
exerClick

Aplicación móvil para el seguimiento de ejercicios en el aula

Septiembre de 2015

Especialidad de Computación

Autor:

Adrián Núñez Marcos

Directora:

Maite Urretavizcaya Loinaz

eman ta zabal zazu



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

informatika
fakultatea



facultad de
informática

Dedicado a...

En primer lugar, gracias a mi directora de proyecto, Maite Urretavizcaya Loinaz, por aceptar llevar mi proyecto y estar tan pendiente de mis dudas y comentarios, sé que pocos directores de proyecto atienden tan rápidamente a los alumnos y tienen tanta paciencia, por eso quiero agradecértelo especialmente. Como no, muchas gracias a Samara Ruiz también por estar siempre pendiente del progreso de desarrollo, haberme dado opiniones o respondido a dudas a cada rato, incluso fines de semana.

Gracias a mis aitas, Luis Fidel Núñez y Maite Marcos, por dejarme estudiar fuera de casa en San Sebastián. Haberme pagado toda la universidad aun con el esfuerzo que supone y no haber estado encima mio, dándome libertad en todo momento, demuestra la confianza que depositáis en mi, estoy muy agradecido por ello, un beso muy fuerte para vosotros.

He de agradecer también al equipo de ayudantes que he tenido para probar *exerClick*. Cada uno de ellos ha aportado su granito de arena y se ha tomado la molestia de invertir su tiempo en mi, ¡muchas gracias a Asier M., Uxue, Erik, Xabi, Mikel B., Ibai y Alex M.!

Finalmente, muchas gracias a todos los que habéis estado durante estos 4 años conmigo apoyándome, me lo he pasado genial con todos y he disfrutado mucho. Con este proyecto cierro una gran etapa de mi vida y me gustaría tomarme algunas líneas para agradecerle a las personas importantes que han pasado por ella. A todos mis compañeros de clase, que no podría ponerme a nombrar porque sois demasiados, hemos vivido muchos momentos juntos y ha sido un placer conoceros a todos, espero no olvidaros nunca. A mis compañeros de piso: Markel, Anartz y Sergio, gracias por aguantarme ahí todo un año, hemos tenido altibajos pero me ha gustado mucho convivir con vosotros. La gente de Ikubo: Uxue, Mikel, Goiatz, Alex y Gorka, estos últimos años habéis sido como una familia para mi dentro de la universidad, siempre ahí cuando os necesitaba, habéis sido un gran apoyo y ayuda. Sin olvidarme de la gente del Consejo de Estudiantes (Aizpea y Joanes) y del SWERC (Eneko, Jon, Mikel A. e Iván). También a todas las otras personas que habéis estado alrededor: me habéis aportado mucho y no podría poner palabras para expresar mi agradecimiento. Un abrazo a todos.

También quería agradecer a los profesores que más han marcado durante estos años: Jesús Ibáñez, Jesús Bermúdez y Juanan Pereira. Me habéis influenciado bastante para decidir lo que quiero hacer tras la universidad y os agradezco haber estado ahí.

Agradecimiento especial a las personas que estuvieron a mi lado todo el rato durante casi cada hora de desarrollo del proyecto: Asier, Xabi, Aitor y Gorka, ¡así da gusto hacer un proyecto!

Resumen

Proyecto de Fin de Grado de la especialidad de Computación. Se ha implementado la aplicación para el seguimiento de ejercicios en el aula llamada **exerClick**. Es una aplicación multiplataforma para móviles, evaluada en Android y en iOS y adaptada a esos sistemas gracias a la plataforma Cordova [6]. En su implementación se han utilizado las tecnologías web HTML5, CSS3 y Javascript, además de PHP para el servidor.

La aplicación permite que los profesores añadan ejercicios y los alumnos le envíen *feedback* sobre su realización mediante dos opciones: marcar una duda en el ejercicio o marcar el ejercicio como finalizado.

Laburpena

Konputazioko espezialitaterako Gradu Amaierako Proiektua. **exerClick** deituriko aplikazio bat implementatu da klaseko ariketen jarraipena egiteko. Plataforma anitzeko mugikorretarako aplikazioa da, Android eta iOS-en balioztatua eta sistema hoietara egokitua Cordova [6] plataformari esker. Honen implementazioan HTML5, CSS3 eta Javascript web teknologiak erabili dira, zerbitzarirako PHPaz gain.

Aplikazioak irakasleari ariketak gehitzea ahalbidetzen dio eta ikasleek ariketa horiei dagokien *feedback*-a bidaltzea bi aukeren bitartez: ariketa bat zalantza bezala edo amaitutxat bezala markatu.

Abstract

Computer Science speciality's Bachelor Thesis. The implemented application, called **exerClick**, aims to monitorize the realization of exercises in classrooms. It is a multiplatform application for mobile phones, adapted to Android and iOS using the Cordova [6] platform. In its implementation have been used HTML5, CSS3 and Javascript web technologies, in addition to PHP for the server.

The application lets the teacher to add exercises and students to send feedback for each exercise using two options: set a question for the exercise or mark the exercise as finished.

Índice general

1. Introducción	1
1.1. Contexto	1
1.2. Propuesta	2
1.3. Organización del documento	4
2. Documento de objetivos del proyecto (DOP)	7
2.1. Alcance y Exclusiones del proyecto	7
2.2. Fases y tareas del proyecto	9
2.2.1. Estudio de alternativas para la creación de una aplicación móvil	9
2.2.2. Formación	9
2.2.3. Documentación del proyecto	9
2.2.4. Presentación y Defensa del proyecto	10
2.3. Análisis de riesgos	10
2.4. Análisis de factibilidad	11
3. Gestión del proyecto	13
3.1. Metología de Desarrollo	13
3.1.1. Metología InterMod	14
3.1.2. Adaptación de InterMod	16
3.2. Equipos y Comunicación	16
3.3. Aspectos de desarrollo considerados	17
3.4. Gestión del proyecto a través de InterMod	18
3.4.1. Step 0 - Análisis del proyecto	19
3.4.2. Iteración 1	20
3.4.3. Iteración 2	21
3.4.4. Iteración 3	22
3.4.5. Iteración 4	22
3.4.6. Iteración 5	23
3.4.7. Iteración 6	25

3.5.	Documentación de la gestión del proyecto	27
3.5.1.	Selección de UOs por iteraciones	29
3.5.2.	Desarrollo de modelos por iteraciones	31
3.6.	Seguimiento	32
4.	Análisis de Requisitos	35
4.1.	Requisitos no-funcionales	35
4.1.1.	Requisitos de la interfaz	35
4.2.	Requisitos funcionales	36
4.2.1.	Requisito común: Autenticación	36
4.2.2.	UO1-S: Responder a un ejercicio	39
4.2.3.	UO2-S: Ver la descripción completa de un ejercicio	41
4.2.4.	UO1-T + UO2-T: Crear-Lanzar un ejercicio simple/detallado	42
4.2.5.	UO3-T: Cambiar el tipo de ejercicio	44
4.2.6.	UO4-T: Ver estadísticas de un ejercicio	45
4.2.7.	UO5-T + UO6-T: Ver la descripción completa y editar un ejercicio	47
4.2.8.	UO8-T + UO9-T + UO10-T: Cerrar sesión, Cambiar el idioma y la asignatura de la aplicación	49
5.	Diseño e Implementación	53
5.1.	M-2: Interfaces o lado del cliente	53
5.1.1.	Interfaz de autenticación de usuario	53
5.1.2.	Interfaz del alumno	55
5.1.3.	Interfaz del profesor	57
5.1.4.	Interfaz del perfil del profesor	58
5.1.5.	Uso de iconos mediante Font Awesome	63
5.2.	M-3: Lógica de negocio o lado del servidor	64
5.2.1.	Configuración	64
5.2.2.	Identificación de usuario	65
5.2.3.	Cargar la interfaz principal (P1 del profesor y P1 del alumno)	67
5.2.4.	Responder a un ejercicio (UO1-S)	73
5.2.5.	Ver detalles de un ejercicio (UO2-S y UO5-T)	74
5.2.6.	Crear-Lanzar un ejercicio (UO1-T y UO2-T)	74
5.2.7.	Cambiar detalles de un ejercicio (UO3-T y UO6-T)	75
5.2.8.	Ver estadísticas un ejercicio (UO4-T)	76
5.2.9.	Cambiar de idioma (UO9-T)	79
5.2.10.	Cambiar de idioma (UO10-T)	79
5.2.11.	Base de datos	79
5.3.	Estructura del proyecto	80
5.4.	Requisitos de dispositivos para su ejecución	81

6. Conclusiones y Líneas futuras	83
6.1. Objetivos alcanzados	84
6.2. Líneas futuras y Propuestas de mejora	86
6.3. Lecciones aprendidas	87
Bibliografía y Referencias	91
A. Actas de Reunión	93
B. Actas de Pruebas	107
C. Bases de datos	109

Índice de figuras

1.	Arquitectura de <i>PresenceClick</i>	3
2.	Progreso del proyecto según selección de UOs	30
3.	Progreso del proyecto según el desarrollo de Actividades	31
4.	Dedicación por tipo de tarea	32
5.	Dedicación por meses	33
6.	Autómata de estados correspondiente a la autenticación del alumno . . .	37
7.	Autómata de estados correspondiente a la autenticación del profesor . .	38
8.	Autómata de estados correspondiente al UO1-S	40
9.	Autómata de estados correspondiente al UO2-S	41
10.	P2: Crear-Lanzar un ejercicio simple (primer modelo)	42
11.	P2: Crear-Lanzar un ejercicio simple	43
12.	Autómata de estados correspondiente a los UOs 1-T y 2-T	44
13.	Iconos utilizados para representar cada tipo de ejercicio	44
14.	Autómata de estados correspondiente al UO3-T	45
15.	Autómata de estados correspondiente al UO4-T	46
16.	Autómata de estados correspondiente al UO5-T	48
17.	Autómata de estados correspondiente a los UOs 5-T y 6-T	49
18.	Autómata de estados correspondiente a los UOs 8-T, 9-T y 10-T	51
19.	Pantalla de autenticación de usuario	54
20.	P1 _S : vista del alumno	55
21.	P1 _S : cambio en el estado de un ejercicio	56
22.	P2 _S : ver los detalles de un ejercicio	57
23.	P1 _T ^A : Vista del profesor	58
24.	Perfil del profesor	59
25.	Pantallas del UO1-T (P2 _T)	59
26.	Interfaz del UO2-T: Crear-Lanzar un ejercicio detallado	60
27.	Pantallas del UO4-T (P5 _T)	61

28.	Pantallas del UO5-T (P3 _T)	62
29.	Pantallas del UO6-T (P4 _T)	63
30.	Estructura de archivos: raíz del proyecto	80
31.	Contenido de la carpetas principales de la raíz del proyecto	81
32.	Primer paso: crear un proyecto nuevo	88
33.	Segundo paso: tener la estructura y archivos correctos	89
34.	Parte de la base de datos de exerClick	109
35.	Base de datos de PresenceClick	110

Capítulo 1

Introducción

1.1. Contexto

Muchas veces nos encontramos con aulas con demasiados alumnos. Estas clases son especialmente comunes en los primeros cursos de los estudios universitarios, donde el número de alumnos alcanza fácilmente las tres cifras. Conocer todos los alumnos que tienen dudas y monitorizar a los alumnos para saber si todos o la mayoría han acabado el ejercicio son tareas difícilmente abordables por una sola persona. La mayoría de las veces se suelen ignorar los problemas y dudas, y se sigue adelante.

