedirà a finalitzar la documentació, es farà el manteniment de l'aplicació i s'efectua l'entrega del producte final. Finalment, es donarà per finalitzat el projecte en la seva totalitat.

En la següent taula es descriuen les activitats a realitzar associades a cada fase del projecte.

Id	Nom de la tasca
1	<b>App que ajuda a reciclar</b>
2	<b>Anàlisi de requisits</b>
3	Entrevista amb el client
4	Elaboració dels requisits
5	<b>Disseny i planificació</b>
6	Disseny del sistema
7	Planificació
8	Informe inicial
9	<b>Desenvolupament</b>
10	<b>Iteració 1</b>
11	Creació esquelet de l'aplicació
12	Sistema de fitxers
13	Comunicació amb el servidor

14	Testeig
15	<b>Iteració 2</b>
16	Funcionalitat càmera
17	Funcionalitat resposta a l'usuari
18	Testeig
19	Informe de progrés I
20	<b>Iteració 3</b>
21	Comunicació amb la Base de dades
22	Dibuixar sobre imatge
23	Procés de classificació
24	Testeig
25	<b>Iteració 4</b>
26	Creació de tutorial
27	Informe de progrés II
28	Internacionalització
29	Testeig final i acceptació
30	<b>Tancament</b>
31	Documentació final
32	Proposta informe final
33	Tancament del projecte

Taula 1. Activitats del projecte

## 5.1 Diagrama de Gantt

El Diagrama de Gantt fet amb l'aplicació de Microsoft Project proporciona una visió gràfica i temporal de les tasques a realitzar en el temps predeterminat pel desenvolupament i finalització del projecte.

En la següent figura es presenta un diagrama de Gantt simplificat fet amb Jira mostrant les tasques principals i les iteracions en una línia temporal.



Fig. 4. Diagrama de Gantt simplificat

A més, a l'apèndix es troba el diagrama de Gantt complet fet amb Microsoft Project (veure Apèndix B).

## 6 REQUISITS

En aquesta secció es descriuen els requisits del sistema recollits al llarg de la fase d'anàlisi de requisits i estan agrupats per funcionals i no funcionals. Aquests requisits estan classificats segons la metodologia MoSCoW [9], la qual prioritza aquestes activitats segons la seva

importància en el desenvolupament del projecte (veure Taula 2 i Taula 3).

Aquestes es classifiquen segons la prioritat:

- **(M)ust have:** Essencial pel funcionament i la integritat del sistema.
- **(S)hould have:** Afegeix valor a l'aplicació i per tant, s'hauria de fer.
- **(C)ould have:** Afegeix valor, però no afectaria en res si no es fes en aquest moment, s'afegeix al *product backlog*.
- **(W)on't:** No és imprescindible, però es pot afegir al *product backlog* per si dona temps i no incrementa l'esforç.

### 6.1 Requisits funcionals

Requisit	Prioritat
L'aplicació ha de poder comunicar-se amb el servidor	M
L'aplicació ha de permetre fer fotografies	M
L'aplicació ha de permetre seleccionar una fotografia de la galeria	S
L'aplicació ha de poder mostrar la resposta del servidor de forma simple i amigable	S
L'aplicació haurà d'ensenyar a l'usuari com classificar un producte mitjançant un mini tutorial	S
El sistema ha de permetre dibuixar un rectangle en la imatge	M
El sistema ha de permetre descriure el producte amb una paraula (classe del producte)	M
L'aplicació haurà d'enviar la informació a la base de dades	M

Taula 2. Requisits funcionals

### 6.2 Requisits no funcionals

Requisit	Prioritat
El temps de resposta màxim ha de ser de 3 segons	C
El llenguatge de programació del sistema és Dart	M
El <i>framework</i> de desenvolupament utilitzat pel desenvolupament és Flutter	M
La transferència de dades es farà mitjançant el protocol HTTP	C
El número màxim de fotografies a classificar és 10	S
El color del rectangle dibuixat serà blau	C
L'aplicació haurà de ser desenvolupada	M

per dispositius Android	
La paleta de colors de l'aplicació ha d'estar relacionada amb la temàtica del reciclatge	C
El format de descripció de les dades de classificació ha de ser JSON	S
El text de l'aplicació haurà de ser multilinguatge (Català, Castellà i Anglès).	C

Taula 3. Requisits no funcionals

## 7 ANÀLISI I DISSENY

En la següent figura es mostra un primer disseny de l'arquitectura del projecte.

