



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

APLICACIÓN MÓVIL PARA LA CORRECCIÓN AUTOMÁTICA DE EXÁMENES TIPO TEST DE RESPUESTA SIMPLE

Alumno: Iván Ibáñez Torres

Tutor: Prof. D. Carlos Javier Ogayar Anguita
Dpto: Informática

Junio, 2017



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don Carlos Javier Ogayar Anguita, tutor del Proyecto Fin de Carrera titulado: APP móvil para la corrección automática de exámenes tipo test de respuesta simple, que presenta Iván Ibáñez Torres, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, JUNIO de 2017

El alumno:

Iván Ibáñez Torres

Los tutores:

Carlos Javier Ogayar Anguita

Tabla de contenido

| | |
|---|-----------|
| Escuela Politécnica Superior de Jaén | 1 |
| Escuela Politécnica Superior de Jaén..... | 1 |
| Grado en Ingeniería Informática | 1 |
| 1. INTRODUCCIÓN AL PROBLEMA..... | 5 |
| 1.1. Descripción del problema..... | 7 |
| 1.2. Motivación | 9 |
| 1.3. Soluciones disponibles..... | 10 |
| 1.4. Objetivos | 12 |
| 1.5. Estructura del documento | 13 |
| 2. Gestión y Planificación..... | 14 |
| 2.1. Planificación del proyecto | 14 |
| 2.1.1. Diagrama de Gantt..... | 17 |
| 2.1.2. Diagrama Pert..... | 18 |
| 2.2. Análisis de costes | 20 |
| 3. Análisis | 22 |
| 3.1. Requisitos | 23 |
| 3.1.1. Requisitos funcionales | 23 |
| 3.1.2. Requisitos no funcionales | 23 |
| 3.1.3. Requisitos del dispositivo del usuario final | 24 |
| 3.2. OCR (Reconocimiento óptico de caracteres) | 24 |
| 3.2.1. Conclusión | 27 |
| 3.3. Procesamiento imágenes | 27 |
| 3.3.1. Conclusión | 30 |
| 4. Diseño | 30 |
| 4.1. Diagrama de clases | 30 |
| 4.2. Diagrama de casos de uso | 32 |
| 4.3. Diagrama de actividad..... | 33 |
| 4.4. Patrón Modelo-Vista-Controlador | 34 |
| 4.5. Diagramas de secuencia..... | 35 |
| 4.5.1. Nuevo examen..... | 35 |
| 4.5.2. Eliminar examen | 36 |
| 4.5.3. Editar examen..... | 37 |
| 4.5.4. Corregir examen | 38 |
| 4.5.5. Eliminar calificación | 40 |
| 4.5.6. Exportar calificaciones | 41 |

| | |
|--|-----------|
| 4.6. StoryBoards | 42 |
| 5. Implementación | 49 |
| 5.1. Entorno de desarrollo | 49 |
| 5.2. Lenguajes de programación..... | 50 |
| 5.2.1. Lenguaje programación C..... | 50 |
| 5.2.2. C++ | 51 |
| 5.2.3. Objective-C | 51 |
| 5.2.4. Swift | 52 |
| 5.3. Detección de respuestas | 52 |
| 5.3.1. Cámara | 55 |
| 5.3.2. Procesamiento visual y reconocimiento de texto..... | 58 |
| 5.3.3. Detección y clasificación de las respuestas..... | 59 |
| 6. Conclusiones y posibles mejoras | 60 |
| Bibliografía..... | 62 |

1. INTRODUCCIÓN AL PROBLEMA

El examen es la prueba que una persona debe superar para demostrar que ha asimilado los conocimientos del tema que se proponga. En la educación, los profesores se valen de estas pruebas para confirmar que los alumnos están preparados para superar la materia.

Alumnos examinándose:



Ilustración 1 – Alumnos examinándose

Tomando la definición de evolución como calcular el valor de algo, se puede decir, que el término examen se encuentra directamente relacionado con este.

Según ha ido surgiendo la necesidad, a lo largo del tiempo se han definido varios tipos de exámenes, caracterizándose unos por un número de preguntas no muy alto y siendo estas de ámbito general respecto al tema que trate y con respuestas extensas. En este tipo de examen el docente, en cada pregunta, pide al alumno un punto del tema y este debe contestar con todo lo que conoce acerca de ese tema intentando ser lo más explicativo posible.

Otro tipo de examen son los que constan de un mayor número de preguntas, pero con unas respuestas mucho más breves y rápidas de contestar, con lo que se intenta que los alumnos demuestren conocimientos más específicos acerca de una cuestión en concreto. Suelen ser preguntas directas en las que se exige brevedad en la respuesta, sin exceder más de dos o tres líneas.

Por otro lado, tenemos otro tipo de examen que se conoce como examen multirespuesta en los que a cada pregunta se le adjuntan varias posibles respuestas y los alumnos deben elegir entre ellas la respuesta correcta o la considerada más correcta ya que se puede dar el caso en el que varias o todas las respuestas son correctas, pero, una de ellas prevalece sobre todas las demás, siendo esta la correcta y las demás incorrectas.

Examen multirespuesta:

| Diplomatura Estadística e I.T.I.G CURSO 3º. EXAMEN BBDD 2 1806113 Usuario: GR_____ | | |
|---|---------------|------------|
| APPELLIDOS _____ | NOMBRE: _____ | DNI: _____ |
| Antes de comenzar el examen se ruega leer las siguientes notas: | | |
| 1.- Al realizar el examen, el alumno entregará este gráfico y un lápiz (incluidas las hojas de encuadre). Si el alumno decide no entregar el examen, deberá cumplimentar la hoja de respuesta y dejarla en el escritorio del examinador. 2.- Durante la realización del examen, el Cónsul o cónsul de estudiantes del examinado deberá de permanecer sobre el pupitre de forma visible. 3.- La duración del examen es de 1 hora. 4.- Cualquier persona que sea sorprendida manipulando un teléfono móvil será automáticamente excluida de la convocatoria. 5.- Marca las respuestas que consideres correctas. Cada pregunta puede tener entre 0 y 4 respuestas correctas. | | |
|  | | |
| 1. (7 puntos) Marca las respuestas que consideres correctas. Cada pregunta puede tener entre 0 y 4 respuestas correctas. Para que una pregunta se dé por correcta deben marcarse todas las respuestas correctas. Las preguntas mal respondidas no restan nota. Cada pregunta correcta suma 0,25 puntos. | | |
| i. La independencia del modelo físico de datos entonces debemos modificar el modelo lógico de datos <ul style="list-style-type: none"> a. Si modificamos el modelo lógico de datos no es necesario modificar el modelo físico de datos b. Cualquier modificación en una BBDD conllevará la necesaria modificación de las aplicaciones que la accedan c. La independencia de datos se establece solo entre la capa física y lógica de la BBDD | | |
| ii. Respecto a las variables en PL/SQl: <ul style="list-style-type: none"> a. Pueden ser de tipo base y de tipo andaluz b. Las de tipo andaluz pueden ser de otras cosas, ser del mismo tipo que una columna de una tabla c. Una de tipo andaluz, entre otras cosas, eliminará varias filas de una tabla d. Las de tipo andaluz pueden, entre otras cosas, almacenar una única fila de un cursor | | |
| iii. Respeto a los procedimientos almacenados: <ul style="list-style-type: none"> a. Es un procedimiento que se ejecuta en el DBMS b. Es un procedimiento que se almacena en el DBMS pero se ejecuta en el cliente que lo invoca c. Es un procedimiento que se ejecuta en el DBMS pero se almacena en el cliente que lo invoca d. Permite compartir código entre distintas sesiones | | |
| iv. Seu la sentencia SQL UPDATE X SET X.a=X.a+1. Un disparador declarado "CREATE TRIGGER TRG_UPD_X_A AFTER INSERT" <ul style="list-style-type: none"> a. Se ejecutará una única vez después de realizar ninguna actualización b. Se ejecutará una vez antes de actualizarse cada una de las filas de la tabla X c. No tendrá acceso a las variables :new ni :old d. Tendrá acceso de lectura a la tabla X | | |
| v. Una tabla mutable: <ul style="list-style-type: none"> a. No puede ser leída por un disparador de tupla b. No puede ser modificada por un disparador de tupla c. Puede ser leída por un disparador de sentencia d. Es una tabla a la que le ha picado una araña radioactiva | | |
| vi. El diccionario de datos: <ul style="list-style-type: none"> a. Un diccionario de datos es el encargado de mantenerlo actualizado b. Un usuario solo puede realizar operaciones de lectura sobre éste c. Describe, entre otras cosas, el modelo lógico de datos de la BBDD d. Describe, entre otras cosas, el modelo físico de datos de la BBDD | | |
| vii. Tareas del DBA son: <ul style="list-style-type: none"> a. Evaluar el hardware disponible para el DBMS b. Instalar el DBMS c. Planificar la Base de Datos d. Realizar copia de seguridad de la Base de Datos completa | | |
| viii. En Oracle, decimos que los grupos de ficheros <i>redolog</i> están multiplexados si y solo si: <ul style="list-style-type: none"> a. Hay más de un fichero <i>redolog</i> online b. Hay más de un fichero <i>redolog</i> por grupo c. El modo ARCHIVELOG está activado d. No hay ficheros <i>redolog</i> offline | | |
| ix. En Oracle, los espacios de tablas o <i>tablespace</i> : <ul style="list-style-type: none"> a. No son parte de la base de datos b. Son organizados en segmentos, extensiones y bloques de datos c. Son manipulados por el proceso database writer (dbwr) d. Se almacenan en uno o más archivos | | |

Ilustración 2 - Examen multirespuesta

Por último, tenemos el examen de respuesta simple, el cual se centra este Trabajo Fin de Grado, este tipo de exámenes suele caracterizarse por tener un número elevado de preguntas y en el que solo se dan las opciones de verdadero o falso. En ocasiones la persona examinada debe decidir entre una de las dos opciones que se le dan a elegir o dejar la pregunta en blanco si no está seguro de cuál es la respuesta correcta, ya que una respuesta mal contestada puede

considerarse como puntuación negativa en el momento de realizar el cálculo global del resultado del examen.

Examen respuesta simple Verdadero/Falso:

| Criptografía y Seguridad en Computadores 5 de septiembre de 2005. Examen | |
|---|---------------|
| Apellidos y Nombre: _____ | D.N.I.: _____ |
| A. Indique si cada uno de los siguientes enunciados es verdadero o falso, colocando una V o una F en la casilla correspondiente. Tenga en cuenta que las respuestas erróneas restan la mitad del valor de una respuesta correcta (5 puntos). | |
| <i>Pregunta</i> | |
| 1. Una curva elíptica está compuesta por una serie de puntos que cumplen cierta condición, más un punto \mathcal{O} situado en el infinito. 2. La criptografía de curva elíptica permite definir versiones sobre este formalismo de los algoritmos asimétricos más comunes. 3. Existen algoritmos eficientes para saber si un número es o no primo, pero no para factorizarlo. 4. Los registros de desplazamiento son el mecanismo más seguro para construir cifrados de flujo, pero son muy lentos. 5. Los criptosistemas simétricos emplean la misma clave para cifrar y descifrar. 6. El algoritmo DES no se considera seguro porque se le han encontrado fallos de diseño. 7. La confusión está asociada al cambio de posición (transposición) de los símbolos. 8. La criptografía asimétrica puede usarse tanto para proteger información, como para firmarla digitalmente. 9. Una S-Caja es la unidad básica de transposición en algoritmos de cifrado por bloques. 10. Los criptosistemas asimétricos emplean la misma clave para cifrar y descifrar. 11. Una red de Feistel tiene la propiedad de que puede usarse tanto para cifrar como para descifrar, empleando las inversas módulo n de las sub-claves en lugar de éstas. 12. Un MDC es una función de autenticación que necesita una clave para poder ser calculada. 13. El modo de operación ECB no permite detectar sustituciones e inserciones de bloques en el mensaje. 14. El problema de los logaritmos discretos puede resolverse de forma eficiente empleando computación paralela. 15. Cuando se genera un par de llaves PGP, conviene obtener un certificado de revocación y guardarlo para su posterior uso. | |
| Sigue en la página siguiente... | |
| 1 | |

Ilustración 3 - Examen respuesta simple

1.1. Descripción del problema

Este trabajo fin de grado, de ahora en adelante TFG, consiste en crear una aplicación móvil para la corrección automática de exámenes tipo test de respuesta simple mediante la cámara del dispositivo.

En este tipo de examen, normalmente se le indica al alumno donde debe escribir la respuesta, la localización de estas casi siempre se encuentran en uno de los márgenes laterales del examen, haciendo coincidir en línea horizontal el lugar de la respuesta con el texto de dicha pregunta. Hay ocasiones en las que las respuestas son anotadas en un cuadrante, ya sea en la misma hoja o en otra a parte en la que se indica el número de la pregunta y la respuesta.

Hay distintos tipos de respuestas que se pueden dar en este tipo de examen como puede ser redondear o tachar la opción correcta, escribir las iniciales de V y F o incluso llenar algún tipo forma geométrica como puede ser un cuadrado o círculo.

Hoja de respuesta de un examen tipo test de respuesta simple:

Ilustración 4 - Hoja respuestas simples

El principal problema a la hora de plantear esta aplicación se da en el momento de diseñar una plantilla de examen por defecto pensando en los alumnos que posteriormente tendrán que rellenarlo, pensando que disponen de un tiempo limitado y debe ser algo relativamente fácil de entender y de contestar.

Pensando en los profesores que son los que deben editar el examen y completarlo con las preguntas correspondientes al tema, se busca un formato que pueda ser modificado con cualquier editor de textos en cualquier momento sin necesidad de marcas especiales.

A la hora de corregir el examen, la aplicación debe simplificar el máximo posible, el proceso de comparar las respuestas del alumno con la plantilla de soluciones. Por lo que el formato del examen debe ser analizado por la aplicación mediante procesamiento visual de la imagen y que este proceso no ralentice la aplicación y suponga un elevado tiempo de respuesta de esta.

En definitiva, el principal problema es la creación de un examen tipo, atractivo visualmente, similar a los que se realizan normalmente en papel, sin que suponga un gran esfuerzo por parte del profesor a la hora de crear uno nuevo, ni un problema para el alumno a la hora de responder las cuestiones que se le plantean y por supuesto analizable por parte del dispositivo en un tiempo razonable.

1.2. Motivación

Este TFG está dentro de la rama Tecnologías de la Información, ya que se utilizan diferentes herramientas y servicios para el proceso de corrección de exámenes y posterior envío de resultados, además está dentro de la mención de Sistemas Gráficos ya que se utiliza procesamiento de imágenes para la captura de los exámenes y mejoras en la calidad para el posterior análisis para la extracción de las respuestas del alumno.

El realizar una aplicación para el móvil me llamaba mucho la atención ya que a día de hoy las tecnologías móviles están en alza y tenía mucho interés por aprender y hacer algo en alguna de las plataformas móvil como es Swift, totalmente desconocido para mí y un nuevo reto.

Otro de los motivos por el cual he escogido este TFG, es porque creo que puede ser una aplicación que puliendo algunos detalles puede llegar ayudar y ser bastante útil tanto a los profesores ahorrando tiempo a la hora de corregir los exámenes como a los alumnos para disminuir la espera de publicación de las calificaciones.

1.3. Soluciones disponibles

Para entender un poco mejor los objetivos de esta aplicación, se hará un breve recopilatorio con el software disponible actualmente con el mismo fin que este TFG.