Los nuevos planes de estudio, que pretenden dejar atrás el sistema de educación mediante clases magistrales y dinamizar las clases, han supuesto un aumento en el número de clases prácticas y laboratorios que se realizan. Algunos centros han optado incluso por dividir las clases en grupos más pequeños para realizar las prácticas, pero muchas veces ésto no es posible. En esas situaciones el profesor acaba por no poder monitorizar completamente la clase.

Con el fin de tener un medio común se han implantado en los últimos años nuevas tecnologías en entornos docentes. Sin embargo, en muchos casos esta tecnología se limita a entornos de apoyo a la docencia más que al alumnado, siendo muy popular el sistema de gestión del aprendizaje Moodle. Además, el uso más frecuente de estos sistemas es el de simples almacenes de recursos bibliográficos (enlaces, apuntes, transparencias, etc.).

Por otro lado, la expansión de las tecnologías móviles y tabletas, con las que los alumnos están cada vez más familiarizados, no ha sido aprovechada. Estas tecnologías están ya mayoritariamente presentes en las aulas, la mayoría del alumnado dispone de alguno de estos dispositivos, pero su uso como herramienta educativa no es real, desperdiando

así todo su potencial como sistema de ayuda al aprendizaje. Es más, muchas veces el uso de estos dispositivos está prohibido o limitado en clase.

1.2. Propuesta

Nuestra propuesta pretende modificar y actualizar los modelos educativos presenciales a través de herramientas que faciliten la captura de la información de lo que sucede en el aula, con el objetivo de proporcionar *feedback* a profesores y estudiantes sobre su progreso en el aprendizaje. Esta propuesta se materializa en la aplicación *PresenceClick* que facilita la captura colaborativa de esta información entre alumnos y profesores de manera ágil.

PresenceClick es un entorno modular colaborativo que facilita el registro en tiempo real de la información sobre los alumnos en sesiones tradicionales de aprendizaje con el propósito de obtener feedback de ayuda para mejorar el proceso de enseñanza aprendizaje.

PresenceClick está compuesto por un sistema web (*webClick*), una aplicación móvil (*pClick*) y un servidor de datos para mantener el modelo del Alumno y el modelo de Grupo (*Learner and Group Models*). Ambas plataformas son a su vez modulares, dividiéndose así en diversos módulos encargados de las distintas interacciones que se registran en clase. La plataforma *pClick* captura las interacciones de aprendizaje en tiempo real. Por otra parte, *webClick* controla y muestra la información capturada durante sesiones cara a cara y permite a profesores y estudiantes seguir el progreso de sus alumnos durante el curso mediante visualizaciones y sus características *web responsive* permiten su acceso desde cualquier dispositivo, por ejemplo: PC, portátiles, tabletas o *smartphones*. La información capturada e integrada en los modelos de Alumno y de Grupo permite a profesores y estudiantes visualizar las actividades, conocimiento y comportamiento de estos. Este entorno cubre principalmente dos objetivos: mostrar a los profesores el estado de aprendizaje de sus alumnos y otras características para ayudarlos a adaptar sus estrategias de enseñanza; y mostrar a los alumnos su propio progreso y su comparación al grupo para promover un estado de reflexión que les permita mejorar su aprendizaje.

Actualmente, el Modelo Interacciones de *PresenceClick* incluye varios tipos de interacciones: de gestión (asistencia, evaluaciones), de actividad (preguntas de profesores y respuestas de alumnos), emocionales (sensaciones de los alumnos mientras realizan actividades de aprendizaje) y de comportamiento (participación, actividad).

La figura 1 muestra las líneas de interacción en la arquitectura de *PresenceClick*: (1) los asistentes comunican su presencia en clase al modelo del alumno a través de dispositivos

de control de presencia, (2) profesores y estudiantes capturas las interacciones que realizan mediante *pClick*, y (3) los profesores y estudiantes interactúan con *webClick* para obtener información sobre el progreso del estudiante y del grupo.

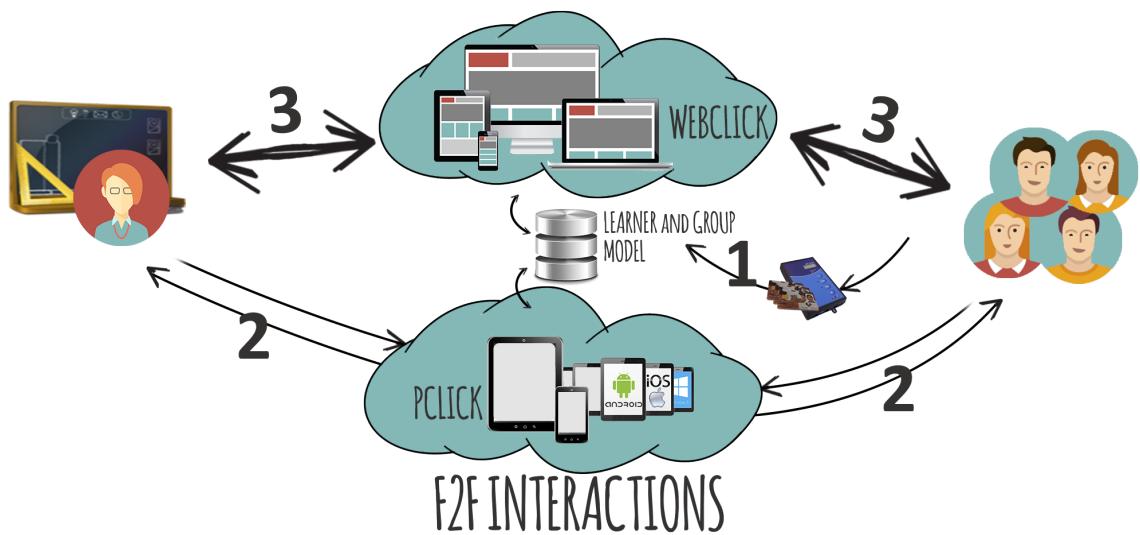


Figura 1: Arquitectura de *PresenceClick*

Actualmente *pClick* cuenta con un módulo llamado *qClick* [5]: un sistema de Pregunta-Respuesta en el aula, en el que el profesor lanza una pregunta a los alumnos y ellos a través de sus móviles pueden contestar a esa pregunta. Los resultados visualizados de forma gráfica a las preguntas realizadas pueden mostrarse a los alumnos en tiempo real. Este sistema permite motivar el debate en clase, bien antes de lanzar la pregunta para que discutan entre ellos las posibles respuestas o bien después de responderlas observando los resultados de la clase.

En este proyecto, nuestro objetivo es crear **un nuevo módulo de *pClick* para capturar las interacciones entre profesor-alumnos en sesiones de ejercicios**. Por el momento se desarrollará como una aplicación independiente y más tarde se integrará en *pClick*.

En esta aplicación el profesor dispondrá de una interfaz a la que accederá mediante su dispositivo móvil o tableta en clase e indicará a sus alumnos los ejercicios a realizar (esta actividad se podrá planificar previa a la clase). Por su parte, los alumnos (que tienen que haber fichado con su tarjeta de alumno para entrar en clase) con sus dispositivos móviles (smartphones o tabletas) recibirán las notificaciones de los ejercicios a realizar y podrán

indicar para cada uno si tienen dudas en su realización o si lo han terminado. El profesor podrá disponer en tiempo real de información sobre el porcentaje de alumnos que lo han realizado, alumnos que indican problemas en su realización y aquellos alumnos que no indican nada. Además el profesor podrá acercarse a comprobar y revisar las soluciones de los alumnos que indican haber terminado el ejercicio, y valorar su nivel de corrección o satisfacción en la realización, añadiendo las notas oportunas en el sistema que le permitirá seguir la evolución de cada uno de sus alumnos durante el curso. También podrá acercarse a aquellos que señalan problemas en su realización, con el fin de ayudarlos y evitar dificultades en su progreso.

Bajo este contexto surge **exerClick**, la herramienta para seguimiento de ejercicios en el aula. Esta herramienta, con todas sus funcionalidades, nace con el propósito de tener una visión más real de lo que hacen los alumnos, tanto una visión global del grupo como una individual y está dirigida a profesores y a los propios alumnos. De esta manera, el docente puede ofrecer un aprendizaje más adaptado e individualizado, aunque los grupos de alumnos sean muy grandes.

1.3. Organización del documento

En esta memoria se ha documentado el desarrollo de la herramienta **exerClick**, dentro del Trabajo de Fin de Grado (TFG) del autor. En el documento se describe la propuesta, la planificación y gestión que esta lleva consigo, la implementación llevada a cabo y las conclusiones finales.

En este primer capítulo se ha introducido el problema a resolver y se ha explicado la propuesta presentada en este proyecto.

En el capítulo 2 se presenta el Documento de Objetivos de Proyecto (DOP). Este recoge el alcance y las fases y tareas del proyecto, el análisis de riesgos y el análisis de factibilidad.

Una vez en el capítulo 3 se explica la gestión llevada a cabo durante el proyecto. Se presentan las metodologías utilizadas: Metodologías Ágiles e InterMod (adaptada a las necesidades de este proyecto). A continuación se detallan cada una de las iteraciones llevadas a cabo (como parte de la metodología InterMod): duración, objetivos y tareas realizadas. Al final del capítulo se muestra la documentación asociada a las iteraciones y los objetivos, además del seguimiento de tiempo realizado.

A continuación, en el capítulo 4 se detalla el análisis de requisitos. Primero se detallan

los requisitos no-funcionales y luego los funcionales (prototipos en papel llevados a cabo durante las primeras iteraciones que dan una visión global del proyecto).

En el capítulo 5 se explica el diseño e implementación llevados a cabo. Se comienza mostrando la estructura de documentos del proyecto, luego el diseño realizado en base al análisis de requisitos del capítulo 4 y finalmente una visión general de la implementación de la lógica de negocio.

Para finalizar, en el capítulo 6 se presentan las conclusiones, líneas futuras para el proyecto y las lecciones aprendidas.

Fuera de la estructura general de la memoria, tenemos la bibliografía y los apéndices. En estos últimos tenemos las actas de reuniones, las actas de pruebas y la vista de relaciones de la base de datos (de la parte utilizada o creada específicamente para el proyecto).

Capítulo 2

Documento de objetivos del proyecto (DOP)

Durante las sesiones prácticas de ejercicios en el aula a menudo el docente se pregunta si todo el mundo ha acabado los ejercicios propuestos con la esperanza de poder continuar con otro ejercicio o dando las explicaciones necesarias para poder continuar. Se puede preguntar si sus alumnos tienen vergüenza de plantear dudas o si pasan de ello, si no estará consumiendo demasiado tiempo en este ejercicio, etc. Con un conjunto de 100 alumnos, por ejemplo, resulta imposible observar el avance de los alumnos en los ejercicios y atender a todas las dudas que surgen. Incluso la estrategia de dividir la clase en grupos más pequeños para las clases prácticas, que puede parecer una buena idea, acaba consumiendo más tiempo, al tener que repetir lo mismo varias veces.

Lo ideal sería que existiera un medio para permitir a los alumnos dejar claro el estado en el que se encuentran, que el profesor pudiera ver como va cada alumno y que si hubiera dudas quedaran en algún sitio almacenadas para no dejar a ningún alumno sin su respuesta.

Con estos objetivos nace exerClick, que pretender ayudar a alumnos y docentes a llevar a cabo esta tarea. En este capítulo describiremos el alcance y las exclusiones del proyecto, además de la organización del documento. Se describirán las fases y tareas del proyecto, así como el análisis de riesgos y factibilidad.

