

---

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

Aguilar Martínez, Víctor; Sánchez Albaladejo, Gemma, dir. Estudi de viabilitat d'una app mòbil per pujar exàmens a Caronte amb validació de l'estudiant. 2021. (958 Enginyeria Informàtica)

---

This version is available at <https://ddd.uab.cat/record/248456>

under the terms of the  license

# Estudi de viabilitat d'una app mòbil per pujar exàmens a Caronte amb validació de l'estudiant

Víctor Aguilar Martínez

**Resum**—La tecnologia que et permet treballar i estudiar online ha avançat molt els últims 10 anys. Això i la situació de pandèmia en la qual vivim n'ha fet evolucionar encara més en aquesta direcció. Per aquest motiu s'ha pensat fer una aplicació mòbil per pujar els exàmens un cop finalitzats, a Caronte, amb validació de l'estudiant mitjançant reconeixement dactilar i autorització del professor encarregat de l'aula. Durant aquest treball s'explica les decisions preses en el disseny de l'aplicació i l'extensió Moodle necessàries, la implementació d'un primer prototip i, un estudi de viabilitat perquè en un futur no es parteix de zero en la implementació sobre un OCR i un centrat d'imatges de les fotografies que fa l'aplicació.

**Paraules clau**—Caronte, Aplicació mòbil, Android, Reconeixement dactilar, OCR, Exàmens

**Abstract**—The technology that allows you to work and study online has come a long way in the last 10 years. This and the pandemic situation in which we live has made it evolve even more in this direction. For this reason it has been thought to make a mobile application to upload the exams once completed, in Caron, with validation of the student through fingerprint recognition and authorization of the teacher in charge of the classroom. This work explains the decisions made in the design of the application and Moodle extension, the implementation of a first prototype and a feasibility study so that in the future we do not start from scratch in the implementation of an OCR and an image centering of the photographs made by the application.

**Index Terms**—Caronte, mobile app, Android, Fingerprint recognition, OCR, Exams

---

## 1. Introducció

Aquest és un projecte ambiciós que barreja diferents àmbits dins de l'enginyeria informàtica motivat per la falta d'opcions a l'hora de lliurar exàmens, també de forma digital. La pandèmia ens ha fet que hàgem de buscar formes alternatives per fer les coses, per exemple els lliuraments dels exàmens. Per això, s'ha plantejat fer un estudi de viabilitat d'una aplicació mòbil que pugui connectar-se a Caronte i poder fer fotografies dels exàmens i penjar-los al Moodle. També s'ha pensat a implementar una eina per poder corregir els exàmens i que aquesta correcció la puguin veure els estudiants en publicar les notes. Aquesta extensió de Caronte és una activitat nova de Moodle per recollir i gestionar els examen.

En aquest projecte hi ha dues parts diferenciades, la primera és l'aplicació mòbil que té com a objectius utilitzar dades biomètriques (empremta dactilar) per verificar la identitat de l'usuari i, fer les fotografies per poder enviar-les cap a la segona part del projecte. Aquesta segona part és una extensió de Moodle que serveix per gestionar les dades que rebrà de l'aplicació i perquè el professor pugui validar l'entrega de l'examen, corregir-lo, posar notes als diferents exercicis i compartir amb

l'estudiant aquestes correccions per generar un full Excel amb el recull de les notes de tota la classe.

Aquest projecte està dividit en dos TFG. El primer s'encarrega de la part que té relació amb la menció d'Enginyeria del Software, que és:

- Aplicació mòbil.
- Enviament de dades entre l'aplicació i Caronte
- Exportació de les notes.
- Eina de correcció

El segon TFG s'encarrega de la part que té relació amb la menció de Computació, que és:

- Extensió de Caronte
- Procés de validació
- Fotografies a PDF.
- Tria del OCR

En l'apartat 2 es veurà els objectius, el 3 es presentarà l'estat de l'art, el 4 es presentarà la planificació, el 5 la metodologia, el 6, 7 i 8 el desenvolupament i, per últim tindrem un apartat de presentació de resultats, de conclusions i la bibliografia.

## 2. Objectius

Els objectius del projecte són:

1. Crear una nova extensió per a Caronte perquè l'aplicació es pugui comunicar amb una aplicació externa.

El primer objectiu consisteix a crear una modificació (plugin) [1] personalitzada que mitjançant "Learning Tools Interoperability" (LTI) [2], un format per enviar dades entre aplicacions, es comuniqui amb l'aplicació externa per autenticar.

2. Crear una aplicació mòbil de validació i pujada d'exàmens.

El segon objectiu és dissenyar i crear l'aplicació que permet validar a l'estudiant i que pugui pujar les fotografies de l'examen que s'enviarà a Caronte. L'aplicació s'implementarà en les últimes versions d'Android (10.0 i 11.0) donat que la llibreria que s'utilitzarà (Android Biometric)[3] que necessitem per comprovar l'empremta només està a partir d'aquestes versions d'Android.