- Gexcat:

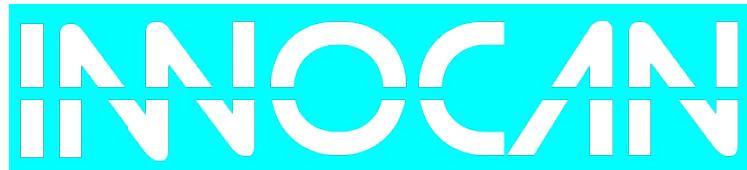


Ilustración 5 - Innocan

Software propiedad de INNOCAN SISTEMAS S.L., empresa TIC dedicada al desarrollo de aplicaciones basadas en el tratamiento de datos e imágenes.

Oferta una aplicación de escritorio con una fácil de generación de exámenes, la cual recibe estos mediante unas imágenes obtenidas con un escáner.

Plantilla examen de ejemplo Gexcat:

Ilustración 6 - Ejemplo examen Gexcat

- EXAMIO:



Ilustración 7 - ExamIO

ExamIO es un programa basado en la tecnología OMR (Reconocimiento Óptico de Marcas) que sirve para diseñar, corregir y analizar resultados de encuestas y pruebas tipo test realizados en papel, de manera automática.

Al igual que la anterior, es una aplicación de escritorio, requiere un escáner como entrada de los exámenes y recomienda uno con alimentador de hojas para facilitar la tarea de escaneo al usuario.

Plantilla examen ejemplo ExamIO:

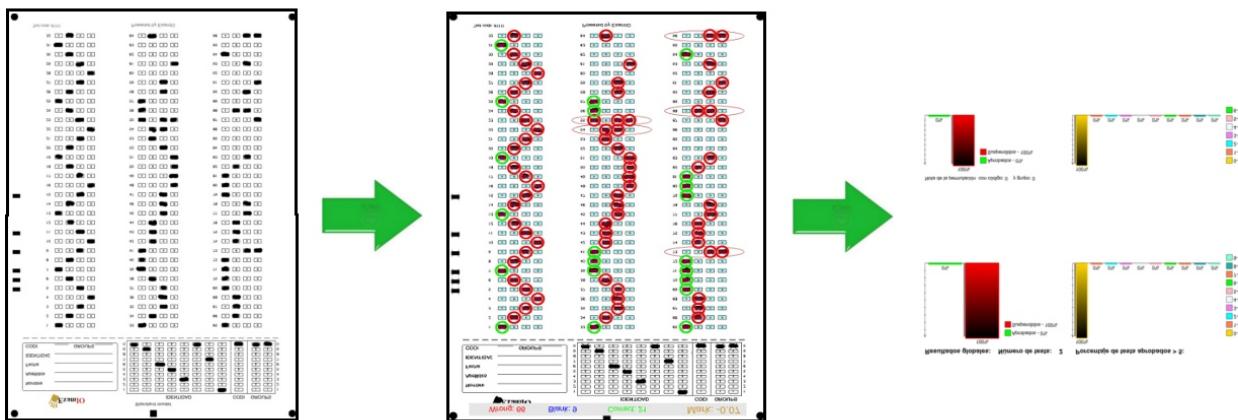


Ilustración 8 - Plantilla ejemplo ExamIO

- COETEST:

Software desarrollado en Java, como aplicación multiplataforma para escritorio, por Darío Gutiérrez, Lucas Díaz Sanzo, del Departamento de Informática de la Universidad de Oviedo.

Permite generar las hojas para realizar el examen, que se imprimen en papel normal. El alumno marca su identificador, el modelo y las respuestas en la hoja. Después únicamente se necesita un escáner (mejor automático), para digitalizar las hojas y llevarlas al ordenador.

Plantilla examen de ejemplo Gexcat:

| | | | | | | | |
|------|---|---|---|---|---|---|---|
| DNI: | 7 | — | 7 | 7 | 7 | — | — |
| 1 | | | | | | | |
| 2 | | | | | | | |
| 3 | | | | | | | |
| 4 | | | | | | | |
| 5 | | | | | | | |
| 6 | | | | | | | |
| 7 | X | | | | X | X | X |
| 8 | | | | | | | |
| 9 | | | | | | | |
| 0 | | | | | | | |
| 1 | | | | | | | |
| 2 | | | | | | | |
| 3 | | | | | | | |
| 4 | | | | | | | |
| 5 | X | | | | | | |
| 6 | X | | | | | | |
| 7 | X | | | | | | |
| 8 | | | | X | | | |
| 9 | | | | | | | |
| 10 | | | | | | | |

| | | | | |
|----|---|---|---|---|
| A | B | C | D | E |
| 1 | | X | | |
| 2 | | X | | |
| 3 | | | X | |
| 4 | | | X | |
| 5 | X | | | |
| 6 | X | | | |
| 7 | X | | | |
| 8 | | | X | |
| 9 | | | | X |
| 10 | | | | |

Ilustración 9 - Hoja respuestas Gexcat

Como se ha podido observar en las opciones listadas, las soluciones software para abordar el problema descrito son pocas, de reducida movilidad y con necesidad de un desembolso si no se dispone de un escáner. Otra característica común en las tres aplicaciones es una hoja de respuestas independiente de las preguntas. Esto puede dificultar la realización de la prueba y provocar fallos en la respuesta.

1.4. Objetivos

El objetivo principal de este trabajo es crear una aplicación móvil que facilite a los docentes la tarea de corregir los exámenes tipo test de respuesta simple de manera cómoda, rápida y eficaz a través de una aplicación simple e intuitiva.

Para llevar a cabo este desarrollo será necesario analizar los requisitos e implementar una solución que los satisfaga.

Para desarrollar este proyecto se necesitará:

- Investigar acerca de las aplicaciones existentes con el mismo fin o parecido para analizar tanto las posibilidades que ofrecen como los puntos donde falla e intentar mejorarlos.
- Realización de la guía oficial de Apple para desarrolladores principiantes en Swift
- Analizar distintos formatos de exámenes tipo test de respuesta simple utilizados por otras herramientas similares.
- Diseñar una plantilla en la que las preguntas y respuestas se encuentren en la misma hoja para facilitar la realización de este al alumno.
- Investigar acerca de las librerías de procesamiento visual disponibles para Swift.
- Investigar acerca de las librerías OCR disponibles para Swift.
- Implementación de la aplicación.

1.5. Estructura del documento

La documentación de este trabajo se divide en varias partes:

Una primera parte presenta la gestión y planificación del proyecto, en este apartado se tratará de documentar cómo se organizará el desarrollo de este proyecto, así como los costes de este.

En la parte del análisis hablaremos de los diferentes requisitos de nuestro proyecto, además estudiaremos los distintos tipos de librerías que podemos utilizar y las herramientas que se utilizarán para el desarrollo de este.

En la parte del diseño se desarrollarán los diferentes diagramas como de clases, casos de uso, actividad o de secuencia.

En la parte de la implementación se tratará de explicar las diferentes herramientas utilizadas, y una explicación más concisa del desarrollo del proyecto. En este apartado se explicarán los códigos más relevantes.

Por último, se presentarán unas conclusiones y mejoras acerca de este proyecto.

2. Gestión y Planificación

En este capítulo se tratará de planificar, localizar en el tiempo y documentar las diferentes tareas en las que se basa este proyecto, además se calculará el coste de este.

2.1. Planificación del proyecto

Este proyecto se comenzó el día 20 de febrero de 2017 y se ha terminado el día 25 de junio de 2017, por lo que se ha estado desarrollando durante 125 días, durante los cuales se han trabajado unas 2'5 horas diarias, alcanzando así y superando las 300 horas obligatorias de este Trabajo Fin de Grado.

El ciclo de vida del desarrollo Software (SDLC en sus siglas inglesas), es una secuencia estructurada y bien definida de las etapas en Ingeniería de software para desarrollar el producto software deseado.

Para este proyecto se ha utilizado una modelo de ciclo de vida incremental, de este modo se han dividido los problemas en sub-problemas más sencillos y en los que cada uno forma parte de un incremento. Cada incremento es indispensable para poder continuar con el siguiente, una vez terminado un incremento se comprueba que todo funciona correctamente y se pasa al siguiente.

Algunos de los objetivos de esta metodología son:

- Poder enseñar un producto operativo pero incompleto al cliente en períodos cortos de tiempo.
- Los requisitos más importantes de la aplicación son incluidos en los primeros incrementos del proyecto.
- Cada incremento tiene un tiempo determinado y breve.
- No hay marcha atrás, cualquier modificación necesaria a un incremento anterior, será incluido en un incremento nuevo y posterior.

Las tareas quedan divididas de la siguiente forma:

| Sub-problemas | Duración (Horas) |
|--|------------------|
| Interfaz gráfica básica | 15 |
| Captura foto examen | 20.6 |
| Mejora calidad imagen | 37.4 |
| OCR y procesamiento texto | 31.4 |
| Gestión y almacenamiento de exámenes | 12.6 |
| Gestión y almacenamientos de calificaciones | 12.6 |
| Mejora visual general de la app | 25.4 |
| Exportación de calificaciones para posterior envío | 2.6 |
| Total | 157.6 |

Tabla 1 – División de tareas

En la siguiente tabla se detallan cada una de las actividades y el tiempo dedicado a cada una de ellas:

| Tarea | Duración (Días) | Horas diarias | Duración (Horas) | Fecha Inicio | Fecha Fin |
|--|-----------------|---------------|------------------|--------------|-----------|
| Búsqueda Bibliográfica | 8 | 2 | 16 | 20/02/17 | 28/02/17 |
| Análisis | 10 | 2 | 20 | 01/03/17 | 10/03/17 |
| Elección de herramientas | 28 | 2 | 56 | 11/03/17 | 07/04/17 |
| Pruebas OpenCV | 11 | 2 | 22 | 11/03/17 | 21/03/17 |
| Pruebas ImageMagick | 8 | 2 | 16 | 22/03/17 | 29/03/17 |
| Pruebas Tesseract OCR | 9 | 2 | 18 | 30/03/17 | 07/04/17 |
| Desarrollo | 78 | 2 | 157.6 | 08/04/17 | 25/06/17 |
| Interfaz gráfica básica | 7.6 | 2 | 15 | 08/04/17 | 15/04/17 |
| Análisis | 4 | 2 | 8 | 08/04/17 | 11/04/17 |
| Diseño | 2 | 2 | 4 | 12/04/17 | 13/04/17 |
| Implementación | 1.7 | 2 | 3.4 | 14/04/17 | 15/04/17 |
| Captura foto examen | 10.3 | 2 | 20.6 | 16/04/17 | 25/04/17 |
| Análisis | 4.3 | 2 | 8.6 | 16/04/17 | 19/04/17 |
| Diseño | 2.7 | 2 | 5.4 | 20/04/17 | 22/04/17 |
| Implementación | 3.3 | 2 | 6.6 | 23/04/17 | 25/04/17 |
| Mejora calidad imagen | 18.7 | 2 | 37.4 | 26/04/17 | 14/05/17 |
| Análisis | 8.3 | 2 | 16.6 | 26/04/17 | 03/05/17 |
| Diseño | 4.3 | 2 | 8.6 | 04/05/17 | 07/05/17 |
| Implementación | 6 | 2 | 12 | 08/05/17 | 14/05/17 |
| OCR y procesamiento texto | 15.7 | 2 | 31.4 | 15/05/17 | 30/05/17 |
| Análisis | 6.3 | 2 | 12.6 | 15/05/17 | 20/05/17 |
| Diseño | 5.3 | 2 | 10.6 | 21/05/17 | 26/05/17 |
| Implementación | 4 | 2 | 8 | 27/05/17 | 30/05/17 |
| Gestión y almacenamiento de exámenes | 6.3 | 2 | 12.6 | 31/05/17 | 05/06/17 |
| Análisis | 2 | 2 | 4 | 31/06/17 | 01/06/17 |
| Diseño | 1.3 | 2 | 2.6 | 02/06/17 | 02/06/17 |
| Implementación | 3 | 2 | 6 | 03/06/17 | 05/06/17 |
| Gestión y almacenamientos de calificaciones | 6.3 | 2 | 12.6 | 06/06/17 | 11/06/17 |
| Análisis | 2 | 2 | 4 | 06/06/17 | 07/06/17 |
| Diseño | 1.3 | 2 | 2.6 | 08/06/17 | 08/06/17 |
| Implementación | 3 | 2 | 6 | 09/06/17 | 11/06/17 |
| Mejora visual general de la app | 11.7 | 2 | 25.4 | 12/06/17 | 23/06/17 |
| Análisis | 4 | 2 | 8 | 12/06/17 | 15/06/17 |
| Diseño | 5.7 | 2 | 11.4 | 16/06/17 | 21/06/17 |
| Implementación | 2 | 2 | 4 | 22/06/17 | 23/06/17 |
| Exportación de calificaciones para posterior envío | 1.3 | 2 | 2.6 | 24/06/17 | 25/06/17 |
| Análisis | 0.7 | 2 | 1.4 | 24/06/17 | 24/06/17 |
| Diseño | 0.3 | 2 | 0.6 | 24/06/17 | 24/06/17 |
| Implementación | 0.3 | 2 | 0.6 | 24/06/17 | 25/06/17 |
| Documentación | 125 | 0'5 | 62'5 | 20/02/17 | 25/06/17 |
| Total | 125 | 2'5 | 312'5 | 20/02/17 | 25/06/17 |