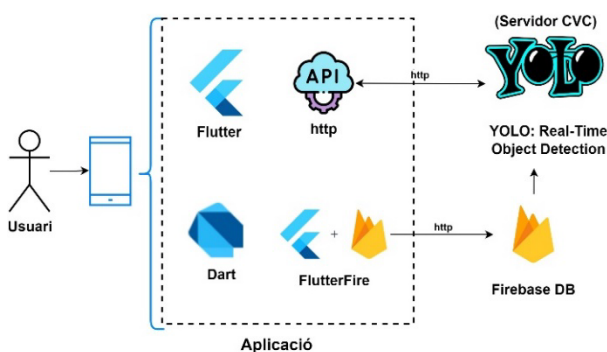


Fig. 5. Arquitectura del projecte

Com es pot observar a la imatge, l'aplicació es va desenvolupar utilitzant el *framework* de Flutter i amb el llenguatge de programació Dart. Amb aquest es té l'avantatge que l'aplicació podrà ser distribuïda tant per dispositius Android, iOS, web, etc. A més, es podrà fer una ràpida implementació, ja que es tenen coneixements previs sobre l'eina.

Per altra banda, tenim dues llibreries, la primera 'http' es tracta d'una llibreria Dart que permet fer peticions HTTP. Amb aquesta es realitzarà la comunicació amb el servidor per tal d'enviar les fotografies i rebre la resposta del servidor. La segona es tracta d'un grup anomenat FlutterFire, amb aquesta som capaços d'aplicar les funcions i comandes per accedir a la base de dades i operar sobre aquesta.

Per a aquest projecte comptem amb el servidor del CVC disposa d'una xarxa neuronal que utilitza l'algorisme de YOLO versió 3, amb aquest som capaços de detectar quin producte és el fotografiat per donar resposta a l'usuari.

YOLO (You Only Look Once) és un sistema de detecció d'objectes en temps real que processa imatges a 30 fotogrames per segon i s'utilitza en tota mena de projectes de detecció d'objectes. Per tal de detectar l'objecte s'usa una CNN (Convolutional Neural Network) per escanejar la imatge i predir els límits del rectangle que ubica

l'objecte i la seva probabilitat [11].

Com a mètode inicial, emmarca la detecció d'objectes com un problema de regressió delimitant quadres separats espacialment i calculant les probabilitats de classes associades a l'objecte. Per la predicció fa ús d'una única xarxa neuronal per una imatge completa en una única avaluació [12].

### 7.1 Diagrama de casos d'ús

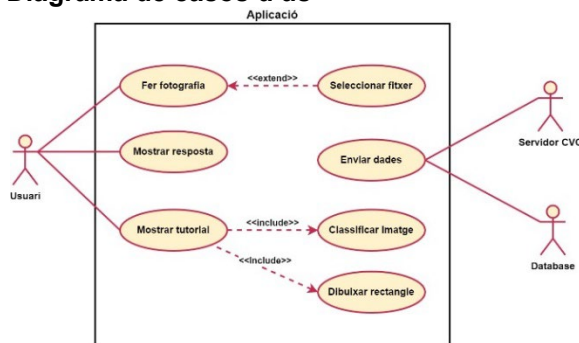


Fig. 6. Diagrama de casos d'ús

Per representar les activitats principals del sistema i com interactuen amb els diferents actors que participen activament i passivament, s'ha escollit un diagrama de casos d'ús (veure Fig. 6).

La funcionalitat principal del sistema consta d'un procés inicial en el qual mitjançant una llibreria de Flutter, l'aplicació és capaç de transferir la imatge cap al servidor i rebre la sortida de la xarxa neuronal.

En el cas que el producte no existeixi en la base de dades es pregunta a l'usuari si vol cooperar per afegir aquest element a la mostra, llavors el sistema inicia el procés de classificació. Un cop identificat el producte amb un rectangle i amb una classe, s'utilitza l'aplicació per enviar la informació a la base de dades.

### 7.2 Alternatives de disseny

Una alternativa que es va pensar a l'hora de fer el reconeixement del producte va ser implementar la xarxa neuronal directament en l'aplicació mòbil mitjançant la conversió del model de YOLO versió 3 actual a "tflite" (TensorFlow Lite) per tal de poder fer ús de la seva llibreria per desenvolupar el model en Flutter.

Amb aquesta opció cap la possibilitat de poder fer el reconeixement en temps real només enfocant l'objecte amb la càmera. Així i tot però, aquesta opció té uns inconvenients com la versió del sistema (versió 3), que és vell i s'hauria d'actualitzar a la versió 5, les actualitzacions periòdiques i el pes de l'aplicació que fan que no es posicioni com disseny principal.

### Altres algorismes de detecció d'objectes

Entre els algorismes que més s'utilitzen a l'hora de desenvolupar xarxes neuronals per la detecció d'objectes es troben el ja mencionat YOLO, Fast R-CNN i SSD.