Per al segon objectiu s'han pensat diverses opcions per assolir l'objectiu. La primera és que des del compte del professor es generi un codi que l'estudiant haurà d'introduir a l'hora de pujar l'examen. La segona opció seria que quan l'estudiant vol entregar l'examen s'envii una notificació al professor i aquest acceptaria a l'alumne. L'opció que s'implementarà serà la segona perquè dóna control al professor de qui entrega a l'examen.

3. Crear els processos de validació.

El tercer objectiu és el procés de validació de l'estudiant i està desglossat en dos subobjectius:

- a. Crear el procés de validació per identificar a l'estudiant.

El primer subobjectiu és a la part de l'estudiant. És desenvolupar una aplicació mòbil externa que identificarà l'estudiant per les seves dades biomètriques (empremta dactilar).

- b. Crear una vista perquè els professors puguin acceptar el lliurament de l'examen.

El segon subobjectiu és a la part dels professors, quan Caronte rebi per part de l'aplicació una resposta afirmativa, el professor pot entrar a la vista de validació de l'activitat on podrà acceptar que s'entregui l'examen o en cas contrari li denegui el lliurament de l'examen.

4. Crear el procés per enviar les fotografies.

Seguint amb l'aplicació externa, el quart objectiu es crear el procés per enviar les fotografies des de l'aplicació a Caronte. Aquest objectiu està dividit en dos subobjectius.

- a. Poder fer les fotografies a l'aplicació.

El primer és que l'aplicació deixi fer fotografies de l'examen a l'estudiant desde l'aplicació.

- b. Transformar-les en format PDF.

El segon seria agafar les fotografies que l'estudiant a fet i centrar-les. Un cop centrades, juntar-les en un mateix arxiu PDF.

5. Crear una eina que pugui utilitzar el professor per a corregir l'examen.

El cinquè objectiu és crear una eina d'edició amb l'objectiu que el professor pugui corregir els exàmens des de Caronte sense necessitat de descarregar l'examen.

6. Crear el procés per a que Caronte pugui llegir automàticament del PDF les notes de les preguntes, i donar la nota final de l'examen.

Un cop l'examen està corregit i amb les notes escrites a mà pel professor, utilitzarem un algorisme de Reconeixement Òptic de Caràcters (OCR) per extreure de forma automàtica aquestes notes. Les notes es guardaran associades a

l'estudiant. Tindrem guardades una nota per cada exercici. També calcularem la nota final de l'examen.

7. Crear el procés d'exportació de notes en format Excel.

L'últim objectiu és crear una opció que li permet al professor exportar un arxiu tipus Excel on hi estiguin totes les notes dels estudiants d'aquell examen.

#### Priorització dels Objectius

Objectiu	Prioritat
1. Extensió Caronte	Crítica
2. Aplicació mòbil	
3.a. Validació estudiant	Alta
3.b. Validació professor	Baixa
4.a. Enviament de les dades	Crítica
4.b. Fotografies en PDF	Mitja
5. Eina de correcció d'exàmens	Baixa
6. Tria OCR	
7. Exportació notes	Mitja

### 3. Estat de l'art

A nivell de d'extensions d'exàmens a Caronte no hi ha res similar, ja que, hi ha extensions que permeten fer exàmens online i, també hi ha extensions per entregar treballs que també es poden utilitzar per entregar exàmens, però sense les funcionalitats que es faran.

A l'hora de plantejar el projecte es va pensar en diferents opcions per implementar un OCR. Fer un OCR propi seria molt costós a nivell de temps i esforç i podria ser un TFG en si mateix, per això s'ha optat per implementar un OCR extern. La millor opció seria utilitzar l'algoritme que proporciona l'empresa Google però aquest és de pagament, per això s'ha agafat un OCR d'una empresa que té una versió gratuïta de l'empresa OCRSpace [6].

Per al centrat d'imatges hi ha la llibreria OpenCV[4], que té una sèrie de funcions que implementen tècniques de visió per computadors que donen molt bons resultats.

Per crear l'eina d'edició per corregir els exàmens des de Caronte hi ha una extensió de Moodle "unoconv"[5]. Aquesta extensió ens permetrà fer conversions de format.

Per desenvolupar la validació en l'aplicació hi ha varies opcions, però s'ha escollit la llibreria d'Android Biometric Manager per comprovar la disponibilitat de la autenticació biomètrica del dispositiu i, Biometric Prompt per mostrar el diàleg i gestionar les credencials.

### 4. Planificació

S'ha dividit la planificació del projecte en cinc "sprints" d'unes tres o dos setmanes i un últim d'una setmana que seria la setmana final abans de la entrega del projecte. Cada "sprint" tindrà els seus objectius.

- Sprint 1: 15 de març – 4 abril
  - Extensió Caronte
  - Aplicació mòbil (no funcional)
- Sprint 2: 5 abril – 25 abril
  - Procés de validació
  - Procés d'enviament de dades
- Sprint 3: 26 abril – 16 maig
  - Eina de correcció d'exàmens
  - Implementació del OCR
- Sprint 4: 17 maig – 30 maig
  - Implementació de la exportació notes examen
  - Ajuntar mòduls
- Sprint 5: 31 maig – 20 juny
  - Ajuntar mòduls
  - Correcció d'errors
- Sprint 6: 21 juny – 28 juny
  - Informe final
  - Presentació

### 5. Metodologia

En aquest apartat s'explicaran les metodologies que es segueixen durant el desenvolupament d'aquest projecte. La primera metodologia es Scrum [7], aquesta ens permet dividir el desenvolupament del projecte en diversos sprints. Cada sprint consta de diverses fases i son independents entre elles, això en permet a l'inici de cada sprint replanificar en cas de que sigui necessari.