Tabla 2 - Planificación horas

2.1.1. Diagrama de Gantt

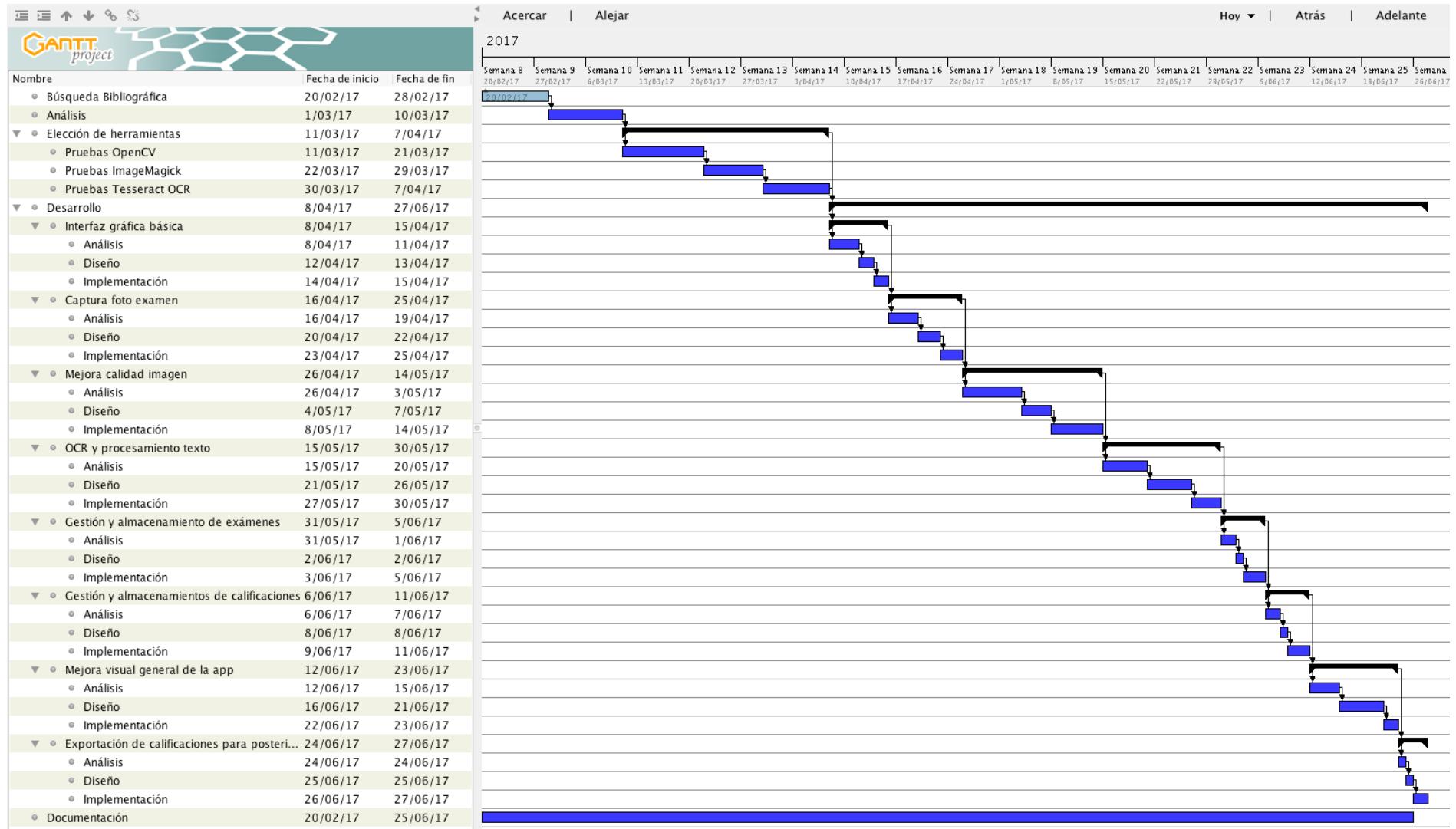


Ilustración 10 - Diagrama Gantt

2.1.2. Diagrama Pert

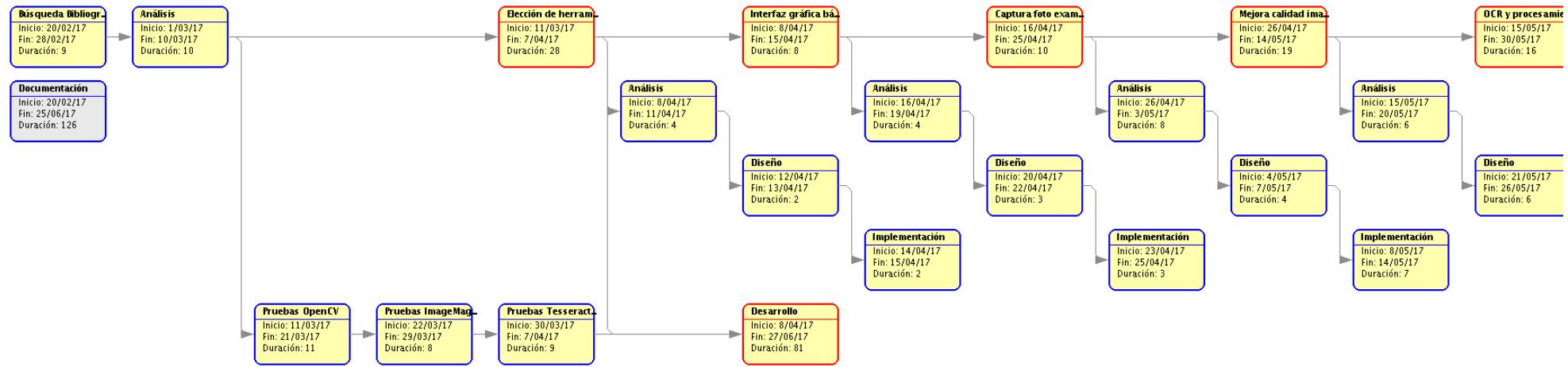
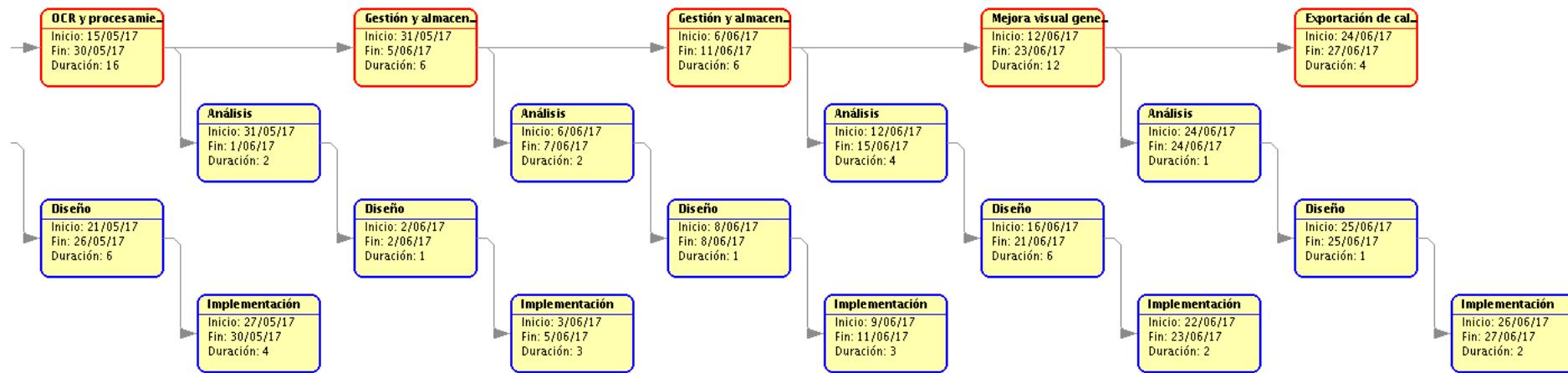


Ilustración 11 - Diagrama de Pert



2.2. Análisis de costes

En este apartado se hará una estimación de los costes que conlleva la realización de este proyecto.

Para ello, hay que tener en cuenta todos los elementos que intervienen en el proyecto como son software y hardware utilizado, conexión a internet para la consulta de información, licencia de desarrollador de Apple para la publicación de la App en la App Store.

Comenzando por el software, para la implementación de la aplicación se ha utilizado el entorno de desarrollo integrado (IDE) Xcode corriendo bajo el sistema operativo macOS Sierra. Para los retoques realizados en imágenes se utilizará el programa de edición GIMP, la edición de esta documentación y los distintos modelos de exámenes utilizados para las pruebas han sido llevados a cabo con la aplicación Microsoft Word incluida en el paquete Microsoft Office 2016, para los diagramas de clase, casos de uso y demás, se utilizará la herramienta Visual Paradigm. Los diagramas utilizados para la representación de la planificación del tiempo han sido editados con la aplicación GanttProject.

Para hacer un análisis de costes lo más real posible, es necesario calcular la parte proporcional del coste del paquete Office, estimando un tiempo de vida útil de 2 años. El coste del paquete se corresponde a 0,00850457€/hora.

Coste software del proyecto:

| Software | Precio |
|-----------------------|--------------|
| Xcode | 0 |
| macOS Sierra | 0 |
| GIMP | 0 |
| Microsoft Office 2016 | 3,18 |
| Visual Paradigm | 19,00 |
| GanttProject | 0 |
| Total | 22,18 |

Tabla 3 - Coste software proyecto

Por la parte hardware, se ha utilizado un MacBook Pro Retina 15", mediados 2014 con un procesador Intel Core i7, 16GB de memoria RAM DDR3 y como dispositivo de almacenamiento un disco SSD de 256GB, por un precio de 1.888,37€. Se supone que el equipo será utilizado para más trabajos y no ha sido adquirido expresamente para el proyecto, por lo que hay que calcular la parte proporcional. La vida útil del dicho equipo suele ser 5 años por lo que es un total de 43800 horas. El coste de uso por hora del equipo corresponde a 0,0443€.

Como dispositivo real para las pruebas de la app se ha utilizado un iPhone SE con un procesador A9 dual-core de 64bits, memoria RAM de 2GB y 16GB de almacenamiento. El dispositivo tiene un precio de 299€. Al igual que el equipo anteriormente especificado, hay que calcular la parte proporcional del coste asociado a este proyecto. La vida útil de este dispositivo suele estar entre los 3-4 años, lo que da unas 35040 horas. Este dispositivo ha estado implicado en un 70% del desarrollo de este proyecto, por lo que en el cálculo también hay tenerlo en cuenta. El coste de utilización de este dispositivo es de 0,0085€/h.

En el proceso visual de las imágenes y el reconocimiento del texto se realizaron una cantidad importante de impresiones de distintos formatos de exámenes con una impresora All In One HP OfficeJet 3833, con un coste 52,99€. Realizando el mismo calculo con una vida útil de 2-3 años el coste por hora equivale a 0,0020€. La implicación de la impresora en el proyecto corresponde a un 25% del total de este.

Coste hardware del proyecto:

| Hardware | Precio Unidad(€) | Unidades | Total |
|---------------------|------------------|----------|---------------|
| Equipo desarrollo | 0,04 | 312,5 | 13,47 |
| Dispositivo pruebas | 0,009 | 306,25 | 2,61 |
| Impresora | 0,002 | 109,38 | 0,22 |
| Total | | | 12,26€ |

Tabla 4 - Coste hardware proyecto

Hay que incluir el coste de personal que tiene el proyecto. Para ello se hacen los cálculos basándose en “XVI Convenio colectivo estatal de empresas de consultoría y estudios de mercados de la opinión pública” publicado en el BOE el 4 de abril de 2009. El sueldo anual de un titulado es de 20954,36 anuales. Al realizar el cálculo del coste por hora del personal se obtiene 9,922€. El total de horas de este proyecto es de 437,5 horas por lo que el coste de personal asciende a 4340,875€. Teniendo en cuenta el gasto de seguridad social que suele rondar el 30% del sueldo, este sube a 5643,1375€.

Toda la información y ayuda necesita acerca del tema trabajado ha sido extraída íntegramente en internet, por lo que se ha necesitado una conexión a internet contratada con ONO de 30Mb simétricos de fibra óptica con un precio mensual de 39,45€. Realizando el cálculo para los 4 meses que dura el proyecto el coste de la conexión a internet asciende a 157,8€.

Por último, para la distribución de la App se necesita una licencia de desarrollador para poder publicarla en la App Store. Esta tiene un coste anual de 99€.

Coste del desarrollo del proyecto:

| Actividad | Precio (€) |
|------------------------|----------------|
| Software | 22,19 |
| Hardware | 12,26 |
| Personal | 5643,14 |
| Conexión Internet | 157,8 |
| Licencia desarrollador | 99 |
| Total | 5934,39 |

Tabla 5 - Coste del proyecto

3. Análisis

Esta fase es la más importante en el desarrollo de una aplicación, es imprescindible un buen análisis para crear una buena base para el proyecto. En ella se analiza el problema detalladamente, los requisitos que debe cumplir la aplicación para un correcto funcionamiento y la solución escogida.

3.1. Requisitos

Los requisitos son las funcionalidades o restricciones que la aplicación debe cumplir. Este apartado es dedicado al análisis de estos requisitos, ya sean requisitos funcionales o requisitos no funcionales.

3.1.1. Requisitos funcionales

En los requisitos funcionales se incluyen las posibilidades que al App debe brindar al usuario final a la hora de corregir exámenes.

Los requisitos funcionales de esta App son:

- Corregir exámenes tipo test de respuesta simple mediante la ayuda de la cámara del dispositivo móvil.
- Gestión de exámenes (añadir, editar, eliminar).
- Gestión de calificaciones asociadas alumnos.
- Exportación y posterior envío de calificaciones.

3.1.2. Requisitos no funcionales

Los requisitos no funcionales describen aspectos del sistema que están relacionados con el grado de cumplimiento de los requisitos funcionales.

Los requisitos no funcionales de esta aplicación son:

- Un formato de examen simple, para una fácil edición a la hora de crearlo.
- Preguntas y repuestas juntas, sin necesidad de hojas independientes.
- Utilización de caracteres en el reconocimiento de respuestas que no influya en la redacción de las preguntas.
- Un formato de examen atractivo visualmente para el alumno que realiza la prueba.
- Debe permitir corregir exámenes en distintas sesiones, almacenando el proceso y reanudándolo donde se dejó.
- Interfaz simple, que permita la corrección de exámenes fácilmente.

3.1.3. Requisitos del dispositivo del usuario final

Como requisito hardware se recomienda utilizar la App en un dispositivo iPhone 5s o cualquiera de los modelos posteriores a este. Por debajo de este modelo no se puede obtener un gran rendimiento de procesamiento.

La aplicación esta compilada para la versiona 10 de iOS por lo que el dispositivo utilizado debe estar actualizado como mínimo a dicha versión o posterior.

3.2. OCR (Reconocimiento óptico de caracteres)

Para el desarrollo de este trabajo, un requisito fundamental es la obtención de las respuestas elegidas por los alumnos para poder calcular la calificación de este en la prueba. Para ello se utilizará el proceso de reconocimiento óptico de caracteres. A continuación, veremos las diferentes alternativas disponibles para iOS, investigadas para este TFG:

- **BlinkOCR SDK:**



Ilustración 12 - microblink

BlinkOCR SDK es un módulo de OCR desarrollado por MicroBlink. Su tecnología OCR está optimizada específicamente para la arquitectura de dispositivos móviles. Ofrecen una API Key de prueba al registrarse en su web con 100 ejecuciones.

- **Anyline OCR Document:**



Ilustración 13 - ANYLINE

Desarrollado por Anyline, SDK multiplataforma que encapsula todo el proceso de reconocimiento mediante la configuración de parámetros como son el tamaño de la letra, diseño de fuente, color, etc. dependiendo del documento a escanear. Ofertan un mes gratuito por registrarte en la web. Después la suscripción es anual con packs de escaneos.

- **SwiftOCR:**

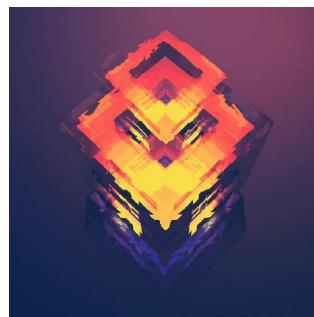


Ilustración 14 - SwiftOCR

Librería OCR escrita por Nicolas Camenisch en Swift bajo licencia Apache 2.0 disponible para iOS y OS X. Solo esta optimizada para el reconocimiento de códigos alfanuméricos cortos de una línea y no para documentos extensos.

- **ABBY Fine Reader Engine:**



Ilustración 15 - ABBYY

Propiedad de ABBY, basado en su plataforma de reconocimiento, aseguran buen rendimiento sin un gran coste de recursos. Ofrece reconocimiento inteligente de caracteres (ICR) y código de barras. Licencia gratuita para prueba de 30 días.

- **Scansbot SDK:**



Scansbot

Ilustración 16 - Scansbot

SDK de fácil integración desarrollado por PSPPDFKit. Disponible para Android e iOS en Java y Objective-C respectivamente. El núcleo esta implementado de forma independiente en C++. Dispone de detección de documentos, *cropping* y optimización de imagen. Puede obtenerse una licencia de prueba poniéndose en contacto con ellos, después el pago se realiza mediante suscripción.

- **Orbit SDK:**



Orbit SDK

Ilustración 17 - Orbit SDK

Desarrollado por OCR Labs, el motor de este SDK combina una alta tasa de captura de imágenes con una conversión óptica de alto rendimiento. Administra inteligentemente los caracteres, la iluminación del entorno e incluso los movimientos de la mano para ofrecer mejores resultados con alta precisión, aseguran un promedio del 98% y menor tiempo. Para obtener una licencia de prueba es necesario contactar mediante correo electrónico.