Fast R-CNN és la versió més ràpida de R-CNN, una xarxa neuronal convolucional (tipus de xarxa neuronal que fa servir matrius bidireccionals per fer l'anàlisi i que és efectiva per tasques de visió artificial) que es basa en la selecció de milers de regions en la imatge pel seu processament. Es tracta d'un procés més lent que l'algorisme anterior, ja que aplica una xarxa neuronal per cada regió [12].

SSD (Single-Shot Detector), un algorisme eficient per la detecció i classificació d'objectes que fan servir les xarxes convolucionals. Es tracta de dividir la imatge en cel·les per processar-les individualment amb una xarxa neuronal. En comparació amb altres algorismes, té millor precisió amb imatges de mida petita [15].

### Base de dades

Per part de la comunicació amb la base de dades, existeix una alternativa que suposa la utilització d'una API de la plataforma Supervisely. Supervisely és una plataforma web molt útil a l'hora d'aportar recursos per l'entrenament de les xarxes neuronals, gràcies a aquesta s'és capaç de classificar i etiquetar imatges per la mostra a utilitzar. Aquesta plataforma és actualment feta servir pel CVC i amb la seva API podríem ser capaços de classificar directament des de l'aplicació i llavors enviar la informació a aquesta mateixa base de dades. A la Fig. 7 es pot observar un exemple de l'aplicació.

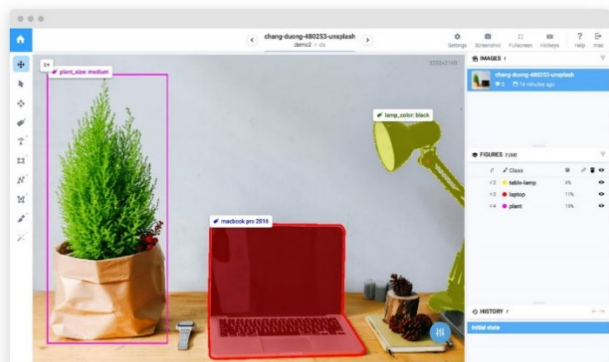


Fig. 7. Aplicació Supervisely

### Plataforma de desenvolupament

Per a aquest projecte es fa ús de Flutter per tal de desenvolupar de forma àgil l'aplicació. Però n'hi ha alternatives que són igual de vàlides, però que no van ser la prioritat.

Per una banda, està Android Studio, el qual a diferència de Flutter, treballa principalment amb codi Java i C++ (encara que pot suportar tots els llenguatges de l'IDE

IntelliJ a més de Kotlin i GO), no té la funcionalitat de "Hot Reload" (Aplicar els canvis immediatament mentre s'executa l'aplicació), té una major comptabilitat amb els serveis de Google i adopta les actualitzacions d'Android molt més ràpidament. No pot desenvolupar aplicacions per iOS.

Finalment, tenim Xamarin, una plataforma de desenvolupament .NET creada per Microsoft i que ens permet la creació d'aplicacions multiplataforma. El llenguatge que s'utilitza és C#. Encara que el seu rendiment sigui a prop de les aplicacions natives, manca en la facilitat de creació de gràfics, la integració de funcionalitats natives d'Android i la complexitat del llenguatge pot ser un obstacle pel ràpid desenvolupament.

## 8 RESULTATS

En aquesta secció es descriu el resultat final del desenvolupament del projecte. Es detallen els primers passos del desenvolupament i cadascuna de les pantalles creades amb la seva funcionalitat i les eines i mètodes utilitzats per la seva codificació.

Primerament, es va construir un prototip de l'aplicació i el seu funcionament, el qual es va presentar al tutor del treball per obtenir la seva aprovació. A partir d'aquest prototip es van desenvolupar les distintes pantalles.

A continuació es presenten aquestes amb una captura de pantalla real i una descripció dels elements presentats i la seva funcionalitat implementada per darrere.

### 8.1 Pantalla principal

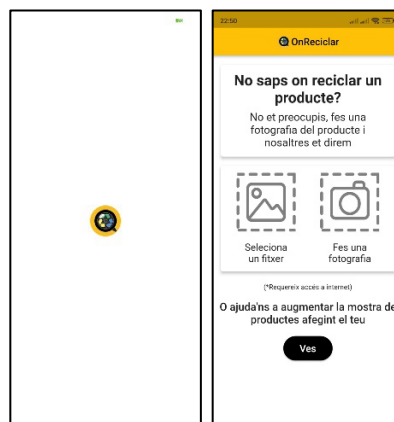


Fig. 8. Splash screen i Pantalla principal

A la Fig. 8 podem observar el Splash Screen i la pantalla principal. El primer es mostra en iniciar l'aplicació i està formada per la icona de l'aplicació i un fons en blanc.

A continuació tenim la pantalla principal conformada per elements de text i dos botons, aquests botons

representen dues funcionalitats.