Com s'ha dit anteriorment aquest projecte està dividit en dos TFGs, per això, la segona metodologia es Kanban [8], aquesta eina ens permetrà dividir i visualitzar millor el flux de treball que portarem a terme.

## 6. Desenvolupament

En aquest apartat s'explicarà el desenvolupament del projecte durant tot el semestre.

### 6.1. Extensió Caronte

Respecte al primer objectiu, hem modificat la manera en la qual volíem enviar dades entre aplicació i Caronte. Com l'aplicació utilitza el servidor Apache on està allotjat Moodle no podem utilitzar LTI, que serveix per connectar-se a un servidor extern, per tant, utilitzarem connexions HTTP [10].

Per assolir el primer objectiu s'ha descarregat l'última versió estable de Moodle disponible a l'inici del projecte (Moodle 3.10.1). Per poder executar la plataforma Moodle en local s'utilitza el programa XAMPP, amb la seva versió més compatible amb la plataforma Moodle escollida, que és la versió 7.4.15 de PHP. Per a la gestió de la Base de Dades s'utilitza el programa phpmyadmin que ens proporciona XAMPP.

#### 6.1.1. Base de l'extensió

Hem creat una extensió de moodle des de zero i l'hem anomenat "Examen". Aquest mòdul ha de contenir uns arxius obligatoris i seguir la jerarquia d'arxius que deixa moodle a la seva documentació [11]. La jerarquia d'arxius és la següent:

- **"db"**: En aquest directori es guarden els arxius relacionats amb la base de dades.
- **"lang"**: En aquest directori guarda tots els arxius de llenguatge, però el directori obligatori és "en" (english) però, hem afegit també en espanyol i en català.
- **"pix"**: En aquest directori es guarden les imatges que s'utilitzen com a icona del mòdul.
- **"lib.php"**: En aquest arxiu estan les funcions que cridarà moodle per afegir, modificar i eliminar l'activitat del mòdul.
- **"mod\_form.php"**: En aquest arxiu es defineix el formulari que el professor omplirà quan afegeix o modifiqui l'activitat.
- **"index.php"**: Moodle utilitza aquest arxiu per escoltar totes les instàncies dels mòduls.
- **"view.php"**: És la vista que s'executarà quan l'usuari entri a l'activitat.

- **"version.php"**: En aquest arxiu s'especifiquen la versió del nostre mòdul.

### 6.1.2. Estudi de viabilitat d'utilitzar un OCR per extreure les notes del examen

Volem estudiar la viabilitat d'un mòdul que agafi les notes escrites a mà de l'arxiu de l'examen corregit i que extregui els números de les notes de l'examen i li passi al servidor perquè aquest assigni les notes a l'estudiant. Aquest apartat explica les diverses opcions plantejades amb l'anàlisi dels resultats i quina és l'opció escollida.

La primera opció és utilitzar un OCR ja desenvolupat com pot ser el de Google. Vàrem analitzar 2 opcions, Google i el de l'empresa OCR Space. Google és de pagament i volíem un que fos gratuït i OCR Space no funciona amb reconeixement manuscrit. Donat que l'única alternativa trobada era de pagament es va descartar aquesta opció.

La segona opció seria utilitzar una xarxa neuronal convolucional[18] per a detectar els caràcters manuscrits, ja que, el reconeixement de caràcters manuscrits amb aquest tipus de xarxes és un problema que està resolt i amb molt bons resultats. Per aquesta opció es muntaria un web service amb el framework Django, aquest és un framework especialitzat en web services de Python. Aquest web service estaria escoltant per peticions HTTP i, un cop rebuda una petició utilitza una xarxa neuronal per reconèixer el caràcter i retornar la resposta.

Tenim dues alternatives per obtenir el resultat del reconeixement, la primera és crear una xarxa neuronal convolucional pròpia amb les llibreries de Python TensorFlow[21] i Keras[22], això vol dir que haurem de crear el model de la xarxa neuronal, entrenar-lo i guardar aquest model entrenat per poder predir les imatges en un futur. La xarxa seria entrenada amb el dataset MNIST[19], aquest dataset és que s'utilitza per entrenar a les xarxes neuronals que es volen dedicar a reconeixement de dígit manuscrits.

Ja que el més costós de les xarxes neuronals és l'entrenament de les mateixes la segona opció seria descarregar-nos d'internet una xarxa preentrenada, això vol dir que no hauríem de crear el model i el pas d'entrenar-la no l'hauríem de fer.

