



Título del Proyecto:

Corrección de test con realidad aumentada.

Autor:

Villar Boronat, Raúl

Director:

Tomás Gironés, Jesús

TESINA PARA LA OBTENCIÓN DEL TÍTULO DE:

Diploma de Especialización en Desarrollo de Aplicaciones para Android

Septiembre del 2017



Contenido

| Introducción |
|---------------------------------|
| Descripción del problema3 |
| Objetivos |
| Motivación4 |
| Tecnologías utilizadas5 |
| Arquitectura de la aplicación5 |
| Esquema del diseño5 |
| Procesamiento de la imagen8 |
| Modelo de datos9 |
| Vistas |
| Cálculo de la nota |
| Conclusiones |
| Anexos |
| Listado de fuentes entregadas14 |
| Manual de usuario14 |
| Instalación de OpenCV Manager15 |
| Modo creación16 |
| Modo corrección17 |
| Modo actualización |



Introducción

Descripción del problema

La realidad aumentada es el término que se utiliza para definir la visión del mundo real a través de un dispositivo tecnológico de manera que el resultado es una combinación de elementos captados por la cámara y elementos virtuales logrando crear de esta manera una realidad enriquecida.

Entre las posibilidades que ofrece la realidad aumentada está la de permitir la corrección automática de un examen o formulario tipo test como se hizo en el proyecto de tesina "Reconocimiento Formularios", realizado por **Alejandro Caballero Platas**, que consistía en el reconocimiento óptico de formularios para la corrección de exámenes.

El presente proyecto incorpora alguna de las ideas utilizadas en el mencionado proyecto, como el algoritmo para detectar cuando una casilla está marcada; reformula otras, como la detección de cuadrados y añade nuevas funcionalidades como el cálculo de la nota y la utilización de una base de datos no relacional con la tecnología Firebase.

Objetivos

El objetivo consiste en un realizar un sistema de realidad aumentada que permita la corrección automática de un examen o formulario tipo test.

Detalles:

- Fase 1: El profesor crea el examen en Word. Ha de introducir cuadros para que el alumno introduzca una cruz (en una segunda fase podría ser también un dígito del 0 al 9)
- Fase 2: El profesor imprime el examen, lo completa correctamente y utiliza el programa para escanear las posiciones de los cuadros y el valor correcto. El resultado de esta fase es un fichero de respuestas, con una lista de cuadros con su posición y valor correcto. Este fichero es almacenado en un servidor Web.
- Fase 3: El alumno completa el examen en papel y utiliza el programa en modo corrección. Primero se descarga el fichero del servidor y luego se pasa a un



modo de realidad aumentada similar al mostrado en la captura.



La aplicación deberá contar con un interfaz de usuario adecuado y muy sencillo.

- Dos pestañas: "test disponibles" y "mis test". En la primera pestaña se muestran todos los test activos con independencia del usuario que los haya creado. En la segunda pestaña se muestran sólo los test creados por el usuario que se haya dado de alta.
- El usuario que se usa es el de Google. En caso de que no se pueda acceder, se considerará que es el usuario anónimo.
- Se entra directamente a "test disponibles" y se muestra un listado con todos los exámenes activos.
- Si selecciona un examen, mediante una pulsación corta, de cualquiera de las dos pestañas, se entra en modo corrección activándose la cámara.
- Otra pestaña, "mis test", muestra solo los test que ha creado este usuario. Hay un botón "+" para crear un nuevo examen. Mediante una pulsación larga, se habilita un menú que permite activar exámenes, editarlos o borrarlos. Este menú también incluye la opción de volver a escanear las preguntas y respuestas correctas.
- **Opcional**: Obtención de una nota que podría ser visualizada en un círculo en la parte superior derecha del examen, simulando que el profesor a corregido el examen con un rotulador. Sería necesario editar el fichero de respuestas generado en la fase dos para añadir información sobre el valor de cada respuesta.
- **Opcional**: Permitir la lectura de dígitos.
- **Opcional**: La información se almacena en Firebase.

Motivación

Aprender y enseñar lo aprendido son dos motivaciones internas que por si solas me parecen suficiente razón para realizar un proyecto tecnológico que tiene vinculaciones con el mundo educativo, concretamente, en la autoevaluación de conocimientos.

Además, debo añadir que los temas dedicados a lo largo del diploma a la implementación de la visión artificial, explicados de mano de **Antonio Albiol**, me parecieron muy interesantes y con un gran potencial para ser aplicado en dispositivos móviles.