La primera és la de fer una fotografia amb el telèfon, molt important, ja que l'objectiu principal del projecte és el de poder donar la resposta a l'usuari en el mateix moment que no sàpiguen on es recicla un producte. Aquesta funcionalitat s'ha desenvolupat fent ús d'una llibreria de Flutter anomenat "camera" amb aquest som capaços d'obrir la càmera del dispositiu per obtenir el fitxer de fotografia.

Per l'altra banda, la segona funcionalitat de seleccionar un fitxer de la galeria s'ha fet ús de la llibreria "image\_picker", la qual el configurem per obtenir la imatge i comprimir-la a una qualitat del 60% per reduir la seva dimensió abans d'enviar-ho al servidor.

A la part inferior es mostra un text i un botó amb el qual podem fer el procés de classificació directament, ja que el botó ens encamina cap a la pantalla Tutorial.

### 8.2 Pantalla càmera

La càmera utilitzada té la seva pròpia pantalla on fer les accions apropiades segons la situació. Com ja es va comentar a l'apartat anterior, es va fer ús d'una llibreria anomenada "camera". Aquest es va escollir per la seva simplicitat i major control sobre les operacions bàsiques de càmera.

Per tal de presentar una ràtio de càmera quadrat, es va fer ús d'una llibreria anomenada "cus-tom\_ratio\_camera". Amb aquesta som capaços d'emascarar la pantalla de càmera en un aspecte d'1:1, és important, ja que la CNN treballa amb imatges d'aquest aspecte i amb una mida de 500x500 píxels. Es va implementar d'aquesta manera per tal de reduir la mida de les dades enviades per HTTP i adaptar les imatges a les mateixes que ja fa servir la xarxa neuronal.

#### Retall i canvi de mida de la imatge

A més, aquesta llibreria ens indica les coordenades del quadrat amb la seva altura i amplada. Amb aquesta informació és possible fer el retall de la imatge i a continuació reduir les seves dimensions per ajustar-se fins als 500x500 píxels.

#### Funcionalitat càmera segons la situació

A la Fig. 9 es mostren dues situacions on la càmera es comporta de manera diferent. La primera es crida des de la pantalla principal, en aquesta s'especifica que només volem fer una fotografia, llavors un cop feta es passa directament a la pantalla Imatge. La segona situació succeeix quan es crida des de la pantalla Tutorial, en aquest cas es permet fer més d'una fotografia en la mateixa pantalla fins a un màxim de 10. Un cop l'usuari

no vol fer-ne més o arriba al màxim, el botó del costat retorna a l'anterior pantalla.

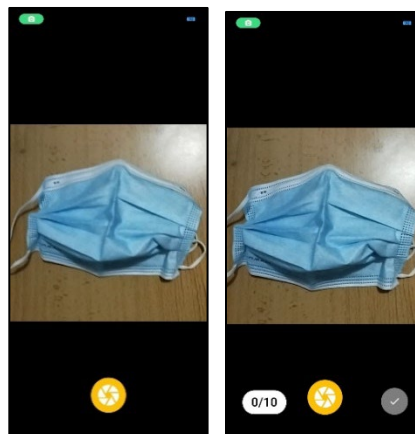


Fig. 9 Pantalla càmera situació 1 i situació 2

### 8.3 Pantalla imatge

Consta d'un widget per mostrar la fotografia recentment capturada i d'un botó amb funcionalitat. Abans d'enviar la foto al servidor web es presenta la mateixa a l'usuari per tal de confirmar-ho (veure Fig. 12).

#### Connexió amb el servidor

El botó d'enviar de la d'aquesta pantalla té la funcionalitat de connectar l'aplicació amb el servidor, enviar el fitxer i esperar per la resposta.

En la següent imatge (Fig. 10) es mostren les crides HTTP que es fan entre l'aplicació i el servidor per tal d'intercanviar informació.

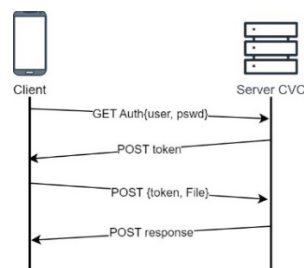


Fig. 10. Connexió client-servidor

Primerament, el client demana un token al servidor. Això ho fa enviant una crida GET amb capçalera de tipus autenticació, aquesta es compon per un objecte tipus JSON amb les dades de nom d'usuari i contrasenya. Aquest objecte es xifra abans d'enviar la petició.

Un cop el servidor rep la petició, envia la resposta amb el token, aquest té una validesa de 30 minuts. Per tal de poder trametre la imatge ja amb el token, en les capçaleres de la petició POST, s'especifica el tipus de dades, en aquest cas és 'multipart/form-data'. Amb aquesta

configuració som capaços d'enviar fitxers de qualsevol mena transformant les dades en format binari.