- **Tesseract OCR iOS:**

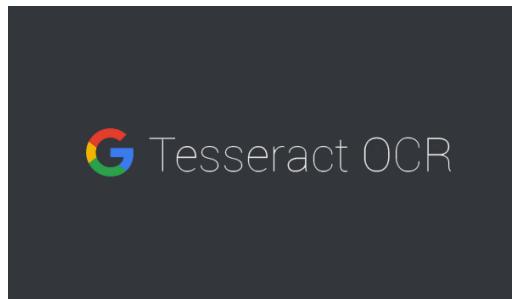


Ilustración 18 - Tesseract OCR

Tesseract es un motor OCR Open Source, desarrollado originalmente como software propietario por Hewlett Packard en 1995, liberado posteriormente y actualmente desarrollado por Google y distribuido bajo licencia Apache 2.0. Considerado de los mejores motores libres disponibles en la actualidad. Con reconocimiento de más de 60 idiomas.

3.2.1. Conclusión

Se han realizado pruebas tanto con las distintas versiones de prueba como con la librería Open Source, comparando tiempo de reconocimiento, recursos consumidos y resultados obtenidos no se ha apreciado una gran diferencia entre el software de pago y el libre, por lo que para este proyecto la librería que será utilizada será Tesseract que no supone ningún coste.

3.3. Procesamiento imágenes

El rendimiento del motor OCR a la hora de detectar los datos en el examen está directamente relacionado con la calidad de la imagen que se le pase. Por lo que la elección de una buena librería para el procesamiento visual de imágenes, es otro de los puntos importantes de este TFG. A continuación, se listan las diferentes alternativas disponibles para iOS:

- **GPUImage2:**

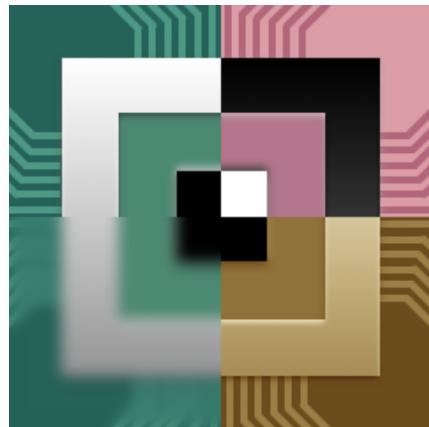


Ilustración 19 - GPUImage2

Esta librería para iOS está distribuida bajo licencia BSD, permite aplicar filtros personalizados a imágenes, captura en tiempo real de cámara y videos. Para agilizar el procesamiento de las imágenes se ayuda de la GPU para realizar los cálculos.

- **Toucan:**



Ilustración 20 - Toucan

Toucan es una librería para Swift que provee de una API limpia y sencilla para el procesamiento de imágenes. Funciones de redimensionamiento de imágenes, recorte y estilizado de imágenes.

- **ImageMagick:**



Ilustración 21 - ImageMagick

Es un paquete software Open Source para la creación, composición y conversión de imágenes de mapa de bits. Compatible con más de 200 formatos. ImageMagick se puede usar para redimensionar, reflejar, recortar. También incluye funcionalidades para realizar ajustes de color, escribir texto o dibujar líneas, polígonos, etc.

- **OpenCV:**



Ilustración 22 - OpenCV

OpenCV (Open Source Computer Vision Library) distribuida bajo licencia BSD. Con interfaz en C++, C, Python y Java y soportado en Windows, Linux, Mac OS, iOS and Android. OpenCV fue diseñado pensando en la eficiencia computacional y centrándose en las aplicaciones de tiempo real. Implementado para aprovechar el procesamiento multi-core.

Esta librería es conocida y utilizada en todo el mundo, con una comunidad de 47000 usuarios.

3.3.1. Conclusión

Analizando las opciones disponibles y realizando algunas pruebas se ha optado por utilizar la librería OpenCV, principalmente por los buenos resultados obtenidos en ocasiones anteriores en otros trabajos realizados y por la gran comunidad de usuarios de la que dispone en la que poder realizar cualquier consulta en caso de duda.

4. Diseño

En este capítulo se analizará más a fondo cada una de las características que tendrá la aplicación, lo que nos permitirá implementarlo de forma efectiva. Sigue un patrón Modelo-Vista-Controlador. Se estudiarán los diferentes diagramas y se mostrará una primera vista de lo que será esta aplicación.

4.1. Diagrama de clases

En este apartado se muestra el diagrama de clases que compone este proyecto.

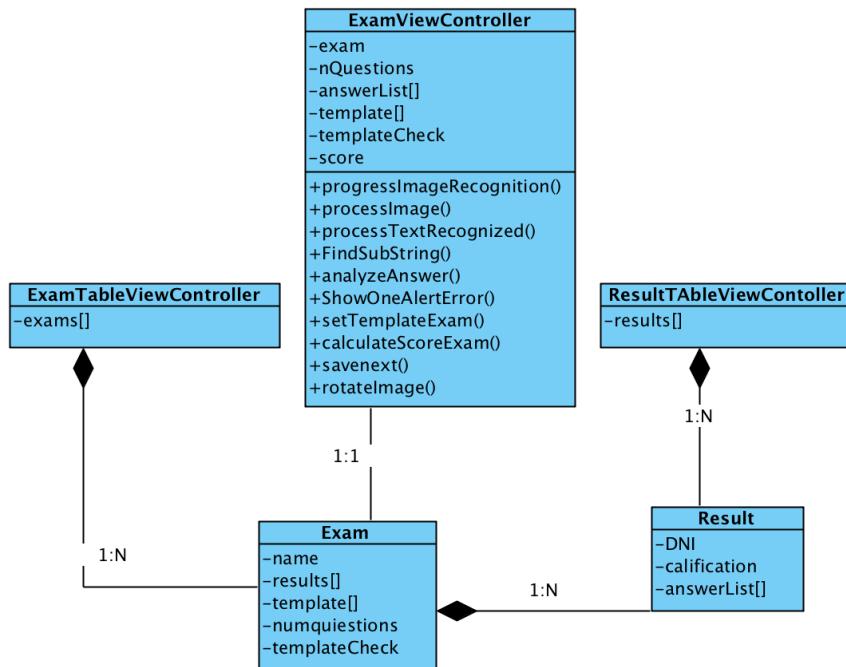


Ilustración 23 - Diagrama de clases

La clase Exam es la base de esta aplicación, es la entidad real que se va a tratar y la que se intenta abstraer. Un examen está compuesto por un nombre para la prueba, un listado con los resultados, una plantilla de corrección y el número de preguntas que consta el examen.

Un resultado está asociado a un alumno identificado por su D.N.I., una calificación obtenida para la prueba y un listado con todas las opciones escogidas por el alumno.

Como se puede observar en el diagrama, la clase con más funciones de la aplicación es ExamViewController, es la encargada de gestionar el procesamiento del examen. En ella están implementados los principales métodos que se utilizan para dicho fin. A continuación, se describe la funcionalidad de cada uno de ellos:

- progressImageRecognition: Encargado de informar acerca del estado del procesamiento y reconocimiento del examen.
- processImage: En este método se realiza una parte del procesamiento visual de la imagen, detección del texto del examen.
- processTextRecognized: El objetivo de esta función es localizar en el texto reconocido el número de la pregunta y con la ayuda de FinSubString la posición de esta en el texto para extraerla.
- analizeAnswer: Una vez extraída la respuesta, este método la evalúa y la codifica en la lista d respuestas según sea verdadera, falsa o sin respuesta.
- ShowOneAlertError: Comprueba de una lista de preguntas que no han sido evaluadas correctamente y muestra una ventana de alerta que permite introducir la respuesta elegida por el alumno.
- setTemplateExam: Cuando todas las preguntas del examen tienen un valor correcto, se da la opción de establecer ese examen como plantilla. Este es el método encargado de hacerlo.
- calculateScoreExam: Calcula la nota obtenida por el alumno en la prueba, una vez que todas las preguntas tienen un valor correcto y ya hay una plantilla de corrección establecida.

- Savenext: Permite almacenar la nota del alumno en corrección y reiniciar la interfaz para continuar con el siguiente.
- rotatetImage: Si la imagen es tomada de lado con esta función se tiene la opción de rotarla.

Las clases `ResultTableViewController` y `ExamTableViewController` son las encargadas de controlar las vistas donde se muestran los listados de exámenes y resultados almacenados dentro de estos.

4.2. Diagrama de casos de uso

A continuación, se muestra el diagrama de casos de uso:

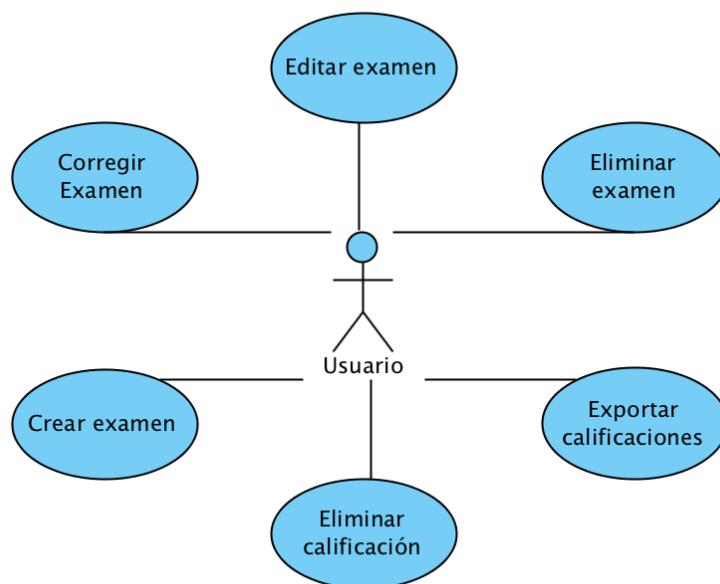


Ilustración 24 - Diagrama de casos de uso

La aplicación permite al usuario:

- Crear un nuevo examen
- Editar examen
- Corregir el examen de un alumno
- Eliminar un examen
- Eliminar la calificación de un alumno
- Exportar los resultados para enviarlos

4.3. Diagrama de actividad

En la siguiente imagen se muestra el diagrama de actividad que sigue la aplicación. A continuación, se detalla cada uno de los pasos.

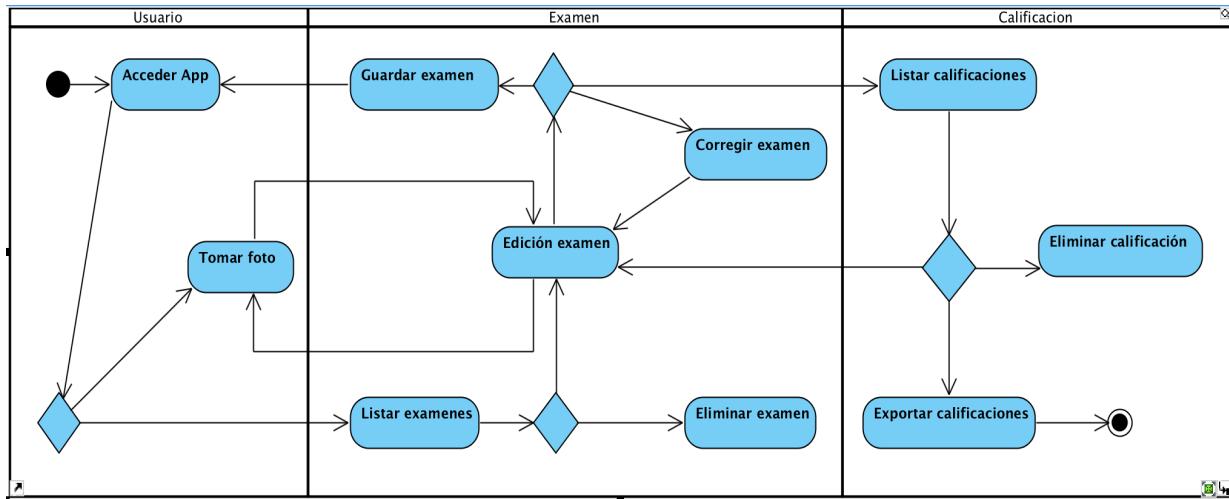


Ilustración 25 - Diagrama de actividad

Al acceder a la aplicación, el usuario se encuentra con dos opciones, puede fotografiar un examen directamente o acceder al listado de exámenes guardados. Desde el listado tiene la opción de crear un nuevo examen, donde tendrá que realizar la configuración inicial con el número de preguntas que tiene y empezar a tomar imágenes para su procesamiento. Otra opción de listado es seleccionar alguno de los exámenes guardados y acceder a la vista de edición donde podrá continuar con la corrección donde se quedó la última vez. Desde el listado también se puede eliminar exámenes deslizando el dedo sobre su fila de derecha a izquierda.

Al seleccionar la opción de fotografiar directamente en la vista principal, se activará la cámara del dispositivo y una vez tomada la foto, el usuario es redirigido a la vista de creación de exámenes para la configuración inicial.

En la vista de edición, una vez fotografiado el examen, el usuario debe procesar la imagen y corregir los errores detectados. Si hay una plantilla establecida para el examen, el usuario podrá obtener la nota del alumno, almacenarla y continuar corrigiendo otro examen. Antes de calcular la nota, el examen debe tener una plantilla de corrección establecida, esta puede ser modificada en cualquier momento.

En esta misma ventana, se encuentra la opción de listar las calificaciones siempre y cuando la lista no este vacía, en este listado es posible eliminar cualquier calificación si hubiera ocurrido algún error en el proceso de corrección o, si se ha terminado de corregir todos los exámenes, exportar el listado a un fichero csv. Este fichero se podrá enviar por correo electrónico, WhatsApp, DropBox, etc.

4.4. Patrón Modelo-Vista-Controlador

El patrón modelo-vista-controlador (MVC) separa la representación de las vistas de la aplicación de la interacción con estas. Es el patrón más extendido en el desarrollo software. Consta de tres capas, como su nombre indica, la capa del modelo, la capa de la vista y la capa del controlador.

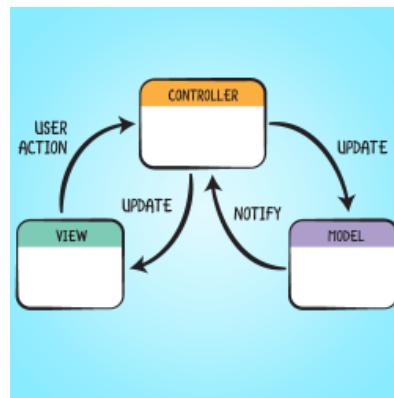


Ilustración 26 - Patrón MVC

La principal ventaja de este patrón es la flexibilidad que ofrece a la hora de poder hacer cualquier modificación a cada capa independientemente. El objetivo es separar la interfaz que se le muestra al usuario de los datos de la aplicación y de la lógica que controla estos datos.

- El modelo es el encargado de mantener la gestión y estructura de los datos de la aplicación. Es el que permite el procesamiento como la creación, eliminación o modificación. Normalmente suele corresponder con un Sistema Gestor de Base de Datos.
- El controlador es el encargado de cargar la información en la vista que el usuario ve. También actúa de intermediario entre la vista y el modelo avisando al modelo de alteración de los datos desde la interfaz.

- La vista es la encargar de dar formato a la información que el controlador le carga proveniente del modelo en forma de interfaz.

4.5. Diagramas de secuencia

Estos diagramas representan la interacción entre los objetos en una línea temporal. Al utilizar un patrón modelo-vista-controlador, el diagrama muestra la interacción entre estas capas.