2.1. Alcance y Exclusiones del proyecto

El alcance del proyecto queda recogido en un conjunto de objetivos que reflejan las necesidades de profesores y alumnos en las sesiones de ejercicios en aula. Por ello dividimos

los objetivos de desarrollo del proyecto, en objetivos del alumno y objetivos del profesor. Las necesidades de los alumnas/alumnos quedan recogidas en los siguiente objetivos de alumno:

- Ver en todo momento cuáles son los ejercicios propuestos en clase.
- Dejar constancia del estado de resolución de los ejercicios propuestos: si se ha terminado, si se tienen dudas o si se está realizando.
- Ver su progreso junto con el de sus compañeros en la realización de ejercicios.

Las necesidades del profesor quedan recogidas en los siguientes objetivos del docente:

- Ver en todo momento cuáles son los ejercicios propuestos en clase, los que se han terminado durante la sesión presente o los que aún no se han propuesto pero están preparados.
- Crear en cualquier momento un nuevo ejercicio (rápidamente o bien preparándolo tranquilamente). Y una vez creado tener la posibilidad de proponerlo a la clase o guardararlo para proponerlo más tarde.
- Ver por cada ejercicio su estado de realización: quiénes lo han terminado, quiénes tienen duda y quiénes no han respondido nada.
- Editar cualquier detalle de un ejercicio en cualquier momento.
- Valorar la realización de un ejercicio a un alumno concreto.

Además, se han fijado dentro del alcance los siguientes requisitos:

- Contar con la opinión de usuarios finales (alumnos y docentes) durante el desarrollo de la aplicación, asegurando su aceptación general.
- Internacionalizar la aplicación con 4 idiomas disponibles: castellano, euskera, inglés y francés.
- Desarrollar la aplicación para que se adapte al mayor número de dispositivos posibles: smartphones, tablets, etc., teniendo en cuenta los diferentes tamaños de pantalla.
- Funcionar en diversos sistemas operativos como Android, iOS y Windows Phone.

Se van a excluir del proyecto todas las funciones que se puedan desarrollar en PresenceClick ligadas a exerClick (es decir, sobre ejercicios en el aula, como pueden ser estadísticas más trabajadas).

2.2. Fases y tareas del proyecto

2.2.1. Estudio de alternativas para la creación de una aplicación móvil

Al principio el proyecto iba a seguir a su antecesor, *qClick*, como una aplicación web. Sin embargo, acabó pensándose durante una reunión inicial en convertirlo en una aplicación móvil para aprovecharse mejor de las ventajas de un móvil: notificaciones, no requiere necesidad de cargar la página cada vez que se entra, etc.

Ya que el proyecto iba a ser una aplicación web y por tanto se iban a usar tecnologías web se acordó usar un sistema como Apache Cordova para la implementación de la aplicación en lugar de crear una aplicación nativa. Así no se requerían de conocimientos de Android y por tanto se mantenían las bases iniciales del proyecto.

Se pensó en Apache Cordova ya que era conocido por uno de los integrantes del grupo GaLan, que estaba desarrollando una aplicación con esta tecnología. De esta forma se podía consultar en caso de alguna duda o pregunta en general directamente a dicha persona en lugar de perder excesivo tiempo buscando respuestas.

2.2.2. Formación

Ya se partía con una base propia de conocimiento en tecnologías web, por lo que no fue un gran problema.

Samara Ruiz fue la que aportó información sobre Responsive Web Design al inicio del proyecto para poder empezar a realizar las interfaces gráficas. No tenía conocimientos al respecto de esta herramienta y fue de gran ayuda una clase rápida introductoria.

Para aprender Apache Cordova se siguieron tutoriales encontrados en la red, tanto para crear la primera aplicación como para simulaciones y pruebas. Para algunas cosas también se consultó con el grupo GaLan en ciertos momentos.

2.2.3. Documentación del proyecto

La memoria del proyecto se empezó en una etapa temprana del proyecto. Se comenzó creando la plantilla de L^AT_EX con todo lo necesario desde cero. La plantilla de la universidad resultaba algo incómoda de usar para cambiar algunos detalles, es por eso que se

decidió crear una propia. Tras tener el estilo definido y los paquetes necesarios añadidos se realizó la estructura del documento, añadiendo sólo las cabeceras de cada sección, subsección, etc.

También al inicio se comenzó a realizar un seguimiento constante. Mediante Google Drive se iban manteniendo hojas de cálculo con cada tarea realizada, su duración y fecha, así como el cómputo total de horas por mes (y dentro de cada mes por iteraciones, si hubiera más de una). De esta forma plasmarlo más adelante fue una tarea muy sencilla.

La memoria no se desarrolló demasiado hasta que la aplicación empezaba a verse acabada, momento en el que empezó a tomar cuerpo con mucho contenido. Debido a que en abril se empezaron los estudios en el programa Erasmus en Alemania la memoria quedó completamente parada hasta junio y se alargó su refinamiento hasta agosto.

2.2.4. Presentación y Defensa del proyecto

La presentación del proyecto se realizará en los meses de agosto y septiembre, momento en el que se dispondrá de todo el tiempo necesario. Se plantea realizar unas transparencias para una presentación de 30 minutos con previo ensayo con la directora del proyecto.

2.3. Análisis de riesgos

En este punto se concretan dos riesgos que pueden dar problemas durante el proyecto y la solución encontrada para mitigarlos:

Programa Erasmus

El mayor riesgo durante el proyecto es el cuatrimestre en el programa Erasmus desde el mes de abril al mes de agosto: se desconocen los eventos que puedan acontecer durante esos meses, si la carga lectiva de las asignaturas y trabajos impedirá el seguir con la memoria, o simplemente puede ser una mala época anímicamente hablando para realizar el proyecto. De cualquier modo, para evitar este riesgo se ha pensado en acabar la aplicación por lo menos antes de marchar para minimizar riesgos.

Pérdida de información

La pérdida de información siempre es una posibilidad cuando se trabaja en una aplicación de esta índole. Para tener menos probabilidades de que se pierda absolutamente todo se realizan varias copias de todos los ficheros:

- Copia principal en el ordenador de trabajo.
- En Github, en el repositorio exerClick (ver sección ??).
- En Google Drive.

Se actualizan periódicamente, sin una frecuencia fija.

2.4. Análisis de factibilidad

Hay poca carga lectiva durante el primer cuatrimestre (3 asignaturas), dando tiempo a desarrollar el proyecto. Aun así, se ha visto que antes de empezar el programa Erasmus hay 3 meses sin ningún tipo de carga a parte del Trabajo de Fin de Grado. Sumado a que debido al Erasmus sólo se puede defender el proyecto en Septiembre y se tiene tiempo en verano para continuar tenemos una gran cantidad de tiempo prevista. Esto hace que la realización del proyecto sea completamente factible.

Capítulo 3

Gestión del proyecto

En este apartado se detallará la gestión llevada a cabo durante el proyecto. En primer lugar se detallan las metodologías de desarrollo utilizadas (las metodologías ágiles y la metodología InterMod) en el apartado 3.1. En el apartado 3.2 se habla de los equipos involucrados en el proyecto y de la comunicación utilizada y en el apartado 3.3 se habla de los aspectos considerados en el desarrollo (tecnologías y herramientas). La gestión a través de InterMod se detalla en el apartado 3.4. En el punto 3.5 se resume esta información y en el punto 3.6 se muestran las gráficas del seguimiento realizado durante el proyecto.

3.1. Metología de Desarrollo

Las metodologías ágiles proponen el desarrollo de software de forma incremental e iterativa, donde varios grupos se unen para obtener requerimientos y soluciones nuevas gracias a su colaboración. Se enfatiza en el cara a cara más que en la documentación. Un ejemplo de esta metodología es SCRUM.

La metodología es bastante flexible, pudiendo adaptarse a cada proyecto. Es decir, no tiene reglas absolutamente fijas. En su base está el *Manifiesto Ágil* en el que se basan todas las variantes de esta metodología. Este manifiesto [4] dice así:

- Individuos e interacciones sobre procesos y herramientas
- Software funcionando sobre documentación extensiva
- Colaboración con el cliente sobre negociación contractual

- Respuesta ante el cambio sobre seguir un plan

Al comienzo del proyecto se fijó el uso de esta metodología, con 4 grupos trabajando en paralelo (los equipos vienen definidos en el apartado 3.2). Sin embargo, las iteraciones acabaron volviéndose excesivamente largas, habiendo varios grupos parados. Si bien se respetó la filosofía de la metodología ágil en cierto sentido, y aun con cierta flexibilidad en su adaptación, al final del proyecto acabó por volverse algo parecido a una metodología ágil sin llegar a serlo.

3.1.1. Metología InterMod

InterMod es una metodología de trabajo, que será utilizada para este proyecto, desarrollada por el grupo de investigación GaLan de la Facultad de Informática de San Sebastián. Se trata de una metodología ágil para el desarrollo de software interactivo de alta calidad.

En InterMod se define el Objetivo de Usuario (User Objective o UO) como el deseo del usuario que puede ser alcanzado mediante una o más funcionalidades. Diferentes UOs son desarrollados durante el proyecto y la unión de todos, en su globalidad, forma la aplicación final. Además, el mismo UO puede incluir uno o más requerimientos funcionales y/o no-funcionales. Existen a su vez diferentes tipos de UO:

- **UO Directo:** Es un objetivo del usuario final.
- **UO Indirecto:** Surge a partir de otros UOs por necesidades internas del desarrollo (no son propiamente deseos del usuario). Aparecen durante el desarrollo debido a la fusión o división de otros UOs.
- **UO Reutilizable:** Es un UO creado y evaluado, total o parcialmente, en otro proyecto o en el proyecto actual que puede ser reutilizado.

Basándose en la propuesta del Object Management Group's Model Driven Architecture, InterMod establece sus actividades basadas en modelos.

Por cada actividad se desarrollan siempre dos fases: la creación del modelo (independiente de la plataforma) y su posterior evaluación. Las evaluaciones de usabilidad son especialmente útiles para los UOs Directos ya que reflejan una necesidad del usuario, por tanto es importante que un grupo de estos esté involucrado. Para agilizar el proyecto, algunos modelos pueden ser evaluados únicamente por expertos en usabilidad. Las actividades no se dan por acabadas y pueden continuar activas durante varias iteraciones hasta conseguir una evaluación positiva.

Existen dos tipos de actividades para el desarrollo de UOs: Actividades de Desarrollo (DA) y Actividades de Integración (IA). Existen tres tipos de DAs:

- **DA-1:** *Análisis y Diseño de la Navegación.*
- **DA-2:** *Construcción de la Interfaz.*
- **DA-3:** *Codificación de la Lógica de Negocio.*

Para asegurar el desarrollo incremental de la aplicación son necesarias las Actividades de Integración (IA). Existen tres tipos de IAs:

- **IA-1:** *Integración de los Modelos de Requerimientos.*
- **IA-2:** *Integración de la Interfaz.*
- **IA-3:** *Integración de la codificación y refactorización.*

Las actividades de desarrollo e integración de cada tipo da lugar a un modelo. Así, las DA-1 e IA-1, relativas al análisis de requisitos, desembocan en el modelo de requisitos (M-1); las DA-2 e IA-2, relativas a las interfaces, crean el modelo de presentación (M-2) y las DA-3 e IA-3, las asociadas a la lógica de negocio, constituyen el modelo de funcionalidad (M-3). El modelo M-1 es un modelo abstracto sobre el que se asientan las bases, basado en él se forma el M-2, que contiene todos los elementos gráficos y otras características definidos en el M-1. Finalmente el modelo M-3 establece la implementación en un lenguaje de programación concreto.

InterMod define una metodología dividida en iteraciones, y, a diferencia del resto, define un paso previo, Step 0. En esta etapa previa se realiza el análisis del proyecto y se definen los UOs iniciales del mismo y el diseño general. A continuación se pasa a la iteración 1, luego la 2, etc. y se continúa así hasta dar por finalizado el proyecto. Cada iteración está dividida en 3 pasos:

- **Step 1.i:** Construir/Actualizar la lista de UOs.
- **Step 2.i:** Planificar las actividades para los diferentes equipos.
- **Step 3.i:** Realizar las actividades planificadas.