### Resposta del servidor

Finalment, un cop enviades les dades i rebem la resposta, la descodifiquem a un format JSON perquè pugui ser accessible des de l'aplicació.

Aquesta resposta està formada per quatre elements (Fig. 11), el primer indica el nom del contenidor al qual s'ha de llençar el producte [plastic, glass, paper, waste, organic]. El segon descriu el nom del producte, el tercer descriu les dimensions del rectangle i la seva posició en la imatge processada. Per últim, s'indica la puntuació de la xarxa neuronal sobre la correcta predicció de l'objecte.

```
{
  "bin": "plastic",
  "product": "bleach",
  "rectangle": {
    "h": 3554,
    "w": 793,
    "x": 1234,
    "y": 583
  },
  "score": 0.9746841794967651
}
```

Fig. 11. Exemple de resposta del servidor

En la següent figura es mostren els distints estats de la pantalla Imatge. La primera mostra l'estat abans de prémer el botó d'enviar, un cop pres el botó es mostra el Widget d'animació de càrrega. Aquesta està activa mentre s'espera la resposta del servidor, si el servidor no està actiu, s'ha expirat el certificat o el token, o si no hi ha internet en el dispositiu es mostra la tercera pantalla comunicant a l'usuari del problema i es retorna a la pantalla principal. Si no succeeix cap problema es passa a la pantalla de Resposta.

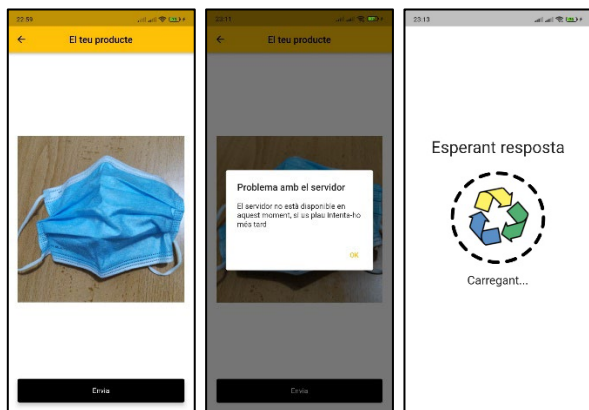


Fig. 12. Pantalla imatge, animació de càrrega i missatge d'error

## 8.4 Pantalla resposta

Es tracta de dues pantalles que es mostren depenent de la resposta del servidor i si la puntuació "score" (precisió del resultat) és com a mínim de 0,6.

### Producte trobat

En la següent figura podem observar la primera pantalla d'èxit. Aquesta es presenta amb les diferents variacions depenent del valor de la variable "bin". El botó de sota retorna a la pantalla principal.

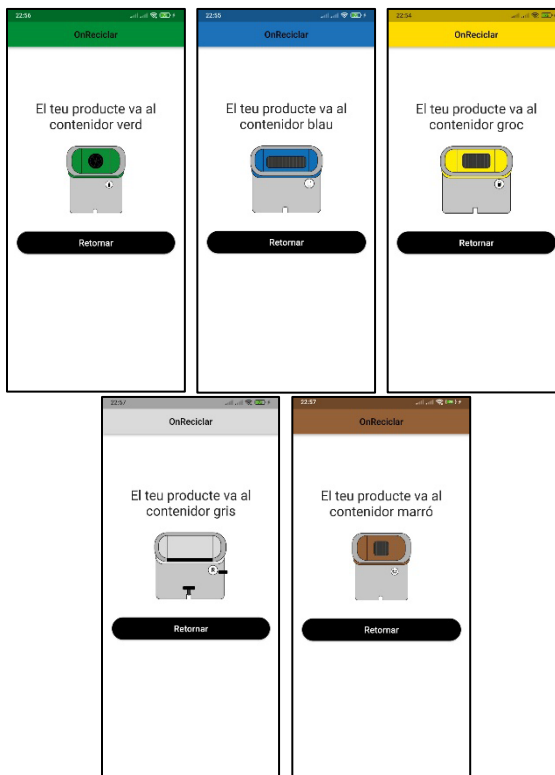


Fig. 13. Variacions pantalla èxit

### Producte no trobat

Si la puntuació ha sigut menor que 0.6 llavors es mostra la següent pantalla (veure Apèndix C).

En aquest cas s'ofereix a l'usuari l'oportunitat de contribuir a ampliar la mostra del servidor per tal de poder reconèixer el seu producte en un futur. En el cas que l'usuari accepti es passa a la següent pantalla: Pantalla Tutorial, en cas contrari retorna a la pantalla principal.