Respecte a la primera alternativa hem creat una xarxa neuronal convolucional de tres capes

convolucionals i tres capes "full connected", el mètode d'activació és "relu", amb un filtre de dimensions 3x3 i una capa "max\_pooling" de dimensions de 2x2.

El següent que es va fer va ser crear un dataset d'imatges de números escrits amb un programa d'escriptura digital com pot ser el programa Paint. El dataset consta d'imatges del 0 al 9 (quatre imatges per número) amb el fons en blanc i els números de color vermell (veure annex 2).

Vam fer proves sobre el dataset i el resultat que ens ha donat accuracy sobre el nostre dataset ha sigut d'un 27.5%, aquest resultat és molt poc, ja que ha encertat una mica més de la quarta part del total de les imatges ha predit. Com s'ha dit al principi d'aquesta opció aquest problema està resolt amb molt bons resultats, per això, hem buscat els motius pels quals no hem obtingut bons resultats.

Els motius poden ser els colors del fons i el número, la forma del número o la quantitat d'imatges disponibles per predir. Per això crearem més datasets per comprovar l'efectivitat de la nostra xarxa. Un motiu de molt pes que afecta directament a les prediccions és l'entrenament, i el fet d'haver entrenat la xarxa amb el dataset MNIST que és bastant diferent del nostre podria ser un factor de pes per al qual surten mals resultats.

S'ha creat un datasets més a part del creat l'original, són els números blancs sobre el fons negre(veure annex 3). Després d'executar les prediccions els resultats sobre el nostre dataset ha sigut d'un 60% accuracy, encara que el resultat és significativament millor, però, encara no arriba l'accuracy esperat que seria aproximadament el 90-95% perquè el professor ni hagués de corregir massa errors.

S'ha decidit tornar a crear un últim dataset, aquest serien números negres amb fons blanc (veure annex 4), per aquest dataset s'han escrit els números en diferents tipografies per si la forma afecta els resultats i hem augmentat a 6 imatges per número. Un cop executades les prediccions el resultat sobre el dataset ha sigut de 28,33%, el resultat millora molt poc el primer i és molt menor al segon.

Com els resultats de la primera alternativa no eren els esperats s'han provat els datasets amb una xarxa preentrenada per comprovar la diferència si el model que hem creat nosaltres.

La xarxa neuronal conté una primera capa de padding, 5 capes convolucional amb els filtres de

dimensions de 3x3 i cada capa convolucional té la seva capa de MaxPool de dimensions de 2x2 amb el mètode d'activació relu, per últim les tres capes "full connected".

Hem executat aquesta xarxa amb els nostres datasets, però cap han passat del 10% d'accuracy. Aquests resultats estan bastant lluny de l'esperat i que ens fa descartar aquesta opció i posar les qualificacions manualment.

Com a conclusions d'aquesta anàlisi podem extreure que encara el segon dataset és el millor, no arriba al resultat esperat. Sabem com funcionen les xarxes neuronals on els entrenaments són claus, podem dir que és el factor determinant, ja que com he dit abans el dataset amb el que s'ha entrenat dista molt del que s'ha avaluat. Això ens porta al fet que necessitaríem una gran quantitat d'imatges per nodrir a la nostra xarxa, per solucionar-ho és podria incorporar les imatges que fan els professors per augmentar el nostre dataset en dues formes: el primer seria en quantitat i el segon en qualitat, ja que la xarxa podria detectar més formes.

Ja que crear una xarxa neuronal per identificar el número de les notes podria ser en si un TFG, ja que, crear una xarxa neuronal i entrenar-la és complex i es poden provar diferent tipus de xarxes i poder integrar-ho amb el servidor per rebre les imatges i enviar els resultats.

### 6.1.3. Validació del professor

En aquest apartat s'explica la validació per part del professor. La validació del professor està a la banda del Caronte.

Si és entra al Caronte amb un compte de professor i accedeix a un examen, s'entra a una vista on es mostra el resum d'aquell examen. El resum mostra el número de participants, el número d'exàmens entregats i el número d'exàmens validats i un botó per anar a la vista de validació de l'estudiant.

La vista de la validació com es veu a la figura 1 consta d'una llista de tots els alumnes de l'assignatura i de cada alumne es mostra el nom de l'usuari (NIU), el nom de l'alumne, la data d'entrega de l'examen, l'estatus de l'entrega (no entregat, entregat, validat, denegat), l'examen de l'alumne, el botó per validar i denegar l'entrega i per últim les notes de cada pregunta, el número d'inputs que hi hagi en aquesta vista dependrà del número que se li hagués especificat quan es va crear la tasca.

Usuario	Nombre	Entregado el	Estatus	Validar/Denegar	Notas
1234567	Victor Aguilera	27 de mayo de 2021, 15:48	Validado	Validar Denegar	2   1.5   0.75   1   0.5   Guardar
7654321	Joen Sanchez	27 de mayo de 2021, 15:48	Validado	Validar Denegar	2   1.75   1.5   2   2   Guardar

Publicar notas  
Exportar notas

Fig. 1 - Vista de la página de validación del profesor.