Este proyecto pretende, pues, poner en práctica una diversidad de conocimientos vistos a lo largo del diploma y ponerlos al servicio de un fin educativo.



Tecnologías utilizadas

Para la realización de este proyecto se ha utilizado la plataforma de desarrollo de aplicaciones móviles de Android SDK, análisis de imagen en Android usado OpenCV y almacenamiento en tiempo real con Firebase. A continuación, se hace una breve reseña de las tecnologías mencionadas:

- Android: Sistema operativo pensado para dispositivos móviles basado en Linux, un núcleo de sistema operativo, libre, gratuito y multiplataforma. Como entorno de desarrollo he usado Android SDK.
- **OpenCV**: Librería de análisis de imágenes que se apoya en la aplicación OpenCV Manager que debe de estar instalada en el dispositivo para que funcione la aplicación.
- **Firebase**: Plataforma de desarrollo móvil que permite desarrollar aplicaciones multiplataforma. Cuenta con varias funcionalidades entre las cuales he usado la base de datos en tiempo real que almacena la información como objetos JSON, que es un formato de texto ligero para el intercambio de datos.

Arquitectura de la aplicación

Esquema del diseño



Se han realizado tres actividades:

 ListadoFichaTestActivity: Es la actividad principal que permite visualizar los test activos de todos los usuarios y los test propios. Desde esta actividad se puede acceder a las otras dos actividades; con una pulsación corta se entra en modo corrección en CamaraActivity; con una pulsación larga se habilita el menú que permitirá editar la información del test mediante FichaTestActivity.



- **CamaraActivity**: Es la actividad donde se combina la información que proviene de la cámara junto el resultado del procesamiento de imagen que se realiza a través de las librerías de OpenCV. Esta actividad permite trabajar en tres modos distintos: creación, actualización y corrección. En caso de que se cancele la creación, la actualización o la corrección del test, se vuelve a ListadoFichaTestActivity. En caso de que se quiera guardar el resultado de la creación o de la actualización se va a FichaTestActivity.
- **FichaTestActivity**: Esta actividad permite editar el título y la descripción del test. Además, visualiza información complementaria como el número de preguntas, el autor y la fecha de creación. También es posible activar o desactivar los test desde esta actividad. Cualquier cambio deberá ser grabado antes de salir de esta actividad.

Otras clases que forman parte de la aplicación son:

- **Aplicación**: Es una clase que extiende la clase Application y que es usada como repositorio de variables globales, como los permisos solicitados en tiempo de ejecución:
 - o opencvManagerInstalado: indica si OpenCV Manager está instalado.
 - o *permisoCamara*: indica si el usuario ha concedido permiso para usar la cámara.
 - *permisoObtenerCuentas*: indica si el usuario ha concedido permiso para acceder a las cuentas.
 - *permisoAlmacenamiento*: indica si el usuario ha concedido permiso para acceder al almacenamiento del dispositivo.
- AdaptadorFichaTest: Es la clase que se usa en el RecyclerView de la actividad ListadoFichaTestActivity para mostrar los test disponibles.
- **FichaTestViewHolder**: Es una clase interna de AdaptadorFichaTest que extiende a RecyclerView.ViewHolder.
- **Procesador**: Es la clase encargada de realizar el procesamiento de imagen utilizando las librerías de OpenCV.

Respecto a las decisiones de diseño, cabe indicar que he escogido el modo de visualización horizontal para toda la aplicación porque la imagen que me devuelve la clase de OpenCV devuelve los frames de la cámara en formato horizontal. En varios foros se comenta la posibilidad de hacer un procesamiento de la imagen para rotar y hacerla coincidir con las dimensiones de la pantalla completa. Esta opción, sin embargo, conlleva un sobrecoste computacional.

La aplicación se ha realizado de manera que detecte cuando el OpenCV Manager no esté instalado y, en ese caso, solicita al usuario que lo instale. Los permisos que se piden son los necesarios para llevar a cabo las funcionalidades que lo precisan. Sin embargo, en la medida de lo posible, la aplicación intenta ser lo máximo operativa, aunque haya un o más permisos que no se concedan. Por ejemplo, si el usuario no ofrece permiso para acceder a las cuentas, se



considerará que corresponde al usuario "anónimo". Si no se ofrece permiso para acceder al almacenamiento, entonces no se permitirá almacenar el resultado de la corrección, pero se podrá continuar usando el resto de funciones de la aplicación. Si no se permite acceso a la cámara o no está el OpenCV Manager instalado, no se podrá utilizar el modo de realidad aumentada, pero se podrá seguir accediendo al listado de test.