### 8.5 Pantalla tutorial

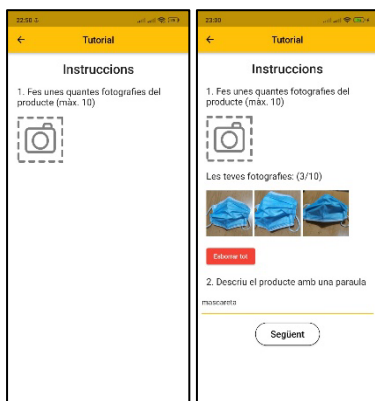


Fig. 14. Pantalla tutorial

En aquesta pantalla (veure Fig. 14) s'indica a l'usuari les instruccions que ha de seguir per tal de poder fer el procés de classificació del seu producte. Primerament, el client haurà de fer diverses fotografies del producte des de diferents angles (situació 2 de pantalla Càmera, veure Fig. 9). A continuació l'aplicació mostra aquests fitxers i dona l'opció d'esborrar la selecció per tal de refer les fotografies. Un cop obtenim la llista de fotos es demana mitjançant un formulari la inserció d'una petita descripció del producte.

Abans de passar al següent pas del tutorial (i següent pantalla) es verifica l'input de l'usuari amb una sèrie de condicions com el nombre màxim i mínim de caràcters, a més per darrere es neteja per eliminar números i caràcters especials del text.

### 8.6 Pantalla dibuixar sobre imatge

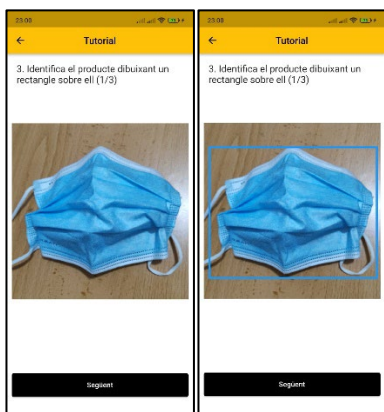


Fig. 15. Pantalla Dibuixar sobre imatge

En aquesta pantalla (veure Fig. 15) es mostren les imatges de la llista carregada anteriorment i permet dibuixar un rectangle arrossegant el dit per pantalla.

Per aconseguir aquesta funcionalitat es fa ús d'una

llibreria de Flutter anomenat "image\_painter", el qual s'ha copiat el codi base per tal de poder desenvolupar una versió personalitzada que s'adapti a les nostres necessitats. En aquesta versió s'han eliminat totes les funcionalitats extra excepte la de dibuixar el rectangle. A més, ha sigut configurat manualment per eliminar el rectangle si es fa l'acció de dibuixar el rectangle novament (lliscant el dit sobre la pantalla).

### Configuració i connexió amb la base de dades

Per tal de poder emmagatzemar les fotografies en una base de dades que permeti aquest tipus de fitxer i per treballar de forma àgil s'ha escollit configurar l'aplicació per treballar amb Cloud Storage de Google Cloud. Aquest servei ens proporciona una integració segura amb Flutter i un ràpid funcionament dels processos. A continuació es mostra la jerarquia de directoris i fitxers de la base de dades.

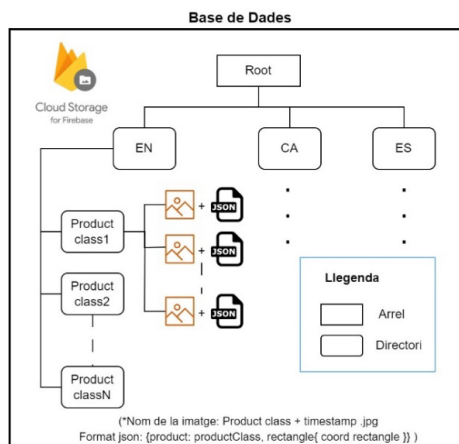


Fig. 16. Jerarquia de directoris i fitxers Cloud Storage

Com es pot observar a la Fig. 16, l'arrel té tres directoris fills [EN,CA,ES], aquests s'emplenaran amb directoris de productes depenent de l'idioma del dispositiu (requisit internacionalització). El "Product class" es tracta de la descripció que l'usuari va fer en el pas anterior del tutorial, per tant, cada cop que l'usuari dibuixa un rectangle i prem el botó de "següent" per darrere es crearà un directori amb el nom del producte i s'aniran pujant les imatges sobre aquest.

A més, per cada imatge s'associarà un fitxer JSON amb el mateix nom que la imatge. Aquest fitxer conté informació valuosa per la xarxa neuronal, ja que descriu la classe del producte i les coordenades per formar el rectangle en la fotografia (coordenada del punt central del rectangle, alçada i amplada). Amb aquests dos arxius, el servidor és capaç d'ampliar la mostra d'entrenament de YOLO.