3.1.2. Adaptación de InterMod

Debido a las características del proyecto se ha decidido realizar algunas modificaciones al esquema de InterMod:

- Los UOs normalmente se denotan por UOX (siendo X el número del UO). En este proyecto hemos distinguido dos usuarios, por tanto hará falta un identificador extra para saber a que usuario corresponde el UO. Se usará la notación UOX-Y, siendo Y la inicial en inglés del tipo de usuario: 'S' para el estudiante (*Student*) y 'T' para el profesor (*Teacher*).
- Siguiendo la línea del punto anterior, se ha hecho un cambio en la notación típica de los modelos. De M-1(X), el primer modelo del UOX, a M-1(XY), el primer modelo del UOX-Y (siguiendo la notación del punto anterior).

3.2. Equipos y Comunicación

Se han identificado 4 equipos que participarán en el proyecto (con diferentes niveles de implicación):

- **Equipo 1:** Formado por el alumno, Adrián Núñez. Se encargará del diseño de las interfaces y de la implementación de la aplicación.
- **Equipo 2:** Realizara las evaluaciones pertinentes y estará formado por alumnos de la facultad.
- **Equipo 3:** Segundo equipo para las evaluaciones, estará formado por miembros del grupo GaLan.
- **Equipo 4:** Se encargará de las evaluaciones con los usuarios finales.

Con el fin de mantener la comunicación con los interesados en el proyecto se plantea el canal de comunicación *Slack* (<https://slack.com/>) al inicio del proyecto. Mediante este canal se intercambiaran mensajes rápidos entre los demás miembros del grupo y Adrián Núñez para cualquier duda u opinión rápida.

También se realizaran periódicamente reuniones presenciales, como mínimo, entre Maite Urretavizcaya y Adrián Núñez para monitorizar el desarrollo del proyecto. A estas reuniones también han asistido Samara Ruiz (como desarrolladora de PresenceClick) y Juan Miguel López (como especialista en evaluaciones de usabilidad).

Para la comunicación con el Equipo 2 se utilizó la herramienta de mensajería móvil *Telegram* (que permite, al contrario que otras conocidas, el intercambio de cualquier tipo

de fichero, vital para pasarles el fichero .apk).

Finalmente, el proyecto y este documento estarán públicos en *GitHub* en el siguiente repositorio: <https://github.com/AdrianNunez/exerClick>. Se realizaran periódicamente actualizaciones del mismo. También se dispone de otras copias en el ordenador personal del autor, Adrián Núñez, y en el espacio de almacenamiento *Google Drive*, en una carpeta compartida con Maite Urretavizcaya y Samara Ruiz.

3.3. Aspectos de desarrollo considerados

En la propia documentación de Apache Cordova vienen indicados los Sistemas Operativos soportados a los que se puede adaptar la aplicación:

- iOS (Mac)
- Amazon Fire OS (Mac, Linux, Windows)
- Android (Mac, Linux, Windows)
- BlackBerry 10 (Mac, Linux, Windows)
- Windows Phone 8 (Windows)
- Windows (Windows)
- Firefox OS (Mac, Linux, Windows)

Ya que el abanico de posibilidades es extenso se decidió reducir las posibilidades a iOS, Android, BlackBerry 10 y Windows Phone 8. El desarrollo principal y todas las pruebas iniciales se realizarán utilizando Android.

Al final del proyecto, debido a incompatibilidades con la infraestructura disponible, se realizaron sólo evaluaciones en dispositivos físicos Android e iOS.

Antes del desarrollo de la aplicación, se concretaron las siguientes tecnologías mínimas a utilizar:

- HTML5
- CSS3
- PHP

- MySQL

Como calidad añadida se utilizan las siguientes tecnologías:

- Javascript
- JQuery

Las herramientas principales para el desarrollo de la aplicación serán Notepad++ y Eclipse. Para la gestión de la base de datos se usará phpMyAdmin y para la gestión del servidor se usará WinSCP.

Para el desarrollo en Android se utilizará la SDK de Android y el emulador Genymotion (mucho más rápido que el emulador por defecto).

3.4. Gestión del proyecto a través de InterMod

El proyecto se ha llevado a cabo usando una adaptación de InterMod (apartado 3.1.2), con una duración de 6 iteraciones totales. Se considera que la última iteración ha acabado cuando la aplicación final se ha terminado. Las horas restantes dedicadas a la redacción de la memoria forman parte del Trabajo de Fin de Grado, si bien se han añadido horas de redacción de memoria dentro de las distintas iteraciones.

Antes de la primera iteración, tal y como especifica InterMod, hay un step0. Aquí se realiza una visión global de la aplicación mediante un brainstorming y se plantean la lista de UOs (User Objective) iniciales. De esta lista se extraen los UOs trabajados durante las siguientes iteraciones.

Las dos primeras iteraciones sirven para pasar esta visión global a unos prototipos en papel de la aplicación (que suponen el M-1 de los UOs). A partir de la iteración 3 se desarrollan los M-2, es decir, el diseño de las pantallas correspondientes a los UOs basadas en los prototipos en papel. En la iteración 5 se comienza la implementación de la lógica de negocio.

En general la iteración 6 es la más larga de todas y, debido a que había mucho trabajo previo y material reutilizable, en ella se desarrollan la mayor parte de los UOs.

Las iteraciones se dividen en los step 1.i (Lista de UOs), 2.i (Planificación de la iteración) y 3.i (Ejecución de las actividades) como se ha comentado en apartados anteriores. A continuación se detalla cada iteración.

3.4.1. Step 0 - Análisis del proyecto

Fecha de inicio: 9/10/14

Fecha de fin: 18/10/14

Dedicación total: 2 horas, 35 minutos

Esta fase previa al desarrollo de la aplicación principal se planteó dentro de la metodología InterMod como una introducción. En este proyecto se realizaron dos tareas principalmente:

- Documentarse sobre el estado del arte en el ámbito de aplicaciones de seguimiento de ejercicios en el aula, sobre bases de datos que representaran el concepto de ejercicio, etc.
- Realizar un *Brainstorming* para plantear todas las ideas posibles para la aplicación. De esta forma se pretender tener una idea más clara de como será la aplicación que se desea (una visión general).

Finalmente se realizó la lista de UOs inicial que incluía muchos UOs no prioritarios. Sin embargo, se decidió dejar la lista completa e ir abordándola por prioridades y actualizándola iteración a iteración.

UOs inicialmente planteados

UOs del alumno

UO1-S: *Responder a un ejercicio.* El alumno quiere indicar que ha acabado o que tiene dudas con un ejercicio que el profesor ha propuesto.

UOs del profesor

UO1-T: *Crear-Lanzar un ejercicio simple.* El profesor quiere proponer un ejercicio rápidamente, sin escribir mucho.

UO2-T: *Crear-lanzar un ejercicio detallado.* El profesor quiere proponer la realización de un ejercicio preparado previamente o con bastantes detalles.

UO3-T: *Dar por finalizado un ejercicio propuesto activo.* El profesor quiere terminar con uno de los ejercicios que propuso.

UO4-T: *Ver estadísticas de un ejercicio.* El profesor desea ver qué tal le ha ido a la clase en general o a un alumno en un ejercicio.

UO5-T: *Ver la descripción completa de un ejercicio.* Un profesor quiere ver la descripción completa de un ejercicio (identificador, enunciado, página, tema, etc.).

UO6-T: *Editar un ejercicio.* El profesor desea editar los atributos de un ejercicio.

UO7-T: *Evaluar el ejercicio de un alumno.* El profesor quiere valorar la realización de un ejercicio a un alumno.

UO8-T: *Cerrar sesión.* El profesor quiere cerrar su sesión activa.

UO9-T: *Cambiar el idioma de la aplicación.* El profesor desea cambiar el idioma con el que lee la aplicación.

UO10-T: *Cambiar de asignatura.* El profesor, que tiene más de una asignatura, quiere cambiar de una asignatura x a otra asignatura y.

3.4.2. Iteración 1

Fecha de inicio: 21/10/14

Fecha de fin: 2/11/14

Dedicación total: 3 horas, 25 minutos

Step 1.1. Lista de UOs

Lista de UOs = UO1-S, UO1-T..UO10-T

Step 2.1. Planificación de la iteración

En esta primera iteración el plan es el siguiente:

- Equipo 1: M-1(1S, 1T..10T)
- Equipo 3: Evaluación{M-1(1S, 1T..10T)}

Step 3.1. Ejecución de las actividades

Con la síntesis de lo acordado en la última reunión se modifica el documento de Brains-torming, que queda corregido para tener todas las ideas previas claras por escrito.

Se empiezan realizando los M-1 (prototipos de papel) de los UOs planteados inicialmente (de la lista entera). Los prototipos en papel se envían durante la iteración a Maite y Samara para valoraciones rápidas y posteriores correcciones.

Al final de la iteración no se consiguen evaluar satisfactoriamente los prototipos en papel debido a los fallos que tenían o mejoras que se podían incluir, de modo que se retoman en la siguiente iteración.

3.4.3. Iteración 2

Fecha de inicio: 3/11/14

Fecha de fin: 10/11/14

Dedicación total: 3 horas, 30 minutos

Step 1.2. Lista de UOs

Lista de UOs = No cambian

Step 2.2. Planificación de la iteración

Se continúa con el mismo plan de la pasada iteración.

- Equipo 1: M-1(1S, 1T..10T)
- Equipo 3: Evaluación{M-1(1S, 1T..10T)}

Step 3.2. Ejecución de las actividades

Los modelos enviados a evaluar en la anterior iteración no eran adecuados, así pues se corrigen los prototipos en papel con lo acordado durante la pasada reunión (síntesis de la evaluación realizada) y se realizan nuevos a limpio para evaluar.

Esta vez los prototipos en papel son satisfactoriamente evaluados, en la siguiente iteración se puede comenzar con los M-2.

3.4.4. Iteración 3

Fecha de inicio: 11/11/14

Fecha de fin: 12/1/15

Dedicación total: 72 horas, 55 minutos

Step 1.3. Lista de UOs

Lista de UOs = No cambian

Step 2.3. Planificación de la iteración

- Equipo 1: M-2(1T, 2T, 3T, 4T)

Step 3.3. Ejecución de las actividades

Se comienza con el M-2 del UO1-T, que servirá como base para otros UOs (ya que casi todos comparten la pantalla madre que se desarrolla en esta fase). Ya que es un requisito para este UO, se ha desarrollado también una interfaz de autenticación de usuario.

El proyecto queda exportado a Eclipse para intentar generar el fichero .apk de modo que cualquiera pueda probar la aplicación, aunque no se consigue que este funcione. Queda pendiente para una futura iteración.

También se fija que la versión de Android mínima para utilizar la aplicación sin problema es la 3.0.

Al final de esta iteración se ha realizado un gran consumo de tiempo en el aprendizaje de Cordova y algunas pruebas, por lo tanto no ha habido tiempo para el desarrollo de UOs (a parte del UO1-T antes mencionado, que no se ha terminado). Se realiza al final una reunión para ver el progreso del M-2 del UO1-T.

Actividades de formación complementarias al desarrollo:

Durante esta iteración se instala el entorno de trabajo de Cordova. Se instalan inicialmente las plataformas Android y Windows Phone. Se decide también iniciar todas las pruebas utilizando Android.