4.5.1. Nuevo examen

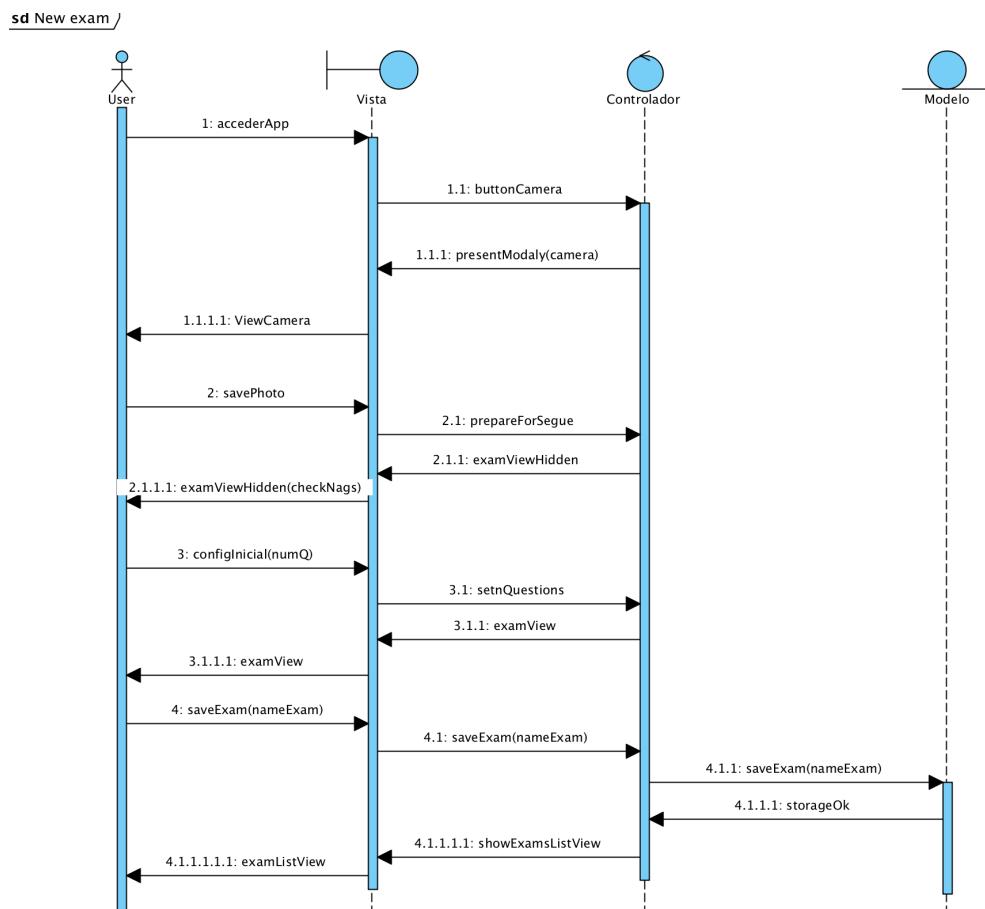


Ilustración 27 - Diagrama secuencia nuevo examen

Nada más abrir la App, el usuario tiene la opción de activar directamente la cámara del dispositivo o listar el histórico de exámenes guardados. En esta secuencia el usuario selecciona la opción de abrir la cámara y capturar la foto del examen.

Una vez realizada la foto del examen, la App es redirigida a la vista de configuración del examen donde el usuario tendrá como primer paso obligatorio

introducir el número de preguntas de este. Ya configurado el número de preguntas, el usuario tiene introducir como mínimo el nombre del examen para que le botón de almacenar examen se active.

4.5.2. Eliminar examen

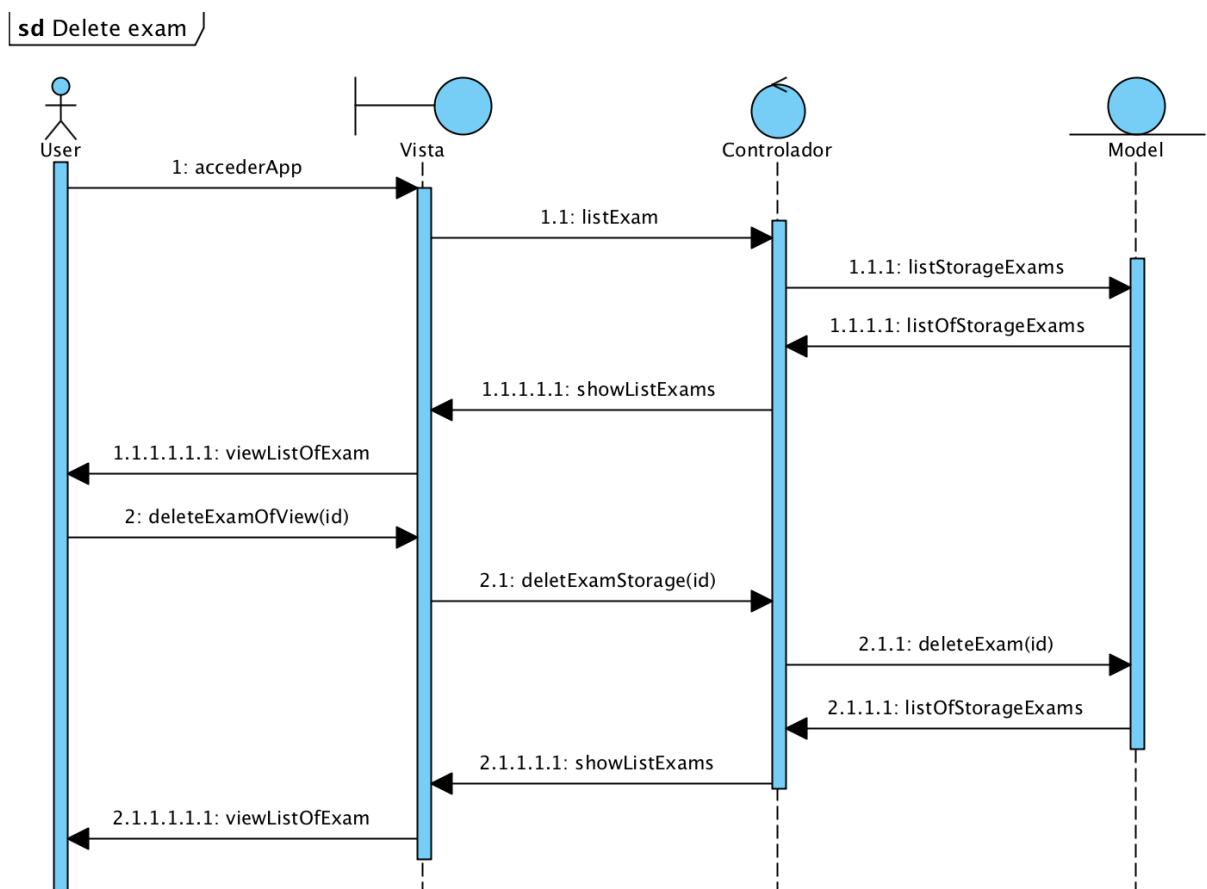


Ilustración 28 - Diagrama secuencia eliminar examen

En esta secuencia, al abrir la App, el usuario selecciona la opción de listar el histórico de exámenes almacenados. Desde el listado, el usuario puede deslizar el dedo de derecha a izquierda sobre la celda del examen que se quiere borrar, a la derecha de dicha celda aparece un botón rojo que hay que pulsar para eliminar el examen.

La vista comunica al controlador que se ha eliminado un elemento de la tabla, este se lo comunicar al modelo y este le devuelve una lista actualizada sin el examen eliminado en la vista. El controlador le devuelve este la lista a la vista y ya el usuario la recibe formateada gráficamente.

4.5.3. Editar examen

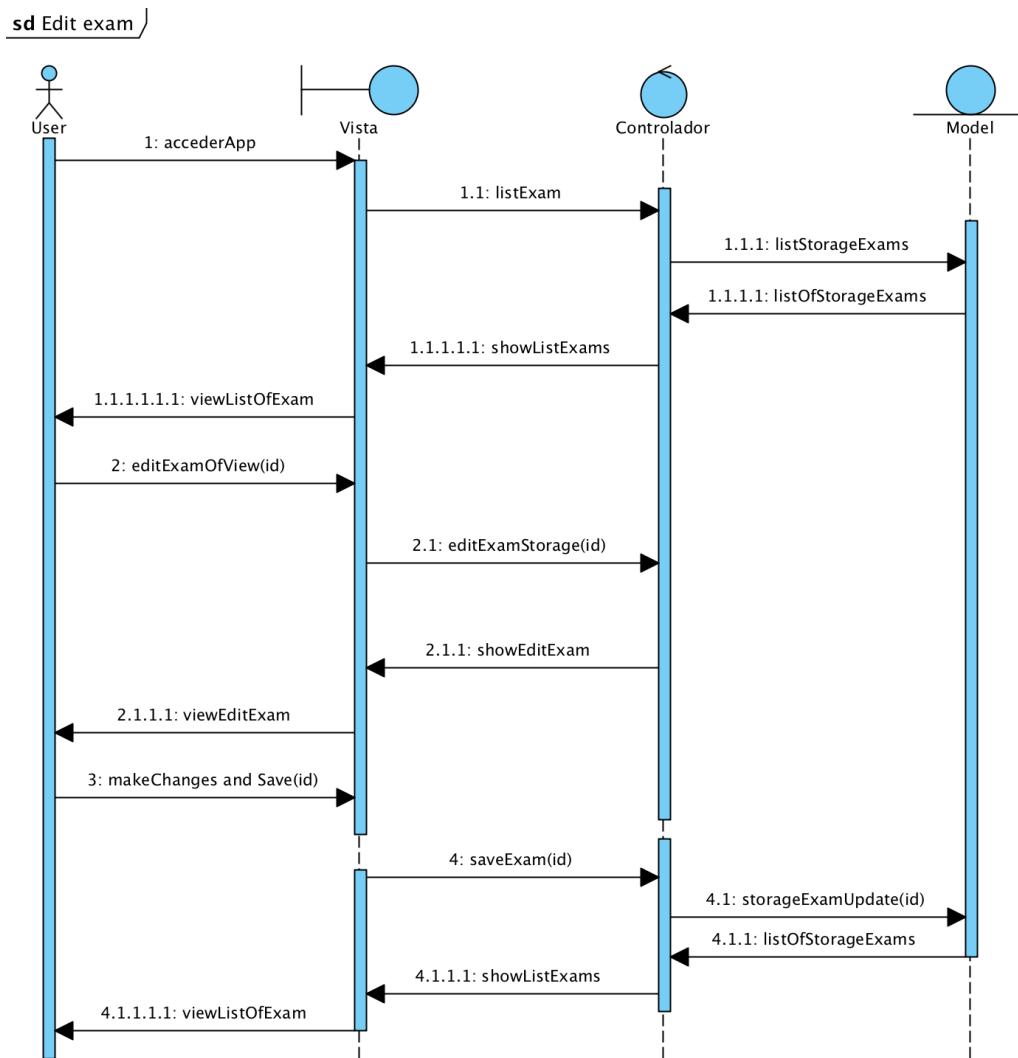
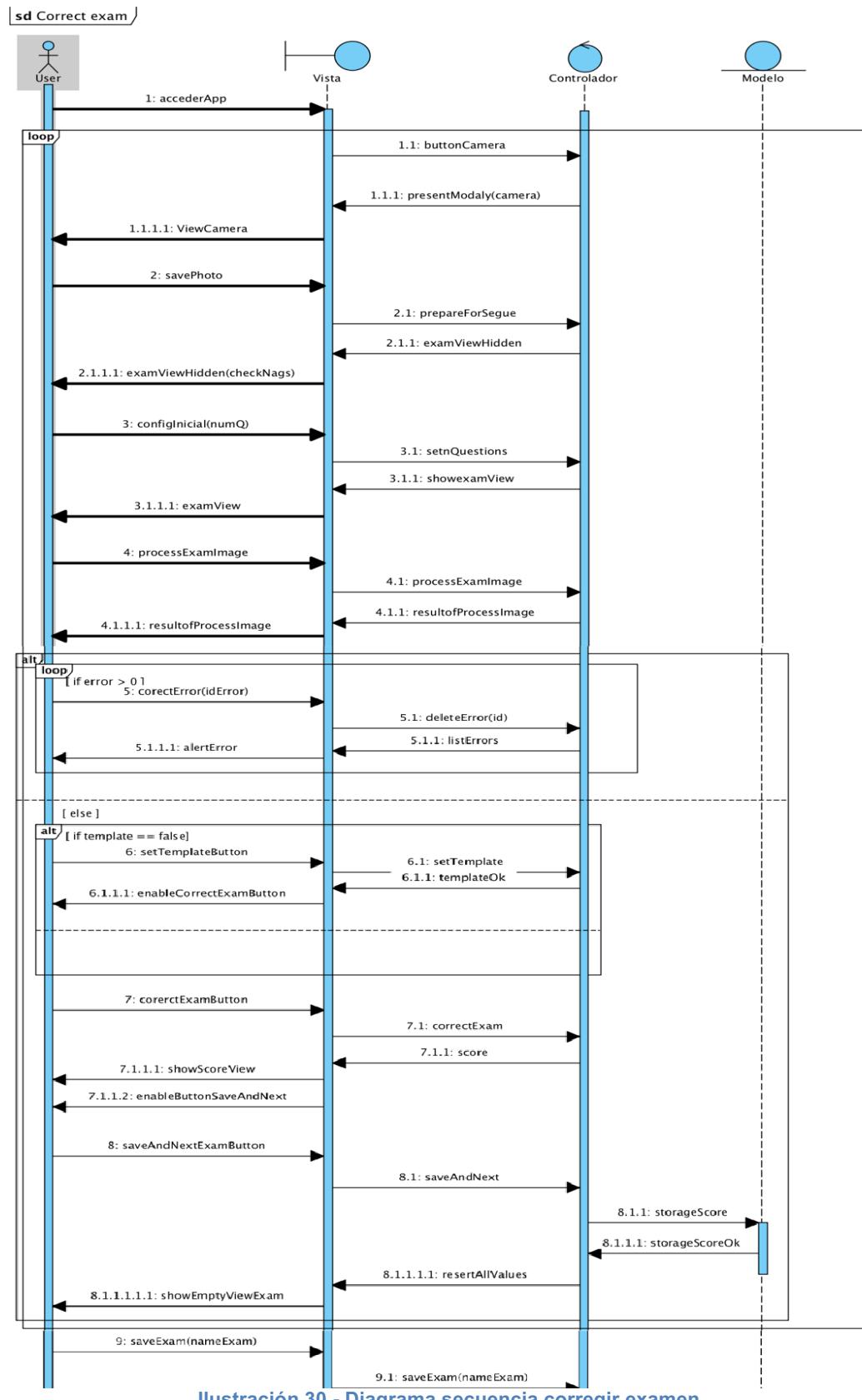


Ilustración 29 - Diagrama secuencia editar examen

En esta secuencia, al abrir la App, el usuario selecciona la opción de listar el histórico de exámenes almacenados. Desde el listado, el usuario puede seleccionar la celda del examen que quiere editar, la vista será redirigida a la de edición de exámenes y en ella el usuario podrá realizar los cambios que deseé.

La vista comunica al controlador que se ha editado un examen de forma correcta, este se lo comunicará al modelo y este le devolverá una lista actualizada con los cambios realizados por el usuario. El controlador, le devolverá este la lista a la vista y ya el usuario la recibirá formateada gráficamente.

4.5.4. Corregir examen



El usuario abre la aplicación y selecciona la activación de la cámara para realizar la foto al examen directamente, una vez tomada la foto deseada, el usuario es redirigido a la configuración inicial del examen.

Procesara la imagen tomada y si hubiera algún error, el usuario puede corregirlos manualmente o capturar otra fotografía para intentar tomar de nuevo las repuestas erróneas. El profesor no podrá continuar el proceso de corrección sin corregir los errores.

Una vez solventados todos los errores, el controlador comprueba si hay una plantilla de corrección establecida para el examen. Si no hay ninguna plantilla establecida, el usuario solo tendrá la opción de establecer ese examen como plantilla de corrección. Si, por el contrario, el usuario ya ha establecido una plantilla de corrección el controlador comunicara a la vista la activación del botón de obtener la nota del alumno.