## 8.7 Pantalla final

A l'Apèndix D es pot trobar una captura de la pantalla final, aquesta mostra simplement un missatge d'agraïment per ajudar a ampliar la mostra després de realitzar el procés de classificació. Està composta d'una imatge estàtica i un botó per retornar a la pantalla inicial.

## 9 CONCLUSIONS

Tot i amb les complicacions amb les dates de les iteracions, es pot concloure que el projecte s'ha acabat amb èxit i complint les expectatives tant de funcionalitats requerides com de terminis de finalització i entrega dels resultats.

Sobre el projecte en si, de manera genèrica i en comparativa amb altres projectes de la mateixa menció, no es contempen aspectes altament tècnics amb un desenvolupament i arquitectura específica sinó que es tracta d'un projecte de desenvolupament d'un front-end orientat a facilitar a l'usuari la interacció amb el back-end. Així i tot, durant el procés de desenvolupament del mateix vaig poder aprendre i comprendre la visió del desenvolupador front-end i la importància de la seva tasca en qualsevol projecte de software.

Després de fer els tests d'acceptació a persones alienes al projecte es va poder descobrir les qualitats del producte, ja que el van trobar simple, útil, fàcil d'utilitzar, accessible per a qualsevol edat i que va directe al gra pel que fa a la seva finalitat.

Finalment, un aspecte important que va impulsar l'interès a completar amb èxit la tasca va ser la temàtica i el missatge que vol transmetre la finalitat del projecte que és el reciclatge. Actualment, sembla que aquest tema ja està conscienciat en la majoria de la societat, però mentre encara quedin dubtes en la gent sobre on llençar un producte, els enginyers hauran de treballar per resoldre'ls donant una solució accessible i òptima.

## AGRAÏMENTS

Primerament, voldria agrair a la meva família per tot el suport que em van donar al llarg del grau i amb aquest projecte. Finalment, voldria agrair al meu tutor del treball pel continu suport durant les reunions donant-me *feedback* constant i responent a les meves qüestions.

## BIBLIOGRAFIA

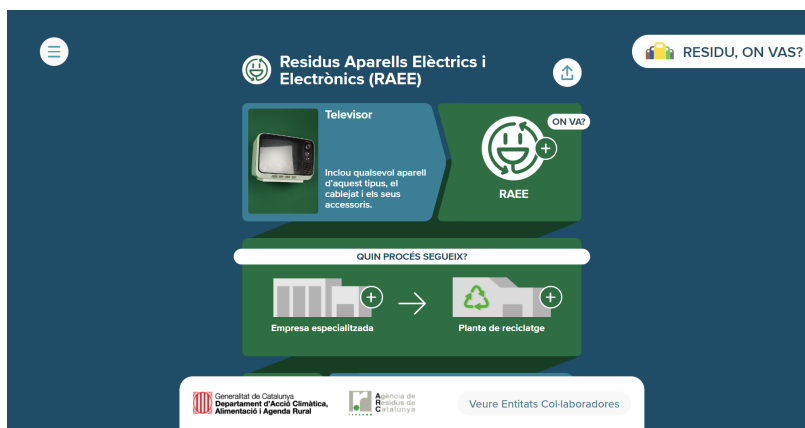
- [1] Campanya "Envàs, on vas?" [https://residus.gen.cat/ca/ambits\\_dactuacio/sensibilitzacio/campanyes/campanya\\_on\\_vas\\_envas/](https://residus.gen.cat/ca/ambits_dactuacio/sensibilitzacio/campanyes/campanya_on_vas_envas/)
- [2] Redacció EFEverde. "4 de cada 10 españoles dicen no reciclar por miedo a equivocarse". EFEverde.

<https://efeverde.com/espanoles-reciclaje-miedo-equivocarse/> (accedit el 21 d'Abril de 2023).