Quan es clica a el botó de validació i denegació es crida a un fitxer JavaScript que agafa l'id del examen i l'id de l'alumne i fa una petició Ajax a un fitxer PHP del servidor que canvia l'estat d'aquell usuari per a aquell examen a validat o denegat.

Per aquells estudiants que hagin sigut validats es mostraran els inputs per posar la nota de cada exercici d'aquell estudiant i al final un botó per guardar les notes. Aquest botó crida a un PHP del servidor mitjançant un formulari HTML. Les notes es guardaran amb el format: "x<sub>1</sub>-x<sub>2</sub>-x<sub>3</sub>-x<sub>4</sub>-x<sub>5</sub>" en format string i on x és la nota de l'exercici, també es guarda la nota final de l'examen en format float.

Per últim, hi ha un botó perquè el professor pugui publicar les notes. Quan es clica el botó es crida a un JavaScript on es fa una petició Ajax que crida un PHP del servidor on canvia l'estat de l'examen a públic, un cop públic els alumnes poden veure la seva nota de l'examen.

### 6.1.4. Exportació de notes d'una mateixa classe

En aquest apartat s'explica com funciona l'exportació de les notes d'una classe per aquell examen.

Al clicar al botó que diu "Exportar notas" dins de la vista de validació de l'estudiant és fa una consulta per obtenir tots els estudiants d'aquell examen a la taula de la base de dades on es guarden les notes i genera un fitxer CSV[23] delimitat per comes amb el següent format: NIU, Nom, Ex1, .. ExN, Nota.

## 6.2. Aplicació mòbil

Per al segon objectiu hem creat una aplicació Android des de zero amb el llenguatge Java. El flux de l'aplicació és veu a l'annex 1. Per més informació veure el TFG anàleg.

### 6.2.1. Validació de l'estudiant

Un cop tenim la base del projecte s'ha treballat la implementació del reconeixement dactilar i les connexions HTTP. Respecte al reconeixement

dactilar s'ha implementat amb les llibreries BiometricPrompt i BiometricManager que ens proporciona Android.

Un cop l'usuari entri a l'activitat d'Autenticació, internament es comprova que el dispositiu tingui el hardware necessari per continuar, i si el té, registra les opcions perquè pugui començar el reconeixement quan l'usuari faci clic al botó de l'activitat. Després de fer el registre del dispositiu es crea una instància de BiometricPrompt, a aquesta instància se li ha de passar l'activitat on s'executa, un executor i per últim les funcions que pot executar al finalitzar el reconeixement.

L'activitat que li passarem serà Autenticació, ja que és on estem en el flux d'execució, una instància de la llibreria Executor inicialitzada amb el context de l'activitat on estem en el flux, i per últim, tres funcions de finalització:

- "onAuthenticationError": Aquesta funció imprimeix per pantalla l'error que ha pogut donar i torna a l'usuari a l'activitat dels Exàmens.
- "onAuthenticationSucceeded": Aquesta funció imprimeix per pantalla que has tingut èxit i envia a l'usuari a l'activitat de la càmera.
- "onAuthenticationFailed": Aquesta funció imprimeix per pantalla que ha fallat l'autenticació i deixa a l'usuari intentar-ho fins al nombre màxim d'intents, si se supera el nombre d'intents s'executarà "onAuthenticationError".

Per últim es crearà la informació que se li mostri a l'usuari quan se li obri la finestra del reconeixement.

Un executor és una forma de crear i executar threads de forma implícita quan es cridi a la funció a la qual li passi l'executor, és a dir, que quan s'executi el biometricPrompt es crearà un thread i aquest executarà el biometricPrompt sense que el desenvolupador li digui quan ha de començar.

Aquestes llibreries són molt útils perquè les dades biomètriques són úniques per a cada persona, és a dir, que aquestes dades són informacions sensibles. Per aquest motiu Android no deixa enviar ni fer captures de pantalla que puguin ser una fallada en la seguretat del sistema. Això és un problema per al nostre projecte perquè des del servidor hauríem de verificar que l'usuari que vol fer l'entrega de l'examen estigui registrat a Caronte sigui realment aquella persona. La solució que hem trobat és implementar CRAM (Challenge Response Authentication Mechanism) [12].

CRAM és un algoritme de verificació basat en un repte, el client demana fer una acció al servidor i aquest li envia un repte, per exemple Captcha [13], i si l'usuari respon correctament al repte li dona permís per fer l'acció.

Aquest mètode s'hauria d'adaptar a la llibreria d'Android pel motiu que hem esmentat abans. Hi hauria dues parts, el registre i l'autenticació.

- Quan l'usuari entri per primera vegada a l'aplicació, el client genera un parell de claus, una clau pública i una clau privada i s'envia al servidor perquè la guardi. Aquesta clau pública li servirà per comprovar que té la clau privada.
- A l'hora de l'autenticació el client li demana un repte, el servidor li retornarà un string que el client haurà de signar amb la seva clau privada. Un cop signat el repte se li envia al servidor perquè comprovi que sigui ell.

## 6.2.2. Procés de comunicació entre l'aplicació i el servidor

En aquest apartat s'explicarà el procés que se segueix l'aplicació per enviar i rebre dades al servidor. Hi ha dos tipus de connexió de l'aplicació: HTTP i FTP.