3.4.5. Iteración 4

Fecha de inicio: 13/1/15

Fecha de fin: 26/1/15

Dedicación total: 16 horas

Step 1.4. Lista de UOs

Lista de UOs = No cambian

Step 2.4. Planificación de la iteración

- Equipo 1: M-2(1S, 1T, 2T, 3T, 4T)
- Equipo 3: Evaluación{M-2(1S, 1T, 2T, 3T, 4T)}

Step 3.4. Ejecución de las actividades

Se añaden las correcciones acordadas en la anterior reunión al M-2 del UO1-T y se inician en esta fase los M-2 de los UO4-T y UO1-S. También se sube el proyecto a GitHub para el control de versiones y tener una copia de seguridad extra. Se consigue terminar los M-2 de los UO 1-T, 4-T y 1-S.

Más avanzados en la iteración se le ha proporcionado al alumno la base de datos de PreseceClick vacía para poder empezar con la lógica de negocio o el M-3 de los UOs realizados. Se han creado dos tablas nuevas exclusivas para exerClick:

- **exercise:** Almacena información sobre un ejercicio.
- **exercisestate:** Almacena el estado de un ejercicio.

También se realiza una fase de aprendizaje de la base de datos proporcionada para aprender a utilizarla. Esto servirá como preparación para afrontar los M-3 (implementación) en las próximas iteraciones, ya que hará falta el uso de la base de datos. El UO2-T no se empieza por falta de tiempo.

Se finalizan los M-2 de todos los UOs seleccionados excepto del UO2-T. Queda pendiente su M-2 para la siguiente iteración. Al haber acabado los M-2 de los otros UOs se puede comenzar con sus M-3.

3.4.6. Iteración 5

Fecha de inicio: 27/1/15

Fecha de fin: 6/2/15

Dedicación total: 23 horas, 15 minutos

Step 1.5. Lista de UOs

Lista de UOs = No cambian

Step 2.5. Planificación de la iteración

- Equipo 1: M-2(2T), M-3(1S, 1T, 3T, 4T)

El UO3-T no necesita un M-2, así que se aborda directamente su M-3.

Step 3.5. Ejecución de las actividades

Al inicio de la iteración se revisa la visión general que se tenía de la aplicación (desarrollada en el step0) y se decide introducir modificaciones en ella para cambiar su aspecto visual.

Se ha empezado realizando el M-3 del UO1-T, luego el UO4-T y finalmente el UO3-T. El objetivo de la iteración es tener una aplicación mínimamente funcional para realizar pruebas. Es decir, poder al menos realizar la función básica de que el profesor lance un ejercicio, el alumno lo vea y responda. Además, ver las estadísticas del ejercicio propuesto también sería importante.

Al abordar el UO3-T se han encontrado otros subobjetivos, derivados también de la reunión del día 27 de enero. En la reunión se decidió que los ejercicios, estuvieran en el estado que estuvieran (activos, finalizados o preparados para lanzarse), podrían cambiarse a cualquier estado. Es decir, el UO3-T tenía como objetivo el cambio de ejercicio activo a ejercicio finalizado, pero con esta decisión se debe de poder realizar cualquier otro cambio posible además del anterior.

UO3-T: *Cambiar el estado de un ejercicio.* El profesor quiere, por ejemplo, terminar con uno de los ejercicios que propuso o quiere volver a poner como activo un ejercicio que por error ha marcado como finalizado.

Este UO sólo necesita la pantalla principal de la vista del profesor (y los botones incluidos en esta). Al estar hecha se ha abordado directamente su M-3.

En esta iteración no se ha implementado la identificación de usuarios por falta de tiempo. Se ha ocultado la pestaña de ejercicios preparados hasta que se aborde el tema, así que en esta iteración no se hace nada con los ejercicios preparados. Además, no se ha

trabajado en el UO2-T debido que se quería una aplicación mínimamente funcional, como se ha comentado, y por tanto se han centrado los esfuerzos en los UO 1-T, 4-T y 1-S. El UO3-T se ha conseguido terminar debido a que era el más simple que el UO2-T y se ha decidido realizarlo primero.

El equipo 3 finaliza la iteración evaluando los UO 1-T, 4-T y 1-S.

Nota:

Para pasar la aplicación al móvil es necesario que el código PHP (de servidor) este en un servidor externo, de otro modo no funcionará. Se ha creado para ese propósito un servidor con el espacio que proporciona 000webhost (<http://www.000webhost.com/>) sin costo alguno para realizar pruebas al principio. Se ha instalado la base de datos proporcionada ahí mismo con las tablas originales más las dos creadas en la iteración anterior.

3.4.7. Iteración 6

Fecha de inicio: 9/2/15

Fecha de fin: 25/3/15

Dedicación total: 102 horas, 15 minutos

Step 1.6. Lista de UOs

Lista de UOs = UOs anteriores + UO2-S

El UO2-S no estaba recogido en la captura de requisitos inicial del step0. Sin embargo es un UO de vital importancia para el alumno que no se había tenido en cuenta. Su M-1, M-2 y M-3 se basan directamente en el UO5-T (al ser prácticamente iguales).

Step 2.6. Planificación de la iteración

- Equipo 1: M-2(2S, 2T, 5T, 6T, 8T, 9T, 10T), M-3(2S, 1T, 2T, 3T, 4T, 5T, 6T, 8T, 9T, 10T)
- Equipo 2: Evaluación de las .apks generadas durante el desarrollo
- Equipo 3: Evaluación de las .apks generadas durante el desarrollo

Los M-3 de los UO 1-T, 4-T y 1-S no fueron considerados adecuados tras la evaluación, por lo que siguen en desarrollo.

Step 3.6. Ejecución de las actividades

La iteración 6 se puede dividir en dos partes importantes: la primera en la que se crean los UOs restantes para tener una aplicación completamente funcional y la segunda en la que se genera finalmente un apk y se comienzan las pruebas.

Tras la reunión que daba inicio a esta iteración se realizaron todas las modificaciones y correcciones acordadas para empezar (no se validaron correctamente los M-3 de los UO 1-T, 3-T y 4-T). El objetivo principal era completar el UO3 (como se comentaba en la anterior iteración) que está formado por el UO1-T, el UO1-S y el UO4-T (es decir, añadir ejercicios, responder a ellos y ver los resultados). Para ello primero había que convertir los datos estáticos en datos obtenidos de la base de datos (asignatura, estadísticas, etc.).

Cuando se había acabado el UO3 se decidieron realizar todos los UOs menores que faltaban. Debido a que el trabajo más tedioso estaba hecho (realizar las interfaces principales) y que muchos de los siguientes UOs basaban su M-2 en UOs ya realizados (UO1-T y UO4-T) se consiguieron realizar todos los UOs del 5 al 10 rápidamente. En general, estas fueron las actividades realizadas posteriores al UO3:

- Se creó el perfil para el profesor y el UO8-T (cerrar sesión) y el UO9-T (cambiar de asignatura).
- Se probó por primera vez en una tablet la aplicación completa.
- Se añadieron estadísticas diferentes para los ejercicios que estaban finalizados.
- Se añadieron los ejercicios preparados.
- Se modificó el funcionamiento de los botones para marcar el estado de un ejercicio en la vista del alumno.
- Se realizaron pruebas con el Grupo 2, formado por alumnos, por primera vez.
- Se añadió la opción de crear ejercicios avanzados.
- UO5-T para ver detalles de un ejercicio y UO6-T para editar un ejercicio.
- Finalmente se tradujo todo a 4 idiomas y se dio la posibilidad de cambiar entre ellos en el perfil del profesor (UO10-T).

Para generar el fichero .apk se usó Eclipse Luna y Java 8. Se exportó el proyecto a Eclipse y se realizaron las configuraciones necesarias. Surgió unas gran cantidad de errores que supusieron la mayor parte del tiempo de esta parte. Finalmente, cuando todo funcionaba correctamente y la aplicación se ejecutaba en el emulador, Eclipse generaba

un fichero .apk correcto. Este fichero le fue pasado a Maite Urretavizcaya, Samara Ruiz y al Equipo 2 para realizar pruebas.

3.5. Documentación de la gestión del proyecto

A continuación se presenta un resumen completo de los objetivos desarrollados en las iteraciones. Con el paso de estas los objetivos se han fusionado, divididos o han ido evolucionando. Estos procesos se muestran de manera gráfica para dar una visión rápida del origen de cada UO. En esta leyenda se muestran todos los UOs desarrollados:

Notación

UOX-Y, donde X corresponde a la numeración del UO y la Y determina el tipo de usuario alumno (S de student) o profesor (T de teacher).

Alumno

UO1-S: *Responder a un ejercicio.* El alumno quiere indicar que ha acabado o que tiene dudas con un ejercicio que el profesor ha propuesto.

UO2-S: *Ver detalles de un ejercicio.* El alumno quiere ver más detalles sobre un ejercicio disponible y activo en la sesión.

Profesor

UO1-T: *Crear-Lanzar un ejercicio simple.* El profesor quiere proponer un ejercicio rápidamente, sin escribir mucho.

UO2-T: *Crear-Lanzar un ejercicio detallado.* El profesor quiere proponer la realización de un ejercicio preparado previamente o con bastantes detalles.

UO3-T: *Cambiar el tipo de ejercicio.* El profesor desea cambiar un ejercicio del tipo que tiene a otro cualquiera (de activo a finalizado, por ejemplo).

UO4-T: *Ver estadísticas de un ejercicio.* El profesor desea ver qué tal le ha ido a la clase en general o a un alumno en un ejercicio.

UO5-T: *Ver la descripción completa de un ejercicio.* Un profesor quiere ver la descripción completa de un ejercicio (identificador, enunciado, página, tema, etc.).

UO6-T: *Editar un ejercicio.* El profesor desea editar los atributos de un ejercicio.

UO8-T: *Cerrar sesión.* El profesor quiere cerrar su sesión activa.

UO9-T: *Cambiar el idioma de la aplicación.* El profesor desea cambiar el idioma con el que lee la aplicación.

UO10-T: *Cambiar de asignatura.* El profesor, que tiene más de una asignatura, quiere cambiar de una asignatura x a otra asignatura y.

UO1: *Lanzar ejercicios.* El profesor desea lanzar un ejercicio simple o avanzado, dependiendo de su necesidad.

UO2: *Lanzar ejercicios y visualizar resultados.* El profesor desea lanzar un ejercicio y, tras lanzarlo, ver el estado de cada alumno en dicho ejercicio.

Generales

UO3: *Lanzar ejercicios, responder al ejercicio y visualizar los resultados.* El profesor desea lanzar un ejercicio, el alumno desea responder al ejercicio y finalmente el profesor desea ver el estado de ese alumno en el ejercicio que ha lanzado previamente.

Autenticación de usuario: *Se requiere que un usuario se identifique como estudiante o como profesor antes de entrar a la aplicación.* No lo consideramos como un UO aunque se mencione aquí, sino como una parte asociada a todos los UOs como desarrollo incremental de estos.

Y los UOs definidos pero que no han sido realizados:

UO7-T: *Evaluar el ejercicio de un alumno.* El profesor quiere valorar la realización de un ejercicio a un alumno.

3.5.1. Selección de UOs por iteraciones

En la figura 2 se muestra el proceso de selección de UOs dividido por iteraciones. Con "seleccionar" se quiere decir que se añade a la planificación de la iteración para realizar alguno de los modelos del UO.

Los UOs directos aparecen con un borde negro grueso y los UOs por Fusión con un borde negro simple. Las líneas que unen los UOs indican una relación de fusión.