Cuando el profesor obtiene la nota, el proceso de corrección para ese alumno ha concluido. Al calcular la nota, el controlador vuelve a comunicar a la vista la activación del botón que permite al usuario almacenar la nota y limpiar la vista para continuar con el examen del siguiente alumno.

4.5.5. Eliminar calificación

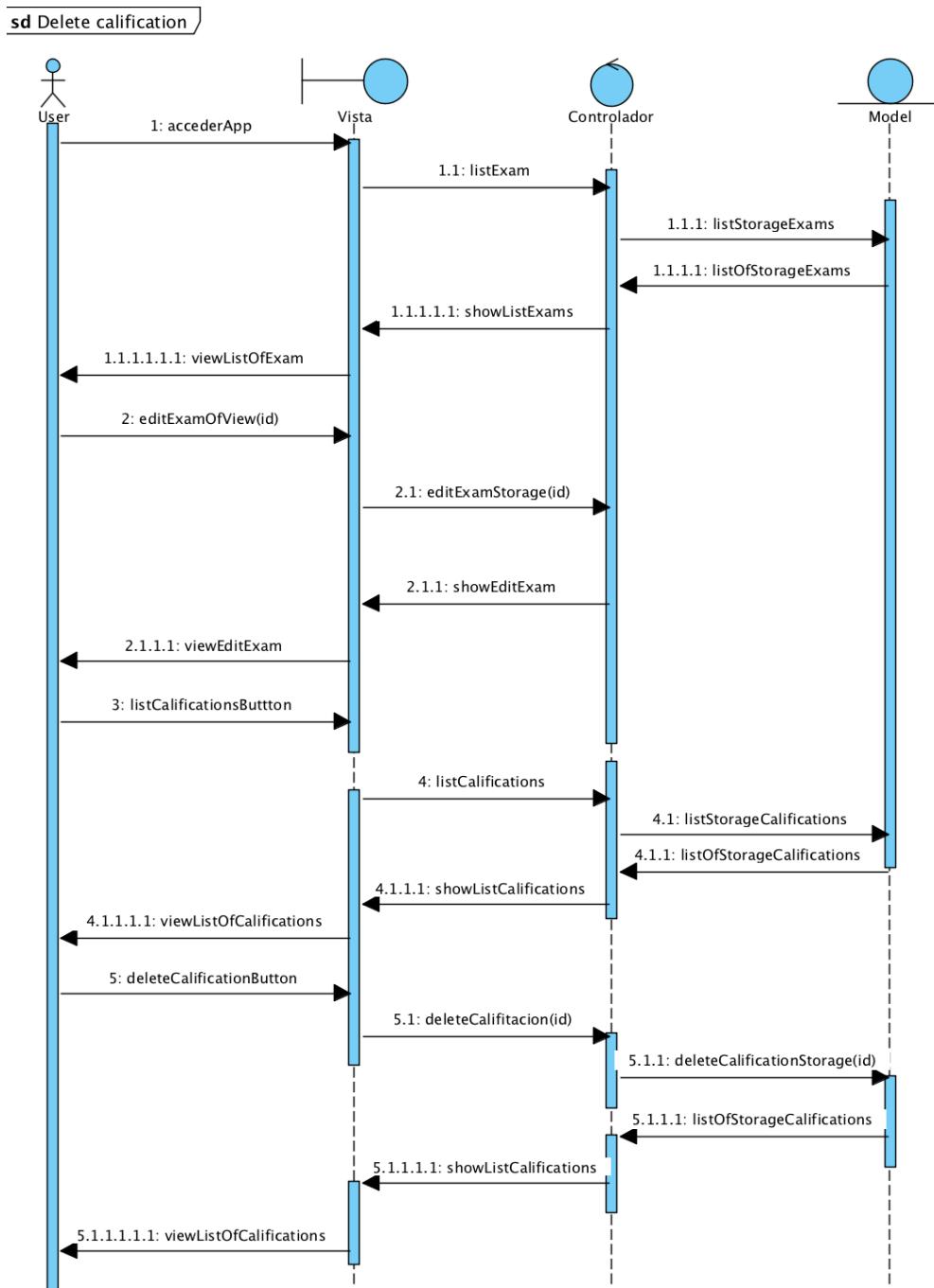


Ilustración 31 - Diagrama secuencia eliminar calificación

En esta secuencia, el usuario, al abrir la App, accede al listado de exámenes almacenados, pulsando sobre alguno de ellos se le presenta vista de edición del examen. Si el examen seleccionado tiene almacenados resultados, el usuario podrá acceder al listado de estos.

En este listado, al igual que en listado de exámenes, el usuario podrá deslizar el dedo de derecha a izquierda sobre la celda del resultado que desea eliminar y aparecerá botón para eliminarla.

4.5.6. Exportar calificaciones

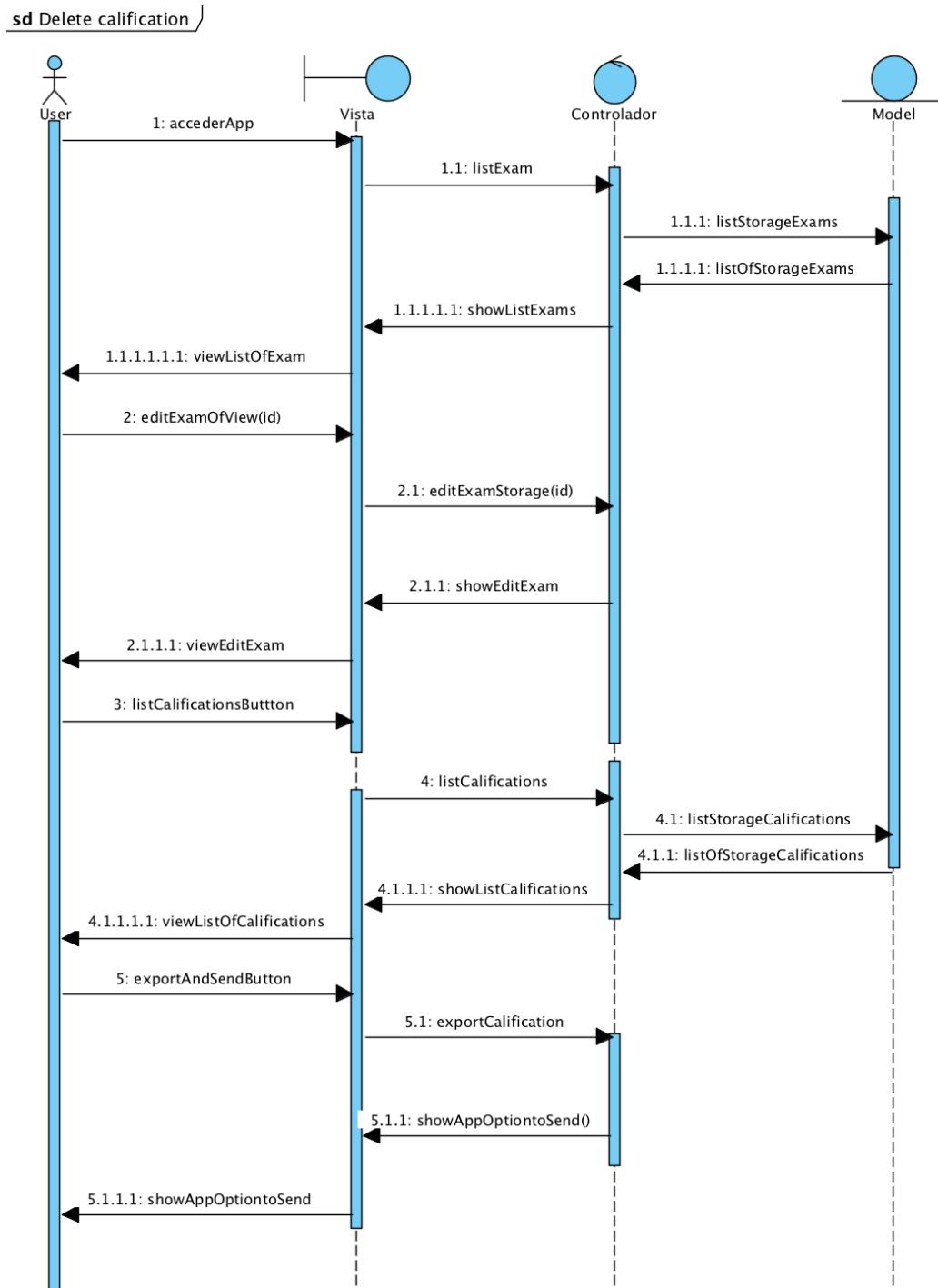


Ilustración 32 - Diagrama secuencia exportar calificaciones

Desde el listado de calificaciones, al que hemos explicado en la secuencia anterior la manera de llegar a él, se puede realizar la exportación de los resultados a un fichero ‘csv’ y mostrar al usuario una lista de aplicaciones disponibles para enviar o compartir este tipo de fichero como por ejemplo son las aplicaciones de correo o mensajería.

4.6. StoryBoards

Este apartado está dedicado al diseño de un prototipo para la interfaz de usuario que tendrá la aplicación. También se ilustrará las distintas opciones que tendrá el usuario y sus resultados.

El uso de StoryBoard, ayuda a tomar una primera visión de cómo la aplicación final, debería llegar a mostrarse al usuario.

A continuación, se muestran las imágenes cómo será la interfaz.

La pantalla principal, muestra las dos opciones que tiene el usuario, realizar una foto para empezar la corrección directamente o consultar el listado de exámenes almacenados.

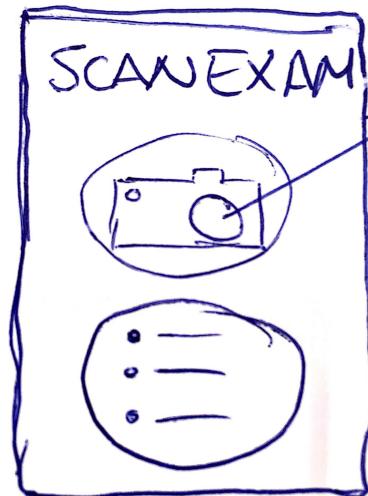


Ilustración 33 - Storyboard principal

La primera vez que el usuario ejecute la App, es muy probable que la primera opción que elija sea la de tomar la foto del examen, por lo que se seguirá ese itinerario y después se continuara por el listado.



Ilustración 34 - Storyboard cámara

La vista de la cámara se compone de tres botones al iniciarla:

- Home: Arriba a la izquierda, este botón permite al usuario volver a la pantalla principal de la aplicación. Este ícono estará presente en casi todo momento.
- Flash: Abajo a la izquierda, este botón es utilizado para entornos con poca iluminación, activa el flash del dispositivo.
- Filter: A la derecha del botón anterior, permite cambiar el filtro en tiempo real para capturar la imagen a color o blanco y negro.

Falta mencionar la funcionalidad más importante de esta vista, el botón central redondo, utilizado para capturar la foto del examen. La siguiente imagen muestra la respuesta de la App al presionar el botón para capturar la imagen.

Ofrece una detección del examen mientras se está enfocando con la cámara, superponiendo un cuadro a modo de indicación para tomar la foto en el momento correcto.

Al capturar la imagen, el cuadrante rojo detectado es tomado como la imagen principal al que se le corrige la perspectiva y de esta forma permitir un mayor ángulo para hacer la fotografía.

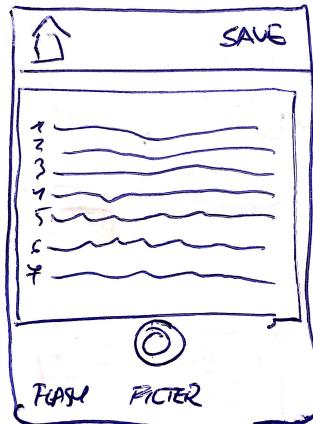


Ilustración 35 - Storyboard captura foto

Al capturar la imagen, esta es presentada ya con la corrección de perspectiva aplicada, para que el usuario decida si la selecciona o la descarta, si decide utilizarla, presionara en el botón Save que aparece al presentar la foto tomada. Si la imagen tomada no le parece de calidad, solo pulsando la imagen tomada podrá volver a tomar otra foto.

La siguiente vista es el resultado de la confirmación de la fotografía tomada al examen. El usuario es redirigido a la vista de creación de un nuevo examen.

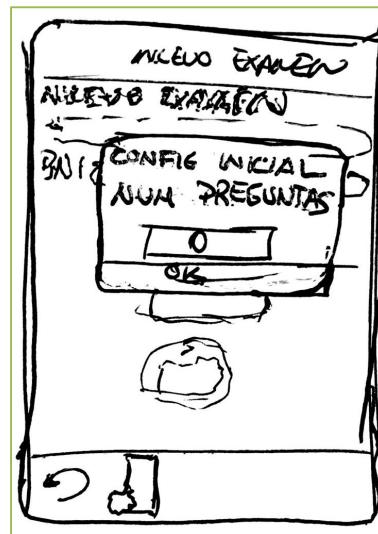


Ilustración 36 - Storyboard configuración inicial nuevo examen

Nada más presentar la vista, se muestra al usuario una alerta para la configuración inicial y obligatoria del examen. En este paso, el usuario no tiene más opción que introducir el número de preguntas.

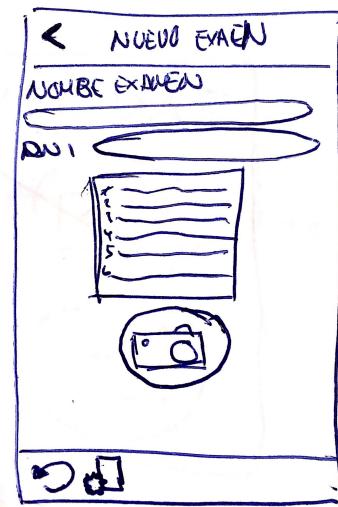


Ilustración 37 - Storyboard vista nuevo examen

Una vez completada la configuración inicial, se obtiene la presentación completa de la vista de creación. Cuenta con una pequeña vista de la imagen del examen y un menú inferior con la opción de rotarla, si es necesario.

El otro botón disponible en el menú inferior es el de procesar la imagen para detectar las respuestas del alumno.

A continuación, se muestra una imagen con las actualizaciones que afectan a la vista según avanza el proceso de corrección.