La decisión más importante ha sido la de trabajar con la hoja de resultados en lugar de con el enunciado. La aplicación espera encontrar las diferentes opciones de una misma pregunta en la misma línea horizontal. Tanto el número de preguntas, como el número de opciones de cada una de las preguntas, puede ser variable y la aplicación se adapta.

La aplicación considera que todas las preguntas valen lo mismo y admite que una pregunta tenga más de una opción válida. La fórmula que se ha usado para calcular la nota es la habitual para este tipo de exámenes. Las respuestas correctas, suman; las respuestas incorrectas, restan y las que quedan en blanco, ni suman ni restan. El valor de bonificación o penalización de cada respuesta varía en función del número de opciones correctas que cada pregunta tiene en relación al número de opciones posibles.

En referencia a la utilización de las librerías de OpenCV, he optado por usar la carga dinámica que se basa en las librerías suministradas por otra aplicación: OpenCV Manager. He realizado otro tipo de pruebas que consisten en incorporar las librerías nativas en la misma aplicación. De este modo, la aplicación se carga estáticamente sin necesidad del OpenCV Manager. La razón por la cual he descartado esta posibilidad es que OpenCV sólo recomienda esta opción en entornos de desarrollo pero no en producción. Al seguir esta recomendación conseguimos un fichero apk de menor tamaño y nos podemos beneficiar de las actualizaciones realizadas en las librerías de OpenCV a través del Play Store.

La decisión de utilizar Firebase como base de datos en tiempo real se explica no sólo por ser uno de los objetivos opcionales que se planteaban inicialmente en este proyecto, sino porque su implementación resultaba más ágil y menos problemática que usar otras soluciones como un servidor en el que haya que implementar la lógica necesaria para llevar a cabo

Para aumentar la estabilidad de los resultados obtenidos cuando se está trabajando en modo de realidad aumentada, he optado por incluir un mecanismo que comprueba que como mínimo se ha realizado tres lecturas consecutivas idénticas antes de mostrarlo al usuario. Como lectura idéntica compruebo que coincidan el número de preguntas, las opciones de cada pregunta y las respuestas realizadas.

En un primer momento trabajé con la idea de buscar el reconocimiento de carácteres pero ante el número de casuísticas encontrado opté por reutilizar la idea implementada en la anterior versión de este proyecto que consiste en contar como mínimo tres segmentos dentro de la casilla, que corresponderían a los espacios que se originan al escribir una cruz dentro de una casilla.



Procesamiento de la imagen

Los pasos principales que sigo para procesar la imagen son los siguientes:

- 1. Binarización adaptativa de la entrada en gris con un tamaño de bloque de 15 píxeles y un contraste de 40 respecto a la media *(Imgproc.adaptiveThreshold)*.
- 2. Filtro morfológico para hacer los objetos oscuros más gruesos mediante una erosión de 5 x 5 píxeles (*Imgproc.erode*).
- 3. Segmentar siguiendo una serie de requisitos que detallo más adelante.
- 4. Ordenar segmentos de arriba abajo y de izquierda a derecha.
- 5. Eliminar segmentos dispares; aquellos cuya área sea más de un 50% diferente al área más frecuente.
- 6. Restaurar los segmentos con la entrada en gris.
- 7. Rellenar un objeto tipo Test con las preguntas y opciones correspondientes.
- 8. Convertir a color.
- 9. Comprobar si es una lectura estable; es decir, que se repita la misma lectura, al menos, 3 veces consecutivamente.

Para segmentar, es decir, para encontrar las casillas, sigo los siguientes pasos:

- 1. Localizar los contornos que son externos pero que contienen, al menos, un contorno interior (*Imgproc.findContours*).
- 2. Descartar los contornos cuya anchura o altura no estén entre 15 y 60 píxeles.
- 3. Descartar los contornos con ratios de aspecto que no estén entre 0,75 y 1,25.
- 4. Descartar los contornos que no tengan cuatro vértices (Imgproc.approxPolyDP).
- 5. Descartar los contornos que no tengan forma cuadrada (*Imgproc.matchShapes*).
- 6. Conservar los contornos resultantes.