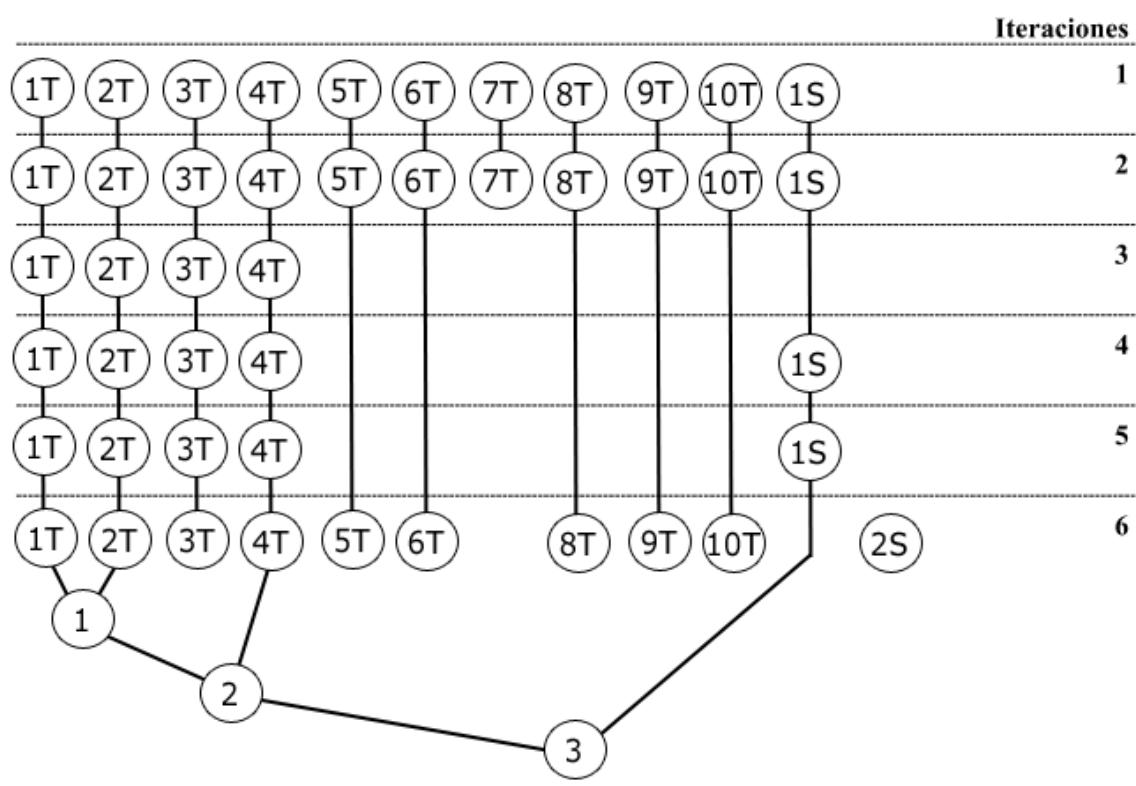


Figura 2: Progreso del proyecto según selección de UOs

3.5.2. Desarrollo de modelos por iteraciones

En la figura 3 se detalla el desarrollo de los diferentes tipos de actividad (de desarrollo o de integración) para cada UO en base a las iteraciones.

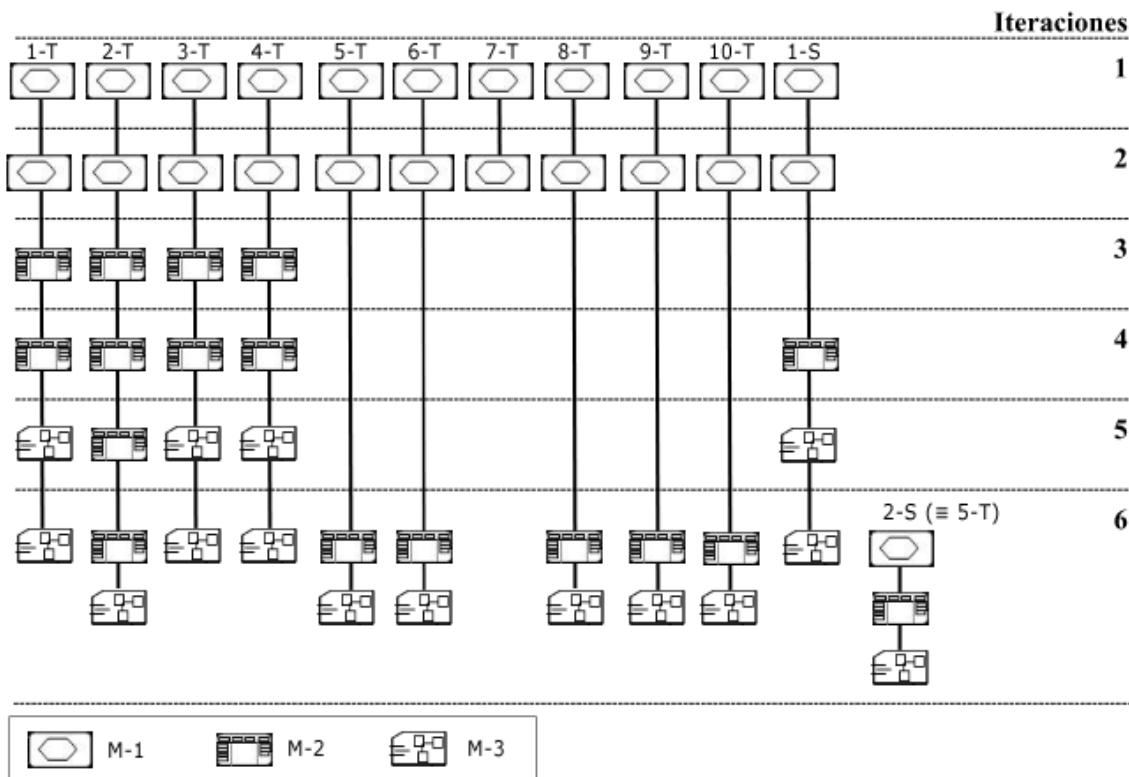


Figura 3: Progreso del proyecto según el desarrollo de Actividades

Las actividades se unen por líneas si estan relacionadas entre sí, estén o no en la misma iteración. Las líneas gruesas indican el flujo normal de desarrollo y las líneas delgadas el flujo de integración.

En la iteración final se integraron varios UOs para formar 3 UOs:

- UO1: UO1-T + UO2-T.
- UO2: UO1 + UO4-T.
- UO1: UO2 + UO1-S.

3.6. Seguimiento

La dedicación total al proyecto ha sido de 264 horas y 56 minutos horas. Al acabar la iteración 6 se habían dedicado 223 horas y 56 minutos.

Dedicación por tipo de tarea



Figura 4: Dedicación por tipo de tarea

Dedicación del progreso por meses

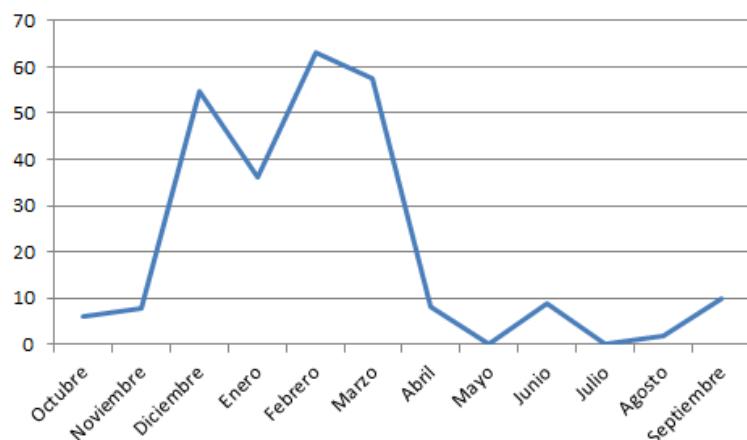


Figura 5: Dedicación por meses

Capítulo 4

Análisis de Requisitos

Este apartado contiene dos secciones mayores: el apartado 4.1 en el que se detallan los requisitos no-funcionales y el apartado 4.2 en el que se explican los requisitos funcionales recogidos de los modelos M-1 de cada UO.

4.1. Requisitos no-funcionales

La aplicación ha sido desarrollada con dos requisitos básicos:

- Que tenga mucha funcionalidad con el mínimo posible de *clicks*.
- Que para el alumno no suponga una distracción.

Así pues se ha creado un diseño agradable que pretende tener mucha funcionalidad, especialmente en la vista de profesor. En la vista del alumno se ha minimizado el diseño y los elementos para que no existan distracciones importantes en clase.

4.1.1. Requisitos de la interfaz

Para hacer más intuitiva la aplicación se han concretado unos colores e iconos específicos para cada tipo de ejercicio, de modo que sea fácil asociar una acción o estado mediante un color o un ícono.

- Ejercicios activos: color rojo, pretendiendo indicar actividad, y un ícono de un avión de papel, simbolizando un ejercicio que ha sido lanzado.

- Ejercicios finalizados: color azul, indicando calma (ya hemos terminado), y un icono de una bandera de meta, simbolizando el fin.
- Ejercicios preparados: color amarillo y un icono de un reloj, indicando que esto no se ha lanzado aun (viendo las aplicaciones móvil más utilizadas resulta intuitivo el significado del reloj).

Para las estadísticas se ha dejado el color verde y un icono de un gráfico de barras. El color azul se emplea para otros botones, sin usar ningún ícono.

4.2. Requisitos funcionales

En este apartado se presentan los requisitos funcionales recogidos en interfaces de papel

Siguiendo las pautas fijadas por los requisitos no-funcionales se han realizado interfaces sencillas e intuitivas. Se le ha dado mucha importancia a la filosofía de "en pocos *clicks*" que sigue la aplicación. Por tanto, ante todo, se han minimizado la cantidad de transiciones entre pantallas y el uso de excesivos botones para buscar mucha funcionalidad en pocos *clicks*.

El resultado queda plasmado en estos modelos de requisitos (M-1) de cada objetivo de usuario (*User Objective*). Los modelos están construidos mediante diferentes pantallas, las comunes al estudiante y al profesor son nombradas PX, donde X es el número de la pantalla, y las que son exclusivas de cada uno se denotan como

$$PX_Y$$

donde X es el número de la pantalla e Y es el usuario (puede tomar los valores S para el estudiante y T para el profesor).

4.2.1. Requisito común: Autenticación

La autenticación es un requisito común para todos los UOs. En las figuras 6 y 7 se ve el autómata de estados correspondiente a la autenticación de un alumno y la de un profesor, respectivamente. Ambos autómatas comparten algunas pantallas, llamadas P0_I y P0_{II}, que se engloban en P0.