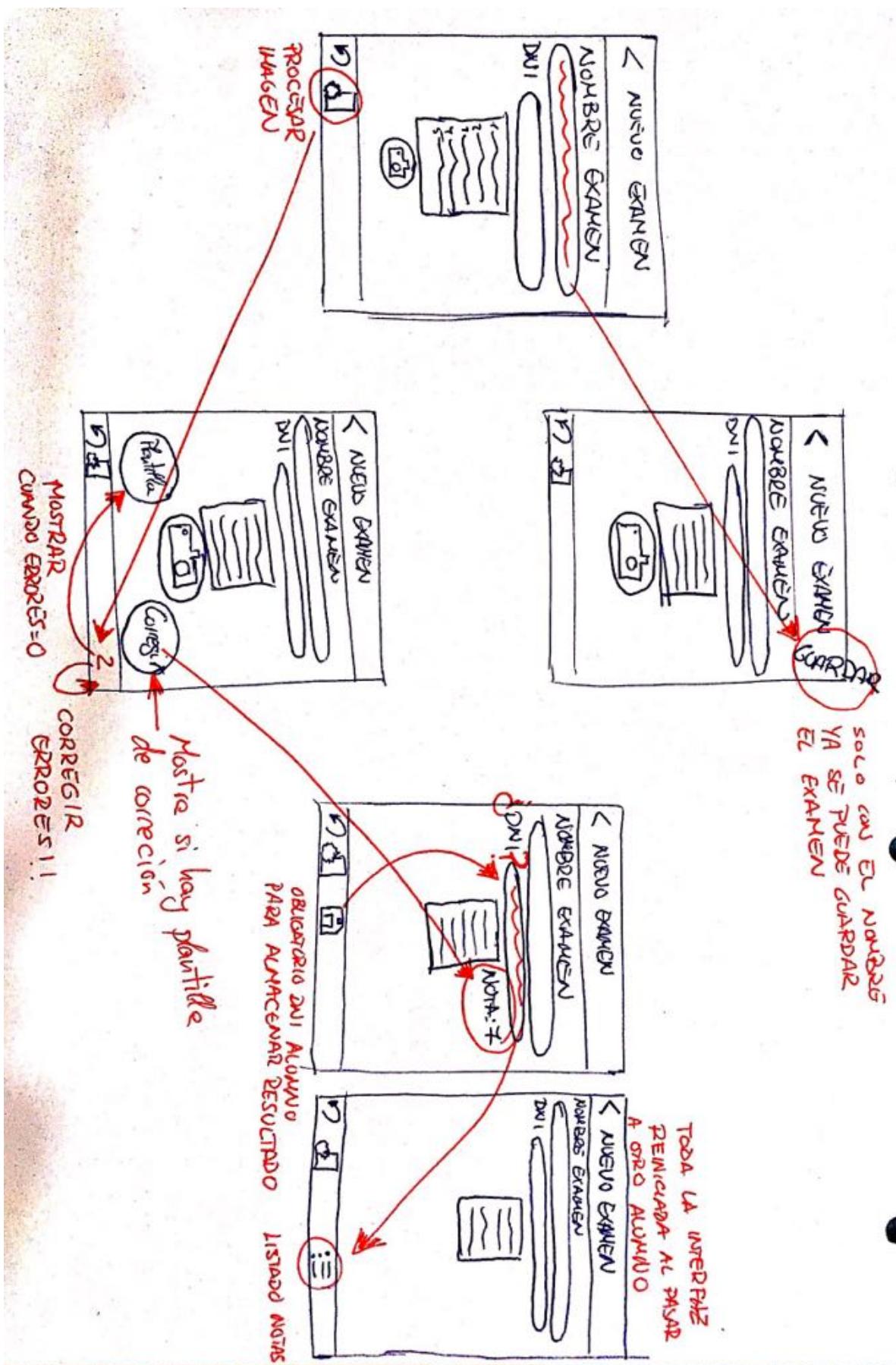


Ilustración 38 - Storyboard actualizaciones vista nuevo examen

La primera flecha hacia arriba indica que un examen se podrá guardar nada más que introduciendo un nombre para este.

La segunda flecha indica el procesamiento de la imagen y detección de las respuestas del alumno, en la segunda vista por abajo aparece una alerta si existen errores en la detección de las respuestas y el botón para establecer la plantilla en el momento que todas las respuestas están correctamente detectadas.

Si la plantilla ya está establecida, en el momento que todas las respuestas estén correctamente detectadas, el botón para obtener la nota se activa en el mismo momento.

Una vez calculada la nota, ya puede ser almacenada y pasar a la corrección del siguiente alumno, siempre y cuando el identificador del alumno este completado.

Finalizado el primer itinerario, se continuará por el listado de exámenes. La apariencia del listado será algo similar a la siguiente imagen.



Ilustración 39 - Storyboard listado exámenes

Desde esta vista, se puede acceder a la creación de un nuevo examen con el botón situado a la derecha del menú superior. Un examen puede ser seleccionado para ser modificado o continuar la corrección por donde se quedó. El gesto de deslizar el dedo de derecha a izquierda por encima de alguna de las celdas, dará la opción de eliminarla de la tabla y consigo, el examen que contiene.

A continuación, se muestra una imagen donde se representa las distintas interacciones con esta vista.

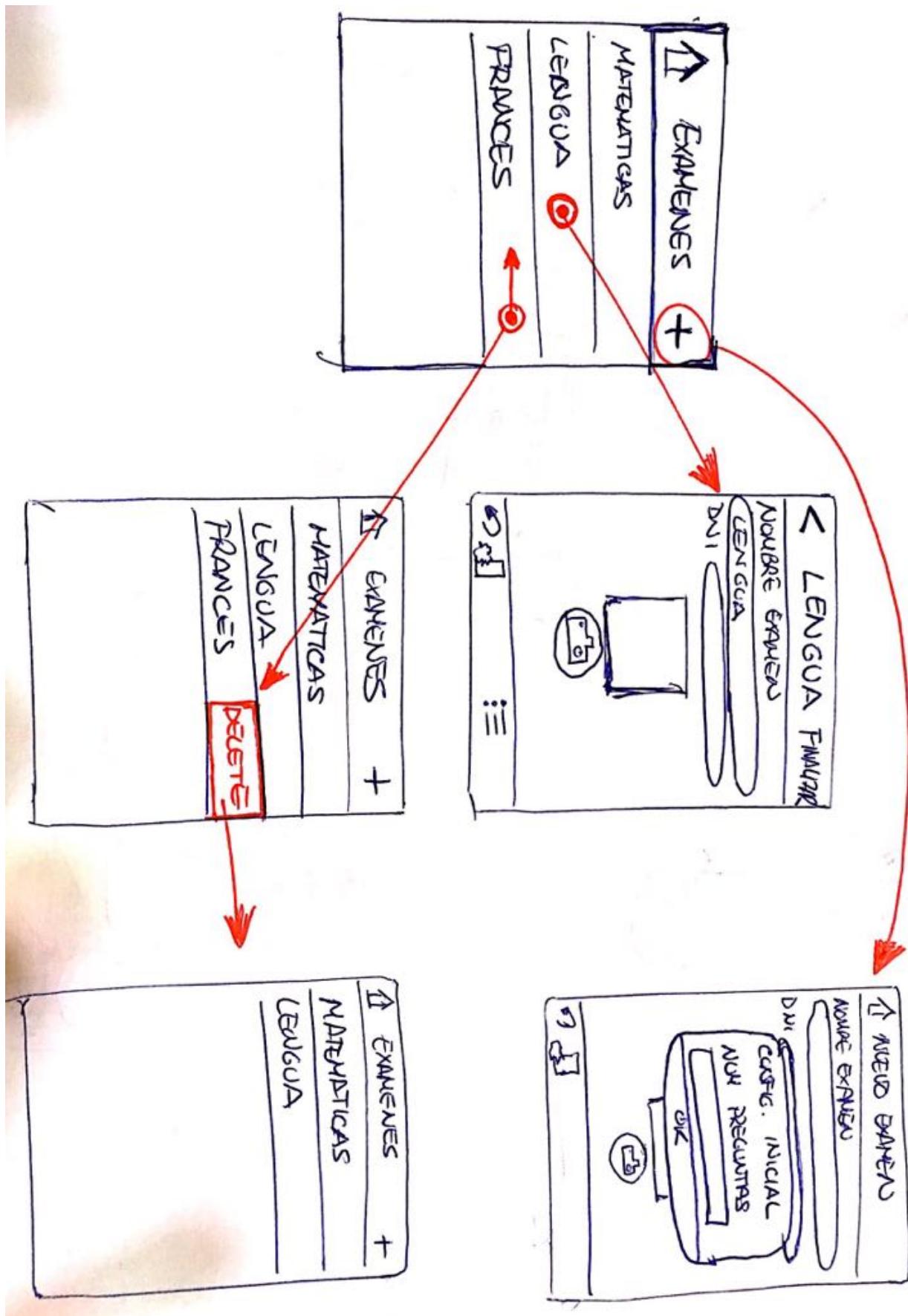


Ilustración 40 - Storyboard actualizaciones vista listado exámenes

Para finalizar este apartado de Storyboard se muestra la vista de lo que será los listados de calificaciones.

Como se puede observar es muy parecida al listado de exámenes, con la misma funcionalidad de arrastrar el dedo sobre la celda de la calificación que por algún motivo debe ser eliminada.

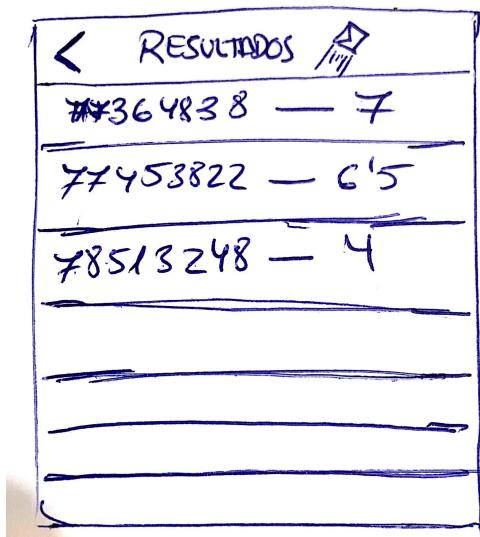


Ilustración 41 - Storyboard listado calificaciones

En el menú superior, a la derecha, hay un ícono de un sobre, este será utilizado para realizar la exportación de las calificaciones a un formato csv y compartirlas mediante las aplicaciones que lo permitan. Finalizado el primer itinerario, se continuará por el listado de exámenes. La apariencia del listado será algo similar a la siguiente imagen.

5. Implementación

En este capítulo hablaremos sobre los componentes utilizados en la implementación, explicando detalladamente el uso de cada uno, así como de su función en este proyecto.

5.1. Entorno de desarrollo

El entorno de desarrollo utilizado en la implementación de la aplicación es Xcode en su versión 8.3, desarrollado y propiedad de Apple. Con total integración

con Cocoa y Cocoa Touch frameworks. Es el entorno para el desarrollo de apps para Mac, iPhone, iPad, Apple Watch, and AppleTV más productivo que hay actualmente.



Ilustración 42 - XCode

Las ventanas dentro de XCode funcionan como espacios de trabajo o workspaces, y lo bueno es que podemos “personalizarlo” para que se adapte a la forma en la que queremos trabajar: puede contener un simple editor de texto, así como un editor gráfico, una librería multimedia, y un debugger, entre otras cosas.

El hecho de poder usar workspaces también permite que el usuario tenga la oportunidad de trabajar en varios proyectos a la vez, como sucede en la realidad. Gracias al uso de los espacios de trabajo podemos subdividir las partes de una aplicación o de lo que sea en que estemos trabajando de forma limpia y organizada, viendo todos los archivos disponibles en todos los proyectos, pero limitándonos a trabajar en uno solo.

5.2. Lenguajes de programación

5.2.1. Lenguaje programación C



Ilustración 43 - Lenguaje C

Lenguaje desarrollado en el año 1972 por Dennis Ritchie para UNIX. Es del tipo lenguaje estructurado. Incluye sentencias como if,else,for, do y while. Se caracteriza por tener una gran facilidad para escribir código compacto y sencillos a su vez.

5.2.2. C++



Ilustración 44 - C++

Desarrollado en el año 1983 por Bjarne Stroustrup, en los laboratorios AT&T. Es un derivado del lenguaje C. C++ intenta llevar a C hacia un paradigma de clases y objetos, con lo que pretende ayudar al programador en la tarea de abstracción de objetos de la realidad y gestionarlos más fácilmente en el código.

5.2.3. Objective-C



Ilustración 45 - Objective C

Es un lenguaje basado en C, desarrollado originalmente en 1980 y usado por Apple en sus sistemas operativos. Con soporte para la programación orientada a objetos. Hereda la sintaxis, primitivas y el control de flujo de las instrucciones de C y añade la sintaxis para la definición de clases y métodos.

5.2.4. Swift



Ilustración 46 - Swift

Es un lenguaje de programación potente e intuitivo para el desarrollo software en cualquiera de las plataformas de Apple.

Es el resultado del análisis de los distintos lenguajes de programación disponibles, combinado con décadas de experiencia en el desarrollo de software por parte de Apple.

Actualmente se encuentra en su versión 4 desarrollado bajo licencia Apache 2.0. Toda la información necesaria, acerca del desarrollo y código fuente se encuentra disponible en Swift.org. Dispone de una comunidad muy activa de desarrolladores en la que poder consultar dudas.

Las características resaltadas en su documentación son:

- Seguro: Las variables utilizadas son siempre inicializadas antes de su uso, se comprueba la posibilidad de desbordamiento para arrays y enteros y la memoria es manejada automáticamente.
- Rápido: Utiliza el compilador LLVM de alto rendimiento, transformando el código Swift en código nativo optimizado para el máximo aprovechamiento del hardware.
- Expresivo: Se beneficia de décadas de avances informáticos para ofrecer una sintaxis fácil de entender.

5.3. Detección de respuestas

En este apartado se explicará la implementación del proceso de reconocimiento de texto de la imagen y la detección de respuestas. Como se ha

especificado en el análisis de este TFG, la librería utilizada para el reconocimiento de texto ha sido Tesseract OCR y la de procesamiento visual OpenCV.

Consultando la documentación de Tesseract, disponible en el repositorio de GitHub, se recomienda hacer una serie de retoques a las imágenes antes de procesarlas con el motor OCR. A continuación, se explican algunas las más importantes:

- Reescalar: Aumento o reducción del tamaño físico de la imagen, alterando el número de pixeles que la componen.

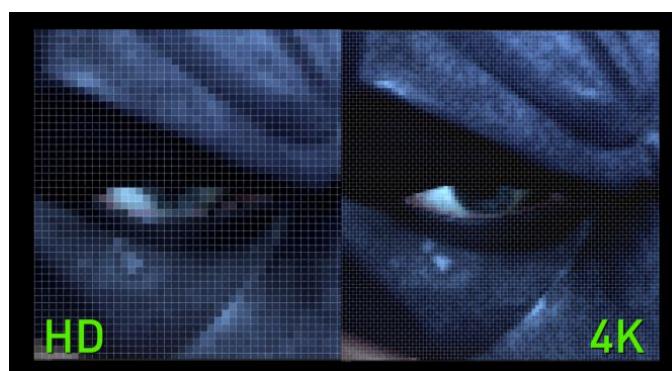


Ilustración 47 - Reescalado HD a 4K

- Binarizar: Consiste en un proceso de reducción de la información de una imagen digital a dos valores: 0 (negro) y 255(blanco).

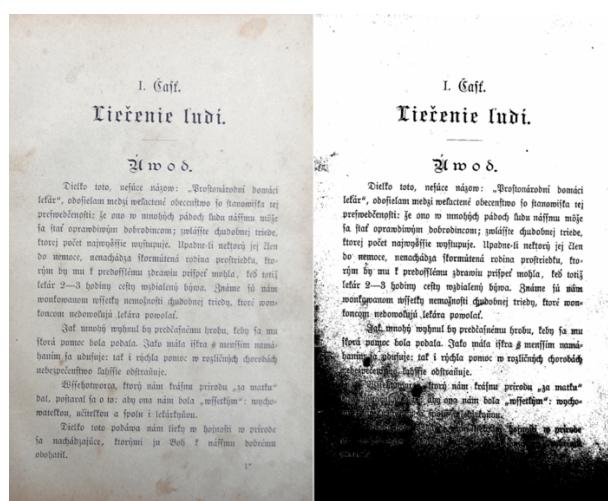


Ilustración 48 – Imagen blanco/negro

- Eliminación del ruido: Las cámaras producen una variación aleatoria del brillo o color de la imagen, dificultando el reconocimiento de texto.