Para rellenar el test, hay dos procesos principales que se realizan simultáneamente:

- Agrupar las casillas en preguntas en función de la distancia que hay entre ellas.
- Detectar las casillas que están marcadas.

Los pasos principales para detectar si una casilla está marcada son:

- 1. Seleccionar un área un 20% menor para reducir los problemas que se derivan cuando la cruz no está completa, es decir, cuando no va de extremo a extremo de la casilla.
- 2. Binarización Otsu (Imgproc.adaptiveThreshold).
- 3. Localizar contornos.
- Si hay al menos 3 contornos, se considera la casilla marcada. En realidad, deberían ser 4 contornos, ya que una cruz dentro de una casilla origina 4 segmentos, pero así damos un margen de seguridad.



Modelo de datos

He optado por usar una estructura de datos para recoger la información obtenida a través del procesamiento de imagen que está compuesto por:

- **Test**: Contiene una lista de preguntas, el número de preguntas y la codificación del test.
- **Pregunta**: Contiene una lista de opciones.
- **Opción**: Contiene un booleano que indica si la opción ha sido marcada, así como las coordenadas de un rectángulo que corresponde a la casilla.

Para manejar los datos que hacen referencia a los test se utiliza una clase específica:

• FichaTest: Contiene los datos que serán almacenados en la base de datos de Firebase.

Para almacenar la información en Firebase he usado una estructura con un nodo principal llamado tests desde el cual cuelgan el resto de nodos correspondientes a cada test. Por cada test almaceno la siguiente información:

- Activo: variable que permite saber si el test está activo o no. En el caso de que su valor sea cierto, estará disponible para todos los usuarios. En caso contrario, sólo estará disponible para el creador indicado en el campo creador.
- **Creador**: almacena el usuario que creó el test. He usado la cuenta de Gmail. Para obtener acceso a esta cuenta se le pide al usuario acceso a las cuentas. En caso de que se deniegue el acceso, figurará el usuario "anónimo".
- **Descripción**: añade información al título del test.
- Fecha creación: un valor número recoge el momento en que fue creado el test.
- Numero de preguntas: recoge el número de preguntas que conforman el test.
- Solución: recoge en un mismo campo la estructura del test y las opciones que son correctas. Cada opción se codifica con una F (opción no marcada) o con una T (opción marcada). Para separar las opciones de una pregunta de las opciones de la siguiente pregunta se usa un asterisco.
- Título: Indica el título del test.



tester-36804

tests

- -Kubi_ugat2Xb4_VOmSW
 - activo: true
 - ----- creador: "raulvibo
 - descripcion: "Planetas y satélites principale
 - fechaCreacion: 150605255400
 - numeroPreguntas: 4
 - solucion: "FTFF*FTFF*FFTF*FFFT
 - titulo: "El sistema solar
- -KubqlvEMfgiGvo8MPZa
- -KueFmJTFile0UF0rhgM
- -Kug1uY5u5uNqC3DtMN6
- -Kuj_Yq3iMu96teP011s
- -KujzOKvdZc_seLL0ZMI

Vistas

La aplicación está compuesta por 3 pantallas principales:



- Listado de test: Punto de inicio de la aplicación. Funciona como el home de la misma, a partir de la cual se puede navegar a las otras dos pantallas.
- **Realidad aumentada**: Se muestra el resultado de fusionar la entrada de la cámara junto con la información obtenida mediante el procesamiento de la imagen. Permite



navegar a la edición de test para guardar o actualizar la estructura de preguntas y respuestas de un test.

• Edición de test: Muestra la ficha de un test y permite modificar el título, la descripción y el estado de activación del mismo.

Cálculo de la nota

Una vez seleccionado un test y escaneado las respuestas, se está en disposición de corregirlo. Esto permitirá ver las preguntas que han sido contestadas correctamente, los errores y también las respuestas correctas. Además, se ofrece una nota cuyo cálculo se realiza conforme a la siguiente fórmula:

$$Nota = \sum (Vp) * \left(\frac{Aciertos}{Opciones v lidas}\right) - (Vp) * \left(\frac{Errores}{Opciones - 1}\right)$$

donde:

$$Vp = \left(\frac{Puntuación máxima}{Número de preguntas}\right)$$

- **Opciones**: Número de alternativas de respuesta: 2 si hay a y b; 3 si hay a,b y c, etc.
- **Opciones válidas**: En el caso que la pregunta sea de respuesta única, este valor será 1. En el caso de que la pregunta sea de respuesta múltiple, este valor será igual al número de respuestas marcadas como válidas.
- Aciertos: Número de respuestas correctas.
- Errores: Número de respuestas incorrectas.