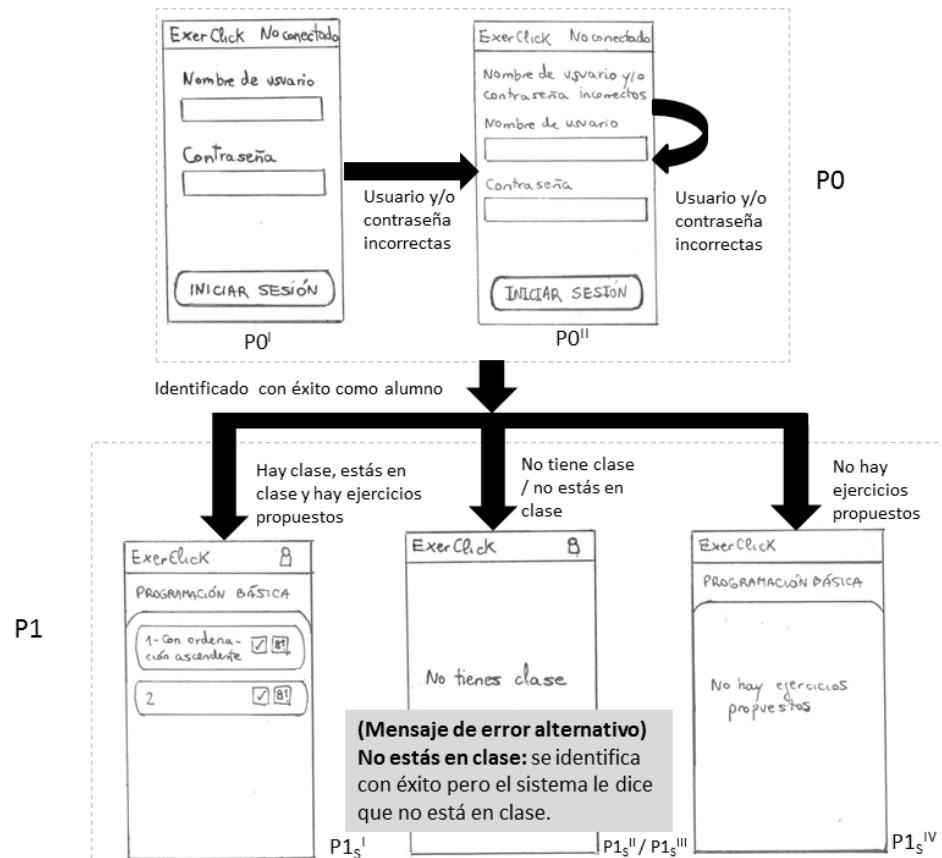


Figura 6: Autómata de estados correspondiente a la autenticación del alumno

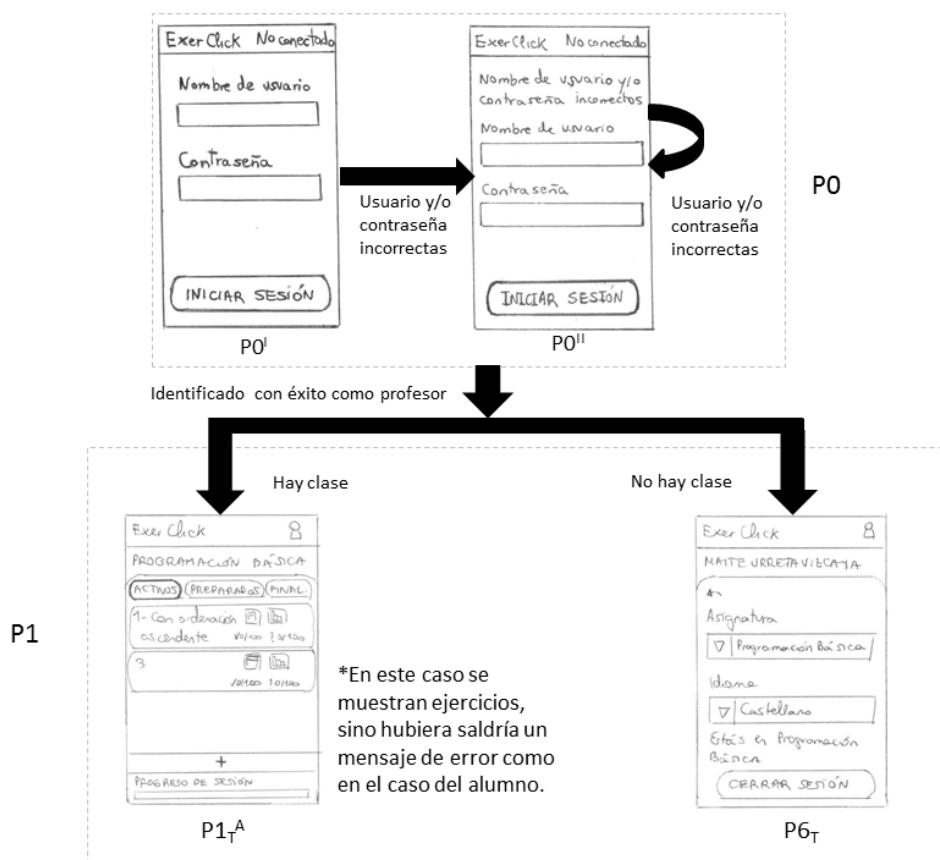


Figura 7: Autómata de estados correspondiente a la autenticación del profesor

En cualquier variante de P0 podemos ver dos campos para introducir nuestro nombre de usuario y contraseña y un botón para acceder. Si los datos son correctos (y tenemos clase¹, además de estar en ella²) veremos la interfaz P1 del profesor o del estudiante (dependiendo del tipo de usuario con el que no hayamos autenticado, P1_S¹ para el alumno y P1_T^A para el profesor). Si los datos introducidos son incorrectos iremos a P0_I. También podemos ver variantes de P1 en el caso del alumno si no tenemos clase, no estamos en clase o no hay ejercicios propuestos.

Durante este capítulo se asumirá que el usuario se ha autenticado correctamente (como alumno o como profesor) y está en su correspondiente pantalla principal. Por tanto se obviará que todos los UOs deben pasar por P0 antes de llegar a sus pantallas iniciales.

4.2.2. UO1-S: Responder a un ejercicio

El modelo M-1(1-S) requiere de la pantalla principal del alumno (P1_S) Como se comentó en el apartado anterior se da por hecho que el alumno se identificó correctamente, tiene clase, está en ella y hay ejercicios propuestos. En P1_S podemos responder a los ejercicios indefinidamente "sin cambiar de pantalla". No nos iremos a otra pantalla, sino que esta variará. Esta transición se puede apreciar en la figura 8.

Nuestras opciones para responder a un ejercicio son marcar el ejercicio como finalizado (este botón será azul) o marcar una duda en el ejercicio (este será rojo). En la figura 8 se puede ver en la parte de arriba el proceso de marcar un ejercicio como finalizado y de desmarcarlo. Análogamente para marcarlo con una duda en la parte de abajo de la misma figura.

¹Se considera que hay clase desde la hora de inicio de la clase hasta la hora final y también durante 15 minutos antes de la hora de inicio

²Para reconocer que un alumno ha entrado en clase este tiene que fichar con su tarjeta de alumno.

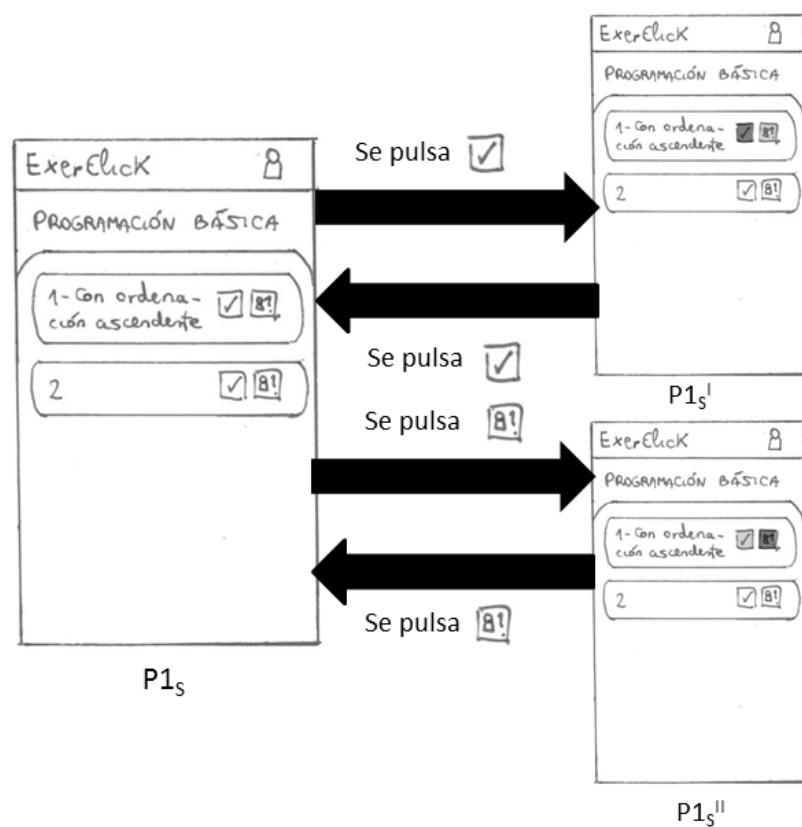


Figura 8: Autómata de estados correspondiente al UO1-S

4.2.3. UO2-S: Ver la descripción completa de un ejercicio

El modelo M-1(2-S) cuenta con una pantalla $P2_S$ a la que se accede mediante $P1_S$. En la figura 9 se puede ver el autómata de estados correspondiente. Haciendo click sobre el contenedor de cualquier ejercicio (sobre la "caja") podremos acceder a $P2_S$. La interfaz se carga con datos sobre el ejercicio sobre el que hemos hecho click (identificador siempre y opcionalmente cualquier otro dato que el profesor haya incluido). Si no hubiera más detalles añadidos aparecerá un mensaje avisando de ello. En $P2_S$ se aprecia un enlace "+Más" que estaba pensado para mostrar más detalles sobre el ejercicio. Este enlace es eliminado de aquí en adelante.

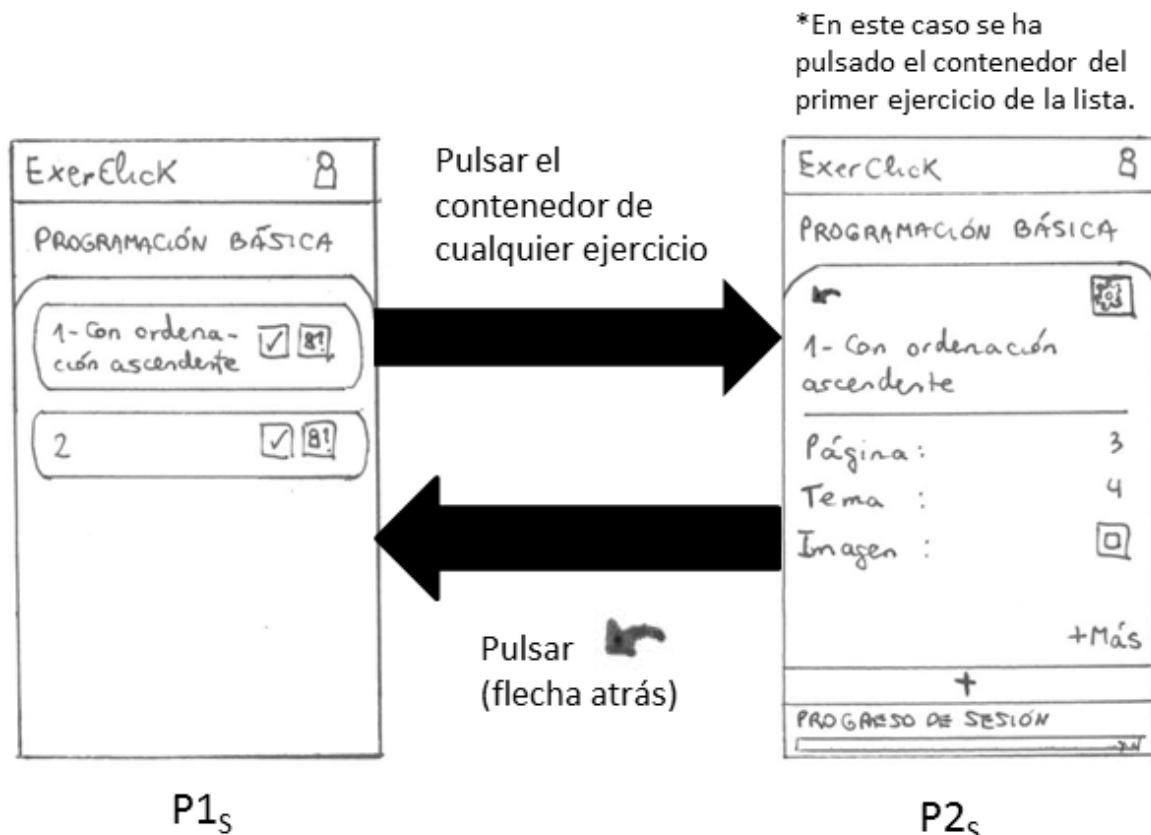


Figura 9: Autómata de estados correspondiente al UO2-S

4.2.4. UO1-T + UO2-T: Crear-Lanzar un ejercicio simple/detallado

En esta sección se comentarán los UOs 1-T y 2-T, ya que el UO2-T es un modelo incremental del UO1-T, y, por tanto, están estrechamente ligados. Se comenzará hablando del UO1-T y más adelante del UO2-T.