L'aplicació utilitza el protocol HTTP per enviar dades que el servidor necessita per resoldre una consulta a base de dades com pot ser l'inici de sessió o la comprovació de l'empremta dactilar.

La classe que fa les connexions HTTP li arriba per paràmetres un objecte de la classe Map [14], l'objecte Map ens permet passar la informació en forma de diccionari (Clau - Valor). La complexitat [15] d'aquesta classe en els millors dels casos amb un bon hash és de  $O(1)$ , mentre que en el pitjor dels casos és de  $O(n)$ . Com nosaltres utilitzem la classe Map per passar tota la informació necessària, la recorrem tota en un for, per tant, en el nostre cas la complexitat serà d' $O(n)$ .

Aquesta classe necessita l'URL [16] del servidor amb format ("url", string amb la url), l'activitat on estigui l'execució amb format ("activity", string amb activitat actual) i per últim tots els paràmetres amb format (nom del paràmetre, valor del paràmetre convertit en string).

Per enviar els exàmens en PDF s'utilitza el protocol FTP.

Primer s'envia al servidor tota la informació necessària perquè Caronte trobi el fitxer, aquesta informació és l'id de l'usuari, l'id del curs, l'id de l'examen, data d'entrega, etc. Després d'aquest pas s'inicia el protocol FTP[20] per enviar el fitxer del mòbil a Caronte.

Per aprofundir en aquest apartat cal llegir el TFG anàleg a aquest.

## 6.2.3. Transformació d'imatges a un document PDF

En aquest apartat s'explica com és la transformació de totes les imatges fetes per l'estudiant en un mateix document PDF.

Des de l'aplicació mòbil i un cop finalitzat el procés de les fotografies es crida a un mètode que fa la transformació.

L'algorisme crea un document PDF. Recorre l'array d'imatges i per cada imatge crea una pàgina de PDF i afegeix la imatge a la pàgina i aquesta al PDF.

Un cop afegides totes les pàgines s'obre un stream per guardar el fitxer al mòbil per posteriorment enviar-ho al servidor.

## 6.2.4. Estudi de viabilitat per centrar les imatges

S'ha estudiat la possibilitat que les imatges se centrin perquè es puguin llegir bé.

La primera opció és fer l'algorisme que centra les imatges des de la pròpia aplicació, però es va descartar perquè afegir la llibreria OpenCV augmenta el tamany de l'aplicació, la falta de comunitat a l'hora de resoldre els problemes i principalment perquè aquest problema en Python és molt més fàcil d'implementar.

La segona opció és fer-ho en Python amb un servidor al qual es puguin enviar les imatges i les retorni centrades. Per centrar la imatge primer s'ha de fer un preprocessament a la imatge, aquesta es passa a escala de grisos i s'aplica els thresholds. Després es troben tots els cantons amb l'algorisme Canny[24], un cop trobats tots els cantons de la imatge es calcula l'àrea més gran que pertany al full de l'examen i s'obté els quatre punts de les cantonades. Un cop tens els quatre punts es fa un resize de la imatge.

S'ha implementat una solució amb molt bons resultats com es veu a la figura 2.



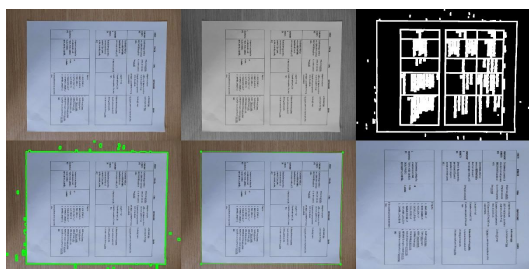


Fig. 2 - Procés de centrat d'una imatge

Encara que la implementació doni molt bons resultats Aquí sorgeix un problema, que és que en l'aplicació tenim les imatges en format bitmap i s'hauria de fer alguna transformació a string per poder enviar-ho al servidor de Python i aquí fer la transformació inversa per obtenir la imatge i aquesta a un format d'imatge que acceptin les funcions de la llibreria. Això és una inversió de temps que no es té en aquesta part del desenvolupament del treball.

## 7. Resultats

Els resultats finals del TFG son els següents:

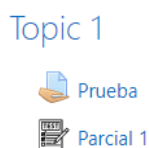


Fig. 3 - Activitat Examen

A la banda del Caronte està la base de l'extensió, és a dir, que es pugui crear, editar i eliminar una activitat Examen com es veu a la Figura 3 i, la validació per part del professor totalment funcional.

A la banda de l'aplicació mòbil estan tots els mòduls fets, funcionals i connectats correctament.

Per veure el funcionament de l'aplicació s'ha gravat un vídeo que es pot accedir per l'enllaç[25] que hi ha l'apartat de bibliografia. Si es visiona el vídeo es pot comprovar com el resultat del qual s'ha explicat durant tot el treball funciona correctament.

Les proves que hem fet per assegurar-nos que funciona correctament és fer el procés el qual es faria el dia de l'examen. Això consta del que es veu en el vídeo per part de l'estudiant i de la validació del professor des de Caronte. També hem provat de guardar les notes, publicar-les i penjar-les.