Esto significa que los errores restan y se acumulan de una pregunta a otra. En el caso que el resultado de esta fórmula fuese negativo, la nota final se fija a cero.

Para obtener la puntuación máxima en una pregunta es necesario acertar plenamente todas las opciones válidas, contemplándose la posibilidad de obtener una puntuación parcial para las respuestas parcialmente correctas.

La penalización para cada pregunta se calcula en función del número de opciones o alternativas posibles de respuestas. La justificación de penalizar los errores está en contrarrestar la puntuación que se pueda obtener por mero azar. Esto explica porque la penalización disminuye a medida que las opciones aumentan, puesto que, a más opciones, menor es la probabilidad de acertar por azar.



| Opciones | Penalización por error | | |
|----------|------------------------|--|--|
| 2 | 100% | | |
| 3 | 50% | | |
| 4 | 33,33% | | |
| 5 | 25% | | |

Dejarse una pregunta en blanco no suma, pero tampoco resta.

Conclusiones

En líneas generales, y con las modificaciones apuntadas, se puede decir que la aplicación cumple con la especificación de realizar un sistema de realidad aumentada que permita la corrección automática de un examen o formulario tipo test. La aplicación permite capturar la información de una plantilla de test, almacenarla en una base de datos y recuperarla cuando se desee para corregir otras plantillas que se correspondan al mismo test.

Tanto a la hora de capturar un test como a la hora de mostrar la corrección, se aporta al usuario una información gráfica que le permite validar visualmente si la captura es correcta y ver qué preguntas han sido respondidas correctamente, cuáles no, y cuáles eran las respuestas correctas en las preguntas que se han dejado por contestar o que no se han contestado correctamente.

Los resultados obtenidos, en cuanto a velocidad para detectar las casillas y los valores de las respuestas, ha ido mejorando a lo largo que se ha ido realizando el proyecto. En este momento, es posible capturar un examen tipo test de 30 preguntas, agrupados en 3 columnas, en apenas unos instantes. Cierto es, que las prestaciones de la aplicación pueden variar por diversas razones, entre las cuales, la inclinación del plano de la cámara con respecto a la hoja que se quiere escanear, el número de opciones, la iluminación o las características de la cámara del dispositivo.

Como líneas de futuro indico las siguientes posibilidades:

- Ajuste del valor de cada pregunta del test. Es decir, que aplicación permita trabajar con preguntas que valgan diferente y que este valor se pueda modificar mediante la misma aplicación.
- Reconocimiento de caracteres que permita identificar dígitos.
- Habilitar el modo vertical.
- Añadir más información en la ficha de los test, como, por ejemplo, el número de preguntas de respuesta única y el número de preguntas de respuesta múltiple.



- Permitir configurar si las preguntas incorrectas penalizan.
- Permitir modificar el modo de funcionamiento de la cámara: modo macro, automático, etc.
- Permitir escoger el usuario que se quiera, en el caso de que haya más de una cuenta de Google.
- Permitir iniciar sesión a través de usuarios de redes sociales.
- La aplicación trabaja con la resolución máxima que puede ofrecer la cámara. Puede ser interesante habilitar la posibilidad de ajustar la resolución para aumentar la velocidad de captura y procesamiento de la imagen.
- Recoger en la BBDD el resultado de las correcciones.



Anexos

Listado de fuentes entregadas

- Código fuente: VillarR_Proyecto_Final_Tester.zip
- Ejecutable: VillarR_Proyecto_Final_Tester.apk
- Vídeo tutorial de la aplicación: <u>https://youtu.be/wji8p3plMvg</u>
- Diapositivas de la presentación (PDF): VillarR_Proyecto_Final_Tester.pptx
- Diapositivas de la presentación (PDF): VillarR_Proyecto_Final_Tester.pdf

Manual de usuario

A continuación, se mostrarán los tres usos de caso principales que son posibles realizar con la aplicación. En primer lugar, se muestra el icono de la aplicación y cómo localizarlo en el buscador de aplicaciones del dispositivo.