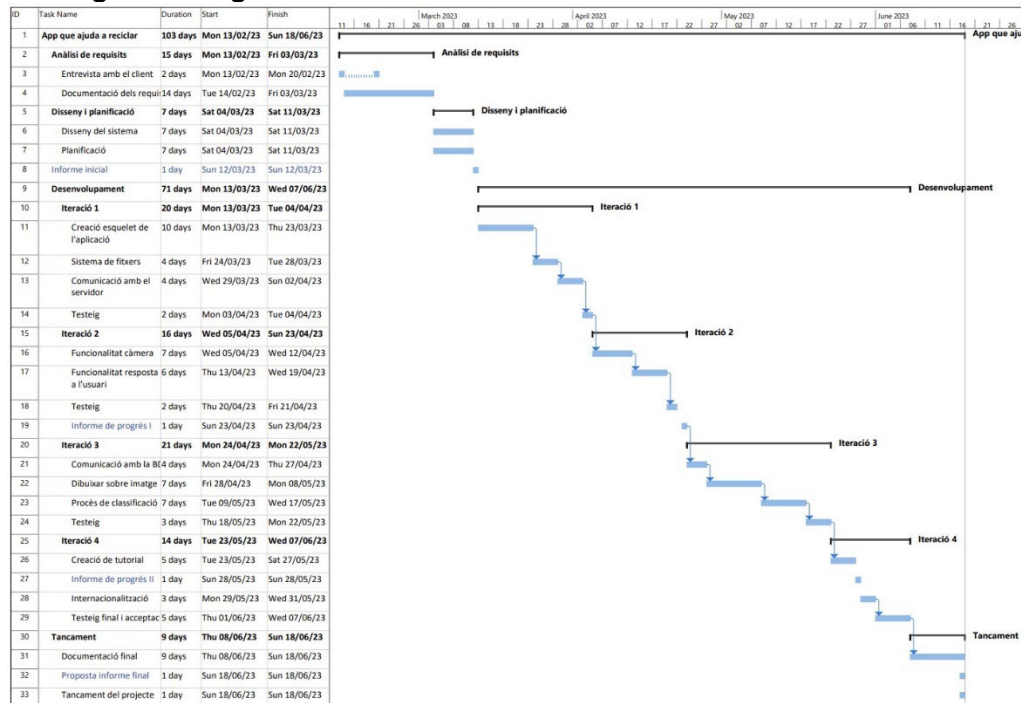
- [3] Aplicació "Residu on vas" <https://www.residuonvas.cat/ca>
- [4] Aplicació RECICLOS <https://www.reciclos.com/>
- [5] Aplicació Aire <https://play.google.com/store/apps/details?id=com.ecoembes.aire&hl=es&gl=ES>
- [6] CLIP: Connecting text and images. (s.f.). OpenAI. <https://openai.com/research/clip>
- [7] Ankit Narayan Singh (25 d'Abril, 2020), Image Recognition and Object Detection in Retail, <https://ankitsingh.medium.com/image-recognition-and-object-detection-in-retail-c690dca48e65>
- [8] Nelson, J. (18 de Juliol, 2020). Retail Store Item Detection using YOLOv5. Roboflow Blog. <https://blog.roboflow.com/retail-store-item-detection-using-yolov5/>
- [9] C. Simões. "MoSCoW. ¿Qué es y cómo priorizar en el desarrollo de tu aplicación?" Blog ITDO - Agencia de desarrollo Web, APPs y Marketing en Barcelona. <https://www.itdo.com/blog/moscow-que-es-y-como-priorizar-en-el-desarrollo-de-tu-aplicacion/> (accedit el 23 d'Abril de 2023).
- [10] API Supervisely <https://api.docs.supervise.ly/>
- [11] YOLO: Real-Time Object Detection <https://pjreddie.com/darknet/yolo/>
- [12] Juan Du, "Understanding of Object Detection Based on CNN Family and YOLO" 2018 J. Phys.: Conf. Ser. 1004 012029 [Online]. Disponible a <https://iopscience.iop.org/article/10.1088/1742-6596/1004/1/012029/pdf>
- [13] Red neuronal convolucional - Wikipedia, la enciclopedia libre [https://es.wikipedia.org/wiki/Red\\_neuronal\\_convolucional](https://es.wikipedia.org/wiki/Red_neuronal_convolucional)
- [14] Object Detection in 2023: The Definitive Guide - viso.ai, Sección "Most Popular Object Detection Algorithms", <https://viso.ai/deep-learning/object-detection/>
- [15] How single-shot detector (SSD) works? | ArcGIS API for Python, web: ArcGIS Developers, <https://developers.arcgis.com/python/guide/how-ssd-works/>
- [16] YOLOv3 to TensorFlow Lite Conversion <https://medium.com/tfug-mumbai-weekly/yolov3-to-tensorflow-lite-conversion-4602cec5c239>
- [17] Flutter Vs Android Studio: What's the Difference? (s.f.). InterviewBit. <https://www.interviewbit.com/blog/flutter-vs-android-studio/>
- [18] "Android Studio - Wikipedia". Wikipedia, the free encyclopedia. [https://en.wikipedia.org/wiki/Android\\_Studio](https://en.wikipedia.org/wiki/Android_Studio)
- [19] Editor. (2016, 28 de gener). The Good and The Bad of Xamarin Mobile Development. AltexSoft. <https://www.altexsoft.com/blog/mobile/pros-and-cons-of-xamarin-vs-native/>

# APÈNDIX

## A. Aplicació web residuonas.cat



## B. Diagrama de gantt



### C. PANTALLA PRODUCTE NO TROBAT



### D. PANTALLA FINAL