## 8. Conclusions

Durant aquest treball hem vist que s'ha plantejat un projecte ambiciós que barreja diferents àmbits dins l'enginyeria informàtica. Ens hem plantejat uns objectius i una planificació inicial que hem anat desenvolupant al llarg del semestre i com aquest han anat evolucionant a mesura que s'avançava.

Hem creat un primer prototip on totes les funcionalitats crítiques estan fetes. Aquestes funcionalitats són la pròpia extensió, l'aplicació, la validació tant en l'aplicació com en Caronte, l'enviament de les dades, que es pugui fer fotografies des de l'aplicació i l'exportació.

Hem vist com és l'execució de la nostra aplicació mòbil, també com és el procés de registre de l'empremta dactilar i la validació per poder entregar un examen.

Aquest és un projecte bastant viable de fer, sobretot si es té experiència programant en Moodle o PHP i en Android, ja que la falta d'experiència ens ha fet gastar recursos i temps en aprendre i buscar solucions a problemes més que en fer l'estudi de viabilitat i implementar la pròpia solució al projecte.

## 9. Bibliografia

- [1] Documentació Moodle. "Plugin" 14 de març de 2021. [https://docs.moodle.org/all/es/Plugins\\_FAQ](https://docs.moodle.org/all/es/Plugins_FAQ)
- [2] Documentació Moodle. "LTI y Moodle" 14 de març de 2021. [https://docs.moodle.org/all/es/LTI\\_y\\_Moodle](https://docs.moodle.org/all/es/LTI_y_Moodle)
- [3] Documentació Android. "Biometric" 14 de març de 2021. <https://developer.android.com/jetpack/androidx/releases/biometric?hl=es-419>
- [4] OpenCV. 14 març de 2021 <https://opencv.org/>
- [5] Documentació Moodle. "Universal Office Convert (unoconv)" 14 de març de 2021. [https://docs.moodle.org/all/es/Universal\\_Office\\_Converter\\_\(unoconv\)](https://docs.moodle.org/all/es/Universal_Office_Converter_(unoconv))
- [6] OCR API. 14 de març de 2021 <https://ocr.space/ocrapi>
- [7] Scrum. "¿Qué es Scrum?" 14 de març de 2021. <https://www.scrum.org/resources/blog/que-es-scrum>
- [8] Kanban tool. "¿Por qué utilizar la metodología Kanban?" 14 de març de 2021. <https://kanbantool.com/es/metodologia-kanban>
- [9] TFG. "Disseny i implementació d'una aplicació mòbil per poder connectar-se a Caronte i pujar

PDFs d'exàmens prèvia validació com a estudiant i autorització del professor" juny 2021.

[10] Protocol HTTP. "Protocolo de transferencia de hipertexto" 24 d'abril de 2021. [https://es.wikipedia.org/wiki/Protocolo\\_de\\_transferencia\\_de\\_hipertexto](https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto)

[11] Modulos. "Activity modules" 23 d'abril de 2021. [https://docs.moodle.org/dev/Activity\\_modules#Standard\\_Files\\_and\\_their\\_Functions](https://docs.moodle.org/dev/Activity_modules#Standard_Files_and_their_Functions)

[12] Algoritme CRAM. "Challenge-response authentication" 24 d'abril de 2021. [https://en.wikipedia.org/wiki/Challenge%20%93response\\_authentication](https://en.wikipedia.org/wiki/Challenge%20%93response_authentication)

[13] Test Captcha. "CAPTCHA" 24 d'abril de 2021. <https://en.wikipedia.org/wiki/CAPTCHA>

[14] Documentació Map. "Inteface Map<K, V>" 24 d'abril de 2021. <https://docs.oracle.com/javase/7/docs/api/java/util/Map.html>

[15] Stack Overflow. "HashMap get/put complexity" 25 d'abril de 2021. <https://stackoverflow.com/questions/4553624/hashtable-get-put-complexity>

[16] URL. "Localizador de recursos uniforme" 24 d'abril de 2021. [https://es.wikipedia.org/wiki/Localizador\\_de\\_recursos\\_uniforme](https://es.wikipedia.org/wiki/Localizador_de_recursos_uniforme)

[17] OCR. "Reconeixement òptic de caràcters" 29 de maig de 2021. [https://ca.wikipedia.org/wiki/Reconeixement\\_%C3%B2ptic\\_de\\_car%C3%A0cters](https://ca.wikipedia.org/wiki/Reconeixement_%C3%B2ptic_de_car%C3%A0cters)

[18] Xarxa Neuronal Convolutiva. "Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)" 29 de maig de 2021. <https://www.mdpi.com/1424-8220/20/12/3344/html>

[19] Dataset MNIST. "THE MNIST DATABASE of handwritten digits" 29 de maig de 2021. <http://yann.lecun.com/exdb/mnist/>