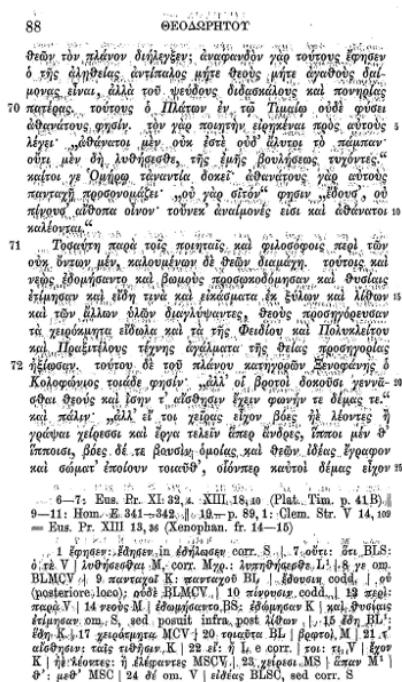


Ilustración 49 - Imagen con ruido digital

- Eliminar ángulo rotación texto: En muchas ocasiones al tomar una foto de un texto, este es capturado con algún ángulo de rotación y no totalmente en horizontal. Es necesario rectificar este desvío.

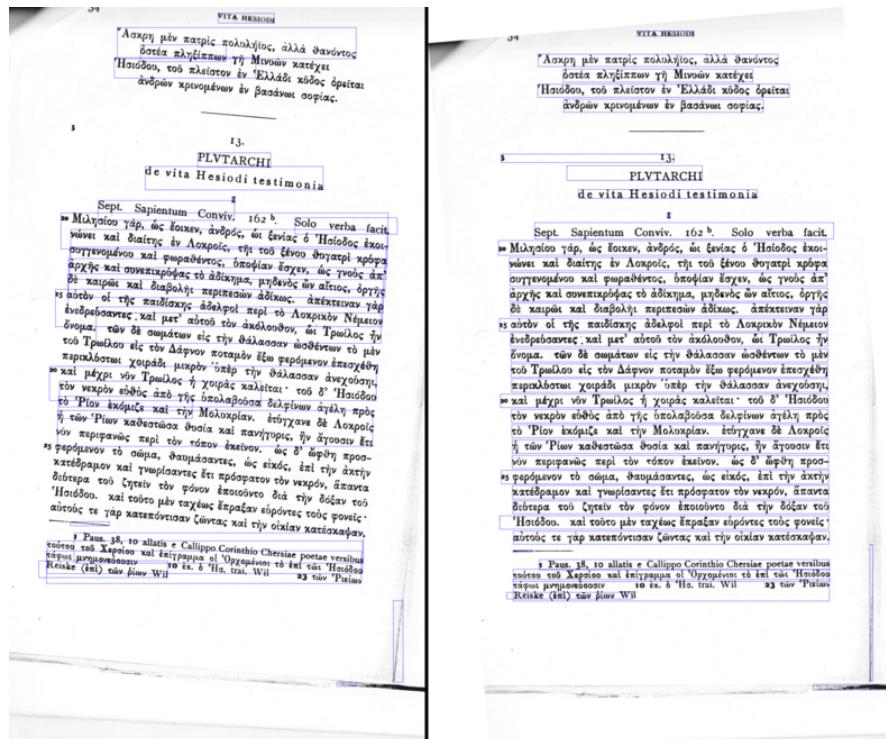


Ilustración 50 - Rectificación ángulo rotación texto

Después de estudiar los requerimientos del motor OCR. Se ha tomado la decisión de descomponer este proceso e implementarlo en 3 partes:

- La implementación de una cámara con reconocimiento de formas que realice el cambio de perspectiva al examen.
- Implementación del procesamiento visual y reconocimiento de texto.
- Detección y clasificación de respuestas para el cálculo de la calificación.

5.3.1. Cámara

Para la implementación de la cámara se ha utilizado IPDCameraViewController, una subclase de UIView implementada en Objective-C con vista previa de la cámara, detección de bordes en tiempo real y corrección de perspectiva para el examen fotografiado.

En la siguiente imagen, se realiza la detección del rectángulo y posteriormente la corrección de perspectiva de la imagen.

```
CIRectangleFeature *rectangleFeature = [self biggestRectangleInRectangles:[[self highAccuracyRectangleDetector]
    featuresInImage:enhancedImage]];

if (rectangleFeature)
{
    enhancedImage_ = [self correctPerspectiveForImage:enhancedImage, withFeatures:rectangleFeature];
}
```

Ilustración 51 - Detección rectángulo y cambio perspectiva

El primer método realiza la detección de la forma rectangular del examen con la ayuda de la clase CoreImage.

El segundo método procesa los rectángulos devueltos por el primer método para escoger el más grande.

```
- (CIDetector *)highAccuracyRectangleDetector
{
    static CIDetector *detector = nil;
    static dispatch_once_t onceToken;
    dispatch_once(&onceToken, ^{
        detector = [CIDetector detectorOfType:CIDetectorTypeRectangle context:nil options:@{CIDetectorAccuracy : CIDetectorAccuracyHigh}];
    });
    return detector;
}

-(CIRectangleFeature *)biggestRectangleInRectangles:(NSArray *)rectangles
{
    if (![rectangles count]) return nil;
    float halfPerimeterValue = 0;

    CIRectangleFeature *biggestRectangle = [rectangles firstObject];
    for (CIRectangleFeature *rect in rectangles)
    {
        CGPoint p1 = rect.topLeft;
        CGPoint p2 = rect.topRight;
        CGFloat width = hypotf(p1.x - p2.x, p1.y - p2.y);

        CGPoint p3 = rect.topLeft;
        CGPoint p4 = rect.bottomLeft;
        CGFloat height = hypotf(p3.x - p4.x, p3.y - p4.y);

        CGFloat currentHalfPerimeterValue = height + width;
        if (halfPerimeterValue < currentHalfPerimeterValue)
        {
            halfPerimeterValue = currentHalfPerimeterValue;
            biggestRectangle = rect;
        }
    }
    return biggestRectangle;
}
```

Ilustración 52 - Detección rectángulo

Este método recibe el rectángulo seleccionado por el método anterior, calcula los puntos de los extremos de la diagonal, el centro del rectángulo e identifica las coordenadas de las cuatro esquinas del rectángulo para después utilizarla en la corrección de la perspectiva.

```

- (CIRectangleFeature *)biggestRectangleInRectangles:(NSArray *)rectangles
{
    CIRectangleFeature *rectangleFeature = [self _biggestRectangleInRectangles:rectangles];
    if (!rectangleFeature) return nil;

    NSArray *points = @[[NSValue valueWithCGPoint:rectangleFeature.topLeft],[NSValue valueWithCGPoint:rectangleFeature.topRight]
        ,[NSValue valueWithCGPoint:rectangleFeature.bottomLeft],[NSValue valueWithCGPoint:rectangleFeature.bottomRight]];

    CGPoint min = [points[0] CGPointValue];
    CGPoint max = min;
    for (NSValue *value in points)
    {
        CGPoint point = [value CGPointValue];
        min.x = fminf(point.x, min.x);
        min.y = fminf(point.y, min.y);
        max.x = fmaxf(point.x, max.x);
        max.y = fmaxf(point.y, max.y);
    }

    CGPoint center =
    {
        0.5f * (min.x + max.x),
        0.5f * (min.y + max.y),
    };

    NSNumber *(^angleFromPoint)(id) = ^(NSValue *value)
    {
        CGPoint point = [value CGPointValue];
        CGFloat theta = atan2f(point.y - center.y, point.x - center.x);
        CGFloat angle = fmodf(M_PI - M_PI_4 + theta, 2 * M_PI);
        return @angle;
    };

    NSArray *sortedPoints = [points sortedArrayUsingComparator:^NSComparisonResult(id a, id b)
    {
        return [angleFromPoint(a) compare:angleFromPoint(b)];
    }];

    IPDFRectangleFeature *rectangleFeatureMutable = [IPDFRectangleFeature new];
    rectangleFeatureMutable.topLeft = [sortedPoints[3] CGPointValue];
    rectangleFeatureMutable.topRight = [sortedPoints[2] CGPointValue];
    rectangleFeatureMutable.bottomRight = [sortedPoints[1] CGPointValue];
    rectangleFeatureMutable.bottomLeft = [sortedPoints[0] CGPointValue];

    return (id)rectangleFeatureMutable;
}

```

Ilustración 53 - Identificación cuatro puntos

Por último, queda la corrección de perspectiva de la foto, implementada en la siguiente imagen. El método recibe las coordenadas del rectángulo y crea una imagen con ellas. Con este método se asegura una dirección del texto aceptable para el reconocimiento de texto.

```

- (CIImage *)correctPerspectiveForImage:(CIImage *)image withFeatures:(CIRectangleFeature *)rectangleFeature
{
    NSMutableDictionary *rectangleCoordinates = [NSMutableDictionary new];
    rectangleCoordinates[@"inputTopLeft"] = [CIVector vectorWithCGPoint:rectangleFeature.topLeft];
    rectangleCoordinates[@"inputTopRight"] = [CIVector vectorWithCGPoint:rectangleFeature.topRight];
    rectangleCoordinates[@"inputBottomLeft"] = [CIVector vectorWithCGPoint:rectangleFeature.bottomLeft];
    rectangleCoordinates[@"inputBottomRight"] = [CIVector vectorWithCGPoint:rectangleFeature.bottomRight];
    return [image imageByApplyingFilter:@"CIPerspectiveCorrection" withInputParameters:rectangleCoordinates];
}

```

Ilustración 54 - Corrección perspectiva

El ultimo método que comentar en este subapartado es el encargado de marcar el rectángulo detectado al usuario a tiempo real en la previa de la cámara. Este método, se ejecuta cíclicamente comprobando si hay coordenadas válidas para el rectángulo.

```

- (CIIImage *)drawHighlightOverlayForPoints:(CIIImage *)image topLeft:(CGPoint)topLeft topRight:(CGPoint)topRight bottomLeft:(CGPoint)bottomLeft bottomRight:(CGPoint)bottomRight
{
    CIIImage *overlay = [CIIImage imageWithColor:[CIColor colorWithRed:0.2 green:0 blue:0.9 alpha:0.5]];
    overlay = [overlay imageByCroppingToRect:image.extent];
    overlay = [overlay imageByApplyingFilter:@"CIPerspectiveTransformWithExtent" withInputParameters:@[@"inputExtent":[CIVector vectorWithCGRect:image.extent], @"inputTopLeft": [CIVector vectorWithCGPoint:topLeft], @"inputTopRight": [CIVector vectorWithCGPoint:topRight], @"inputBottomLeft": [CIVector vectorWithCGPoint:bottomLeft], @"inputBottomRight": [CIVector vectorWithCGPoint:bottomRight]}];
}

return [overlay imageByCompositingOverImage:image];
}

```

Ilustración 55 - Indicador detección rectángula en cámara

5.3.2. Procesamiento visual y reconocimiento de texto

Para este proceso, se eligió como mejor opción, utilizar la librería OpenCV. El siguiente método recibe la imagen tomada en el paso anterior, la binariza, selecciona los diccionarios y es procesada con el motor OCR.

```

@IBAction func processImage(_ sender: UIBarButtonItem) {
    if let tesseract = G8Tesseract(language: "asc+eng") {
        tesseract.delegate = self
        print("Starting the processing of image...")
        OpenCVWrapper.show(withStatus: "Procesando la imagen...")
        DispatchQueue.global().async() {
            // Background Thread
            self.image.image = OpenCVWrapper.scale2grayandcleannoise(self.image.image)
            //self.image.image = OpenCVWrapper.changeScaleGrayAndCleanNoise(self.image.image)
            print("Binarization and clean noise finish...")

            tesseract.image = self.image.image
            tesseract.recognize()

            DispatchQueue.main.async {
                self.image.image = tesseract.image
                self.textFieldRecognized.text = tesseract.recognizedText

                //Analyze text recognized by TesseractOCR
                self.processTextRecognized()
                // Run UI Updates
                var errorPercentage = ( self.errors.count * 100 ) / self.nPreguntas
                if(errorPercentage == 0) {
                    OpenCVWrapper.dismissProgressRing("Escaneo finalizado sin errores!")
                } else if(errorPercentage < 35) {
                    OpenCVWrapper.showError(withStatus: "Detectados algunos errores, puedes volver a captura una fotografía e intentare solucionarlos o corregirlos desde la barra inferior de la pantalla, el icono rojo de la derecha.")
                    OpenCVWrapper.dismissWithDelay()
                } else if(errorPercentage > 35) {
                    OpenCVWrapper.showError(withStatus: "Demasiados errores, ¿Quedan mas hojas?, si no es asi, deberias volver a repetir la captura del examen.")
                    OpenCVWrapper.dismissWithDelay()
                }
            }
        }
    }
}

```

Ilustración 56 - Función processImage()

La función anterior se ayuda de métodos de métodos implementados en C para la binarización de la imagen. A continuación, se muestra el código que lo realiza.

```
+(UIImage*)scale2grayandcleannoise:(UIImage *)inputImage {
    //UIImage to cv::mat
    cv::Mat input = [inputImage CVMat];
    cv::cvtColor(input, input, CV_BGR2GRAY);

    //Process image...
    cv::Mat result;
    cv::Size size = input.size();
    cv::bitwise_not(input, input);

    return [UIImage imageWithCVMat:result];
}
```

Ilustración 57 - Función imagen escala grises

En este mismo proceso se incluyó la limpieza de ruido de la imagen. Después de varias pruebas en distintos escenarios, se comprobó que el tiempo requerido para dicha operación y el resultado obtenido, no merecían suficientemente la pena como para incluirlo. La implementación de este método es la siguiente:

```
+(UIImage*)changeScaleGrayAndCleanNoise:(UIImage *)inputImage {
    //UIImage to cv::mat
    cv::Mat input = [inputImage CVMat];
    cv::cvtColor(input, input, CV_BGR2GRAY);
    cv::Mat result;
    cv::bitwise_not(input, result);
    cv::fastNIMeansDenoising(result, result);
    return [UIImage imageWithCVMat:result];
}
```

Ilustración 58 - Función imagen escala de grises con limpieza de ruido

Una vez realizada la modificación de la imagen a escala de grises, se procesa con el motor OCR y se extrae el texto del examen, finalizando esta parte de la implementación.

5.3.3. Detección y clasificación de las respuestas

se ha la inclusión rápida y sencilla de framework idepara la usce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna.

de Edición para la usce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna.

6. Conclusiones y posibles mejoras

suarios de Edición para la usce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna.

Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin pharetra nonummy pede. Mauris et orci.

Aenean nec lorem. In porttitor. Donec laoreet nonummy augue.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna. Nunc viverra imperdiet enim. Fusce est.

Tabla de ejemplo:

| A | B | C |
|----------|----------|----------|
| habitant | | |
| | habitant | |
| | | senectus |

Tabla 6

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa.

Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna. Nunc viverra imperdiet enim.

Fusce est. Vivamus a tellus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin pharetra nonummy pede.

Mauris et orci. Aenean nec lorem.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna. Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin pharetra nonummy pede. Mauris et orci. Aenean nec lorem. In porttitor. Donec laoreet nonummy augue.

(Velasco, 2009)

Suspendisse dui purus, scelerisque at, vulputate vitae, pretium mattis, nunc. Mauris eget neque at sem venenatis eleifend. Ut nonummy. Fusce aliquet pede non pede. Suspendisse dapibus lorem pellentesque magna. Integer nulla.

Fórmula de la velocidad:

$$\lim_{X \rightarrow X_0} \frac{X - X_0}{T - T_0}$$

Donde:

X = posición final

Xo= posición inicial

(Fórmula 2.1)

Bibliografía

Benkler, Y. (2012). *El Pingüino y el Leviatán: Por qué la cooperación es nuestra arma más valiosa para mejorar el bienestar de la sociedad.*

Velasco, J. G. (2009). *Energías renovables*. Editorial Reverte.