Instalación de OpenCV Manager

En el caso que se quiera acceder al modo de realidad aumentada y no esté instalada la aplicación OpenCV Manager aparecerá un mensaje informándose de la necesidad de instalar la aplicación.



Si estamos de acuerdo, se abrirá el Play Store para instalar la aplicación. Esta aplicación permite a otras aplicaciones usar las librerías de OpenCV sin necesidad de tener que incrustarlas dentro de cada aplicación que las use. Esto supone un ahorro de espacio y un beneficio a la hora de contar con la corrección de errores mediante las actualizaciones.





Modo creación

Este modo permite crear un nuevo test. Para acceder es necesario acceder a la pestaña "Mis Test" lo que habilitará un icono "+" en la esquina superior derecha. Clicando ese icono se activará la cámara.

| N | | | ⊖ ⊽⊿ 8 1:40 |
|---|---|----------|-------------------|
| Tester | | | ÷ |
| | TEST DISPONIBLES | MIS TEST | |
| Título: Test 2 Descripción: Pr Número de preg Autor: raulvibo Fecha de creac | ueba juntas: 10 ión: 25-09-2017, 12: 2 | 20:54 | Activo: Sí |

Una vez localizado el test, se muestra mediante realidad aumentada las casillas detectadas y con un aspa azul las casillas marcadas. Si pulsamos sobre guardar, accederemos a la pantalla de edición.



En esta pantalla podemos añadir un título y una descripción al test. Además, se muestra información adicional como el número de preguntas, el autor y la fecha de creación.





Antes de grabar la información mediante el icono que representa un disquete, podemos activar el test. Si decidimos activarlo haremos que esté disponible para el resto de usuarios.

Modo corrección

Este modo permite seleccionar un test almacenado en la base de datos de Firebase y

| | ⊖ ▼⊿ 🛿 1:45 |
|--|---|
| ester | |
| TEST DISPONIBLES MIS TEST | |
| | |
| Táula: Tast 1 | Activo: Sí |
| Descripción: Prueba | |
| Número de preguntas: 10 | |
| Autor: raulvibo Fecha de creación: 25-09-2017, 12:09:46 | J |
| | ester TEST DISPONIBLES MIS TEST Título: Test 1 Descripción: Prueba Número de preguntas: 10 Autor: raulvibo Fecha de creación: 25-09-2017, 12:09:46 |

Mediante una pulsación corta se entra en modo corrección. La aplicación procesa las imágenes que provienen de la cámara en busca de un test. Cuando la lectura es estable, se muestra el resultado. Las casillas se destacan y aparecen con una cruz en color azul aquellas que han sido marcadas.

Si la estructura del test no coincide con la estructura del test que se ha seleccionado en la pantalla anterior, se mostrará el resultado escaneado pero no se habilitará el botón de corrección.





Una vez escaneado el test, se da la opción de corregirlo. Pulsando sobre el icono de la cámara, la aplicación permite realizar de nuevo el escaneo del test. Pulsando sobre el aspa se vuelve a la pantalla anterior.



Se muestran con un visto bueno verde las respuestas correctas; con un aspa roja, las respuestas incorrectas y con un punto verde, las respuestas correctas que no se habían marcado.

En la parte superior derecha se muestra un círculo con la nota calculada. También quedan sobreimpresa la información referente al título del test en la parte superior izquierda y el usuario y el momento de realización de la corrección. Mediante el último icono se crea una imagen del resultado.

Modo actualización

Este modo nos permite modificar la información asociada a un test que previamente ha sido creado en el modo de creación.





Para activar el modo de actualización es necesario hacer una pulsación larga sobre un test. En este momento aparecerá un menú específico que nos permitirá realizar las siguientes acciones:

- Activar/desactivar el test: se consigue pulsando sobre el icono que muestra un ojo. Con cada pulsación se cambia el estado de activación. Esta opción también puede cambiarse a través de la pantalla de edición.
- Modificar el test: pulsando sobre el icono del lápiz se accede a la pantalla de edición.
- Volver a escanear el test: a través del icono de la cámara accedemos a la pantalla de realidad aumentada. Al detectar el test, nos dará la opción de actualizar



Si aceptamos, pasaremos a la pantalla de edición donde podremos comprobar que el número de preguntas es correcto y grabar los cambios efectuados.





Una vez guardados los cambios, ya estamos en disposición de usar el nuevo test con el sistema de realidad aumentada.