[20] FTP. "Protocol de transferència de fitxers" 29 de maig de 2021. [https://ca.wikipedia.org/wiki/Protocol\\_de\\_transfer%C3%A8ncia\\_de\\_fitxers](https://ca.wikipedia.org/wiki/Protocol_de_transfer%C3%A8ncia_de_fitxers)

[21] TensorFlow. 25 de Juny de 2021. <https://www.tensorflow.org/>

[22] Keras. 25 de Juny de 2021. <https://keras.io/>

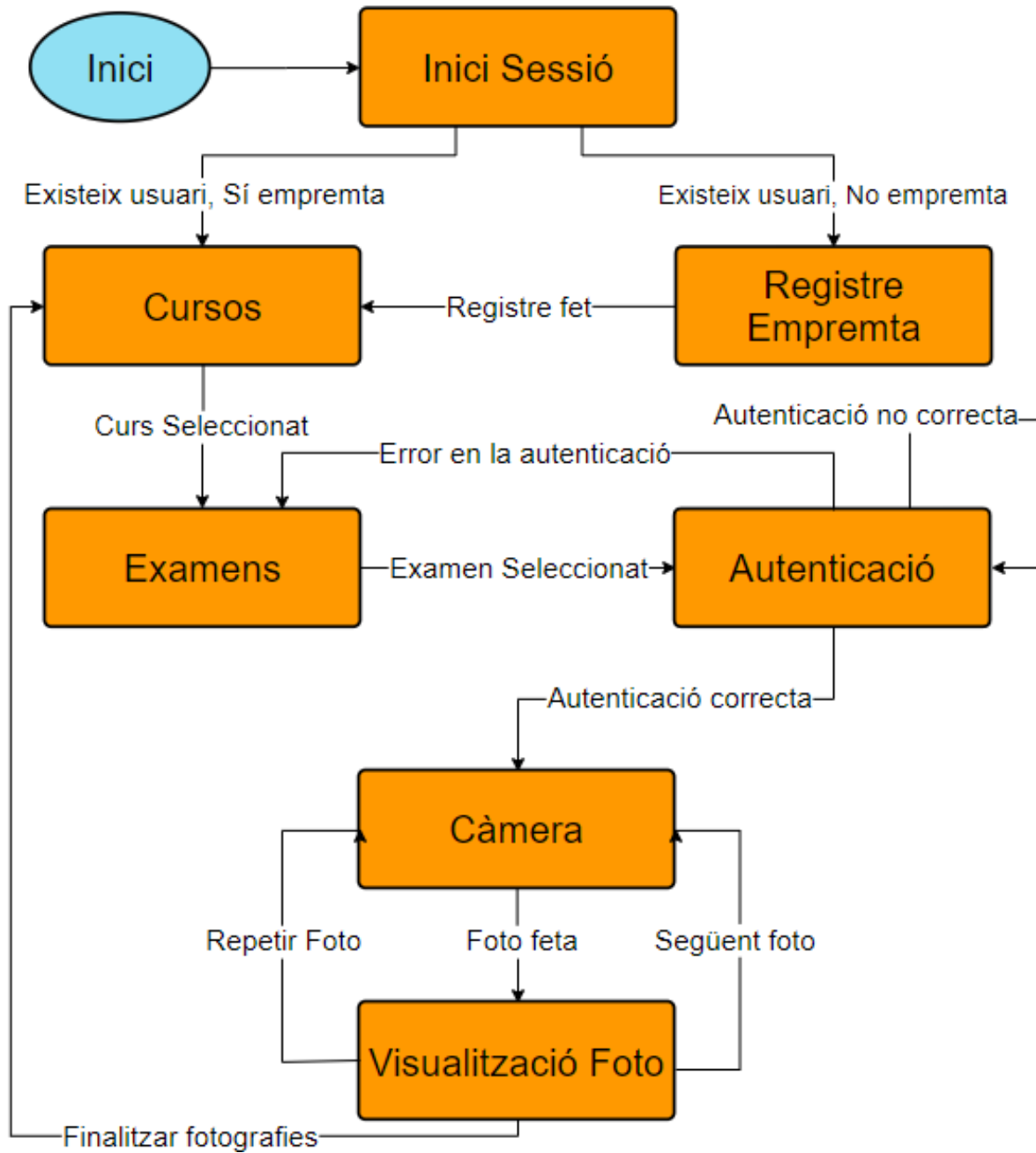
[23] CVS. 25 de Juny de 2021. <https://ca.wikipedia.org/wiki/CSV>

[24] Canny Edge Detection. 25 de Juny de 2021. [https://docs.opencv.org/3.4/da/d22/tutorial\\_py\\_canny.html](https://docs.opencv.org/3.4/da/d22/tutorial_py_canny.html)

[25] Video demostració de l'aplicació. 27 de juny de 2021. <https://youtu.be/gswR-sXQO9o>

## 10. Annex

### 10.1. Annex 1



10.2. Annex 2

0000 1111 2222  
3333 4444 5555  
6666 7777 8888  
9999

10.3. Annex 3

0000 1111 2222  
3333 4444 5555  
6666 7777 8888  
9999

10.4. Annex 4

0000 1111 22  
002222 3333  
4444 44555555  
666666 7777778  
88888 999999