UO1-T: P2_T (primer modelo)

Estando en el sistema autenticados y con acceso a la asignatura, podemos crear-lanzar ejercicios pulsando en el símbolo '+', que nos da acceso a la pantalla nueva P2_T, accesible desde cualquier variante de P1. En la primera versión de P2_T (figura 10) había 4 campos básicos para crear un ejercicio: un identificador, la página, el tema y una imagen. Todos los campos menos el identificador eran opcionales. Sin embargo, más tarde se pensó en reducir esto para que se viera más simple.



Figura 10: P2: Crear-Lanzar un ejercicio simple (primer modelo)

Tenemos dos botones disponibles: el de la parte superior (con un avión de papel) que lanzará directamente el ejercicio como un ejercicio activo y el de la parte inferior que guardará el ejercicio como ejercicio preparado para que podamos seguir editándolo antes de lanzarlo. Para guardar o lanzar un ejercicio es necesario que el campo del identificador no esté vacío. La flecha hacia atrás cerrará la interfaz y volveremos a P1_T. Finalmente, disponemos de un enlace "Avanzado" que nos llevará a la interfaz P2_T^I.

UO1-T: P2_T (nuevo modelo)

En la nueva versión de la interfaz P2_T (figura 11) se utiliza la interfaz de P1, añadiendo por encima una pestaña nueva en la que podremos crear un ejercicio simple. Estos ejercicios sólo necesitan de un identificador, por tanto, en esta pequeña pestaña sólo hay un campo para añadir ese identificador. Los campos omitidos pueden ser añadidos mediante el UO2-T (en la pantalla P2_T^I). Además se añade la opción de cerrar la ventana haciendo click fuera de ésta. El enlace en el que antes ponía "Más" ahora por "Avanzado", pero en cuanto a funcionalidad sigue siendo lo mismo (muestra mas detalles para personalizar el ejercicio). Precisamente, mostrar más detalles es el objetivo del UO2-T.

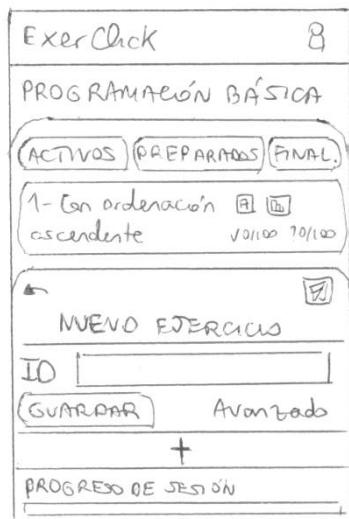


Figura 11: P2: Crear-Lanzar un ejercicio simple

UO2-T

Desde P2_T es posible acceder a P2_T^I pulsando sobre el enlace "Avanzado" (como se ha comentado previamente). Esta pantalla tiene un gran parecido con el primer modelo de P2_T. Añade varios campos extra (página y tema) respecto a la nueva versión de P2_T. La flecha hacia atrás cierra por completo la ventana superpuesta sobre P1_T. Nuevamente el enlace "Más" se elimina más adelante. En la figura 12 se ve el autómata de estados completo que fusiona los UOs 1-T y 2-T.

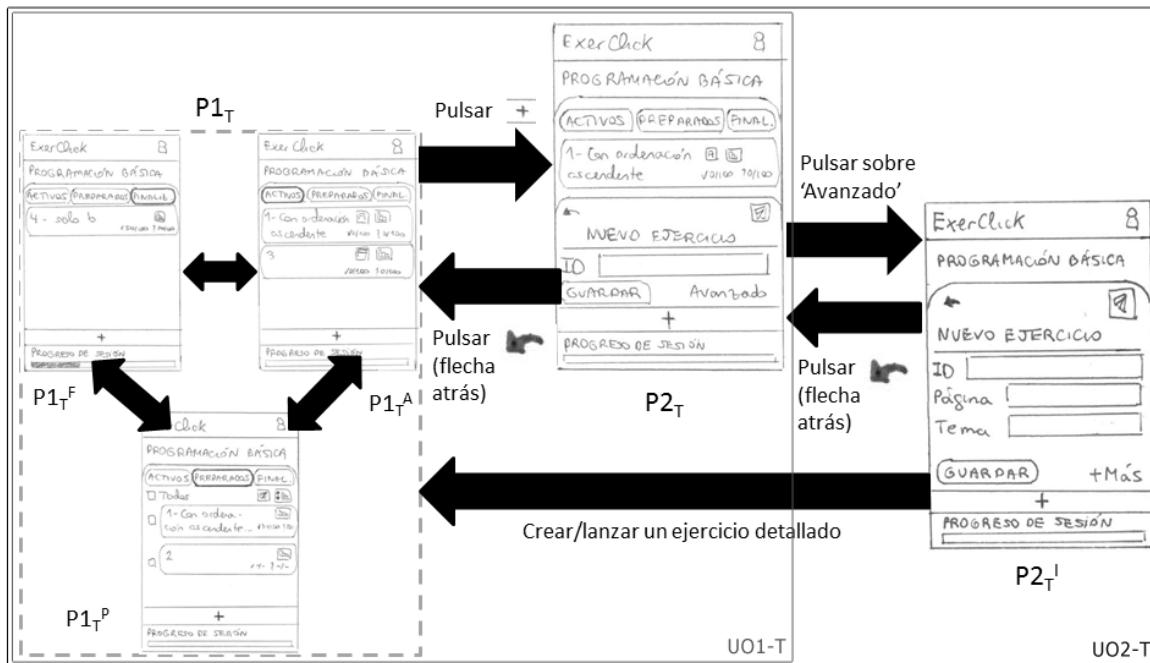


Figura 12: Autómata de estados correspondiente a los UOs 1-T y 2-T

4.2.5. UO3-T: Cambiar el tipo de ejercicio

Este UO al principio tenía como finalidad cambiar un ejercicio de activo a finalizado (es decir, dar por finalizado un ejercicio propuesto en clase). Se cambió a "cambiar el tipo de un ejercicio" para dar más libertad y poder enmendar errores (ya que se daba la posibilidad de cambiar de cualquier tipo a otro). En los prototipos en papel no aparecen todos los botones, sólo la versión vieja (con el botón para finalizar ejercicios nada más).

Se deciden usar 3 iconos (figura 13) para representar cada tipo de ejercicio para que resulten intuitivos sin usar texto.

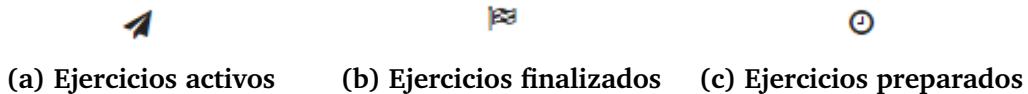


Figura 13: Iconos utilizados para representar cada tipo de ejercicio

Desde cualquiera de las variantes de P1: P1_T^A (ejercicios activos), P1_T^F (ejercicios finalizados) y P1_T^P (ejercicios propuestos) podemos hacer cambios. En la figura 14 se puede ver el autómata de estados por el que se guían estos cambios.

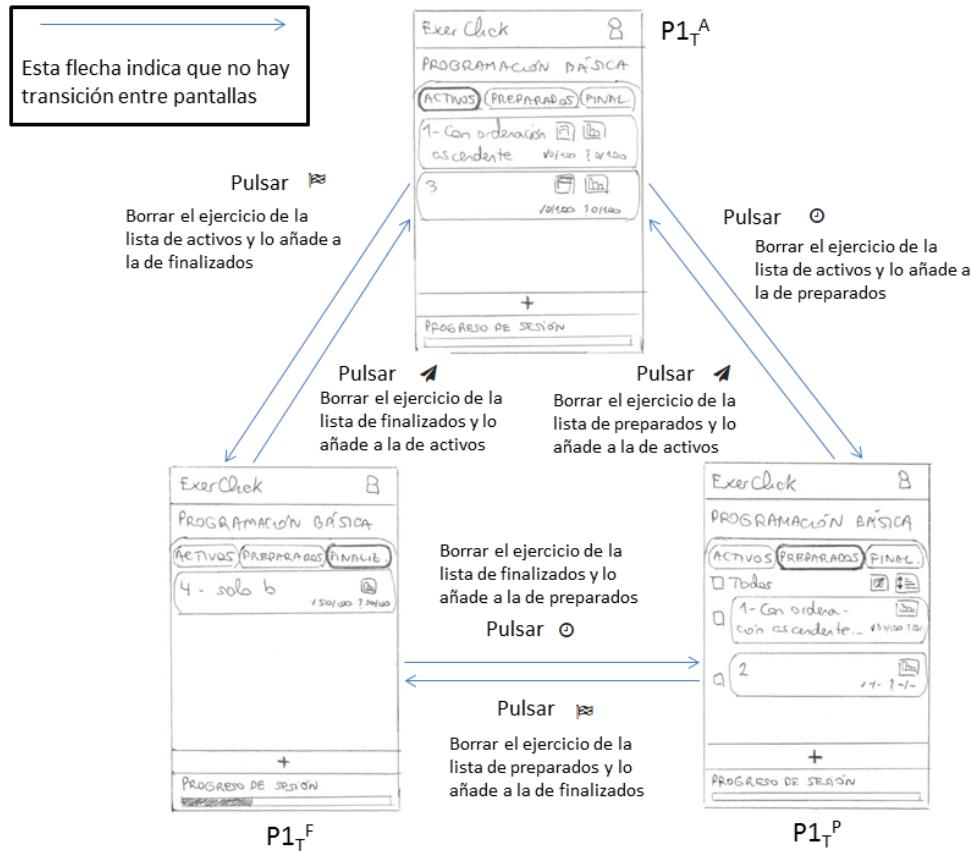


Figura 14: Autómata de estados correspondiente al UO3-T

Finalmente, existe una limitación a estos cambios para un caso especial. Cuando el profesor entra en la aplicación (puede hacerlo en cualquier momento, aunque no haya clase, tal y como se ve en la figura 7) y quiere lanzar un ejercicio. Un requisito especial es que los ejercicios sólo pueden ser lanzados en clase. Es decir, si el profesor accede a la aplicación y no hay clase no puede crear ejercicios directamente para lanzar (UO1-T/UO2-T) ni cambiar un ejercicio al tipo activo.

4.2.6. UO4-T: Ver estadísticas de un ejercicio

El profesor puede ver las estadísticas de un ejercicio, siempre y cuando sea un ejercicio activo o finalizado. Las estadísticas se muestran en tres pantallas diferentes: $P2_T^T$, $P5_T^A$ y $P5_T^D$.

- $P5_T^T$: nos muestra a todos los alumnos involucrados en el ejercicio (que estén en la asignatura) en forma de lista. Muestra, por cada alumno, el estado de realización del ejercicio: acabado, con alguna duda o sin marcar nada.

Tanto $P5_T^A$ como $P5_T^D$ son filtros aplicados a la lista de $P5_T^T$.

- $P5_T^A$: nos filtra sólo los alumnos que han marcado el ejercicio como acabado.
- $P5_T^D$: igual que $P5_A$ pero marcando como duda el ejercicio.

En la figura 15 podemos ver el autómata de estados correspondiente al UO4-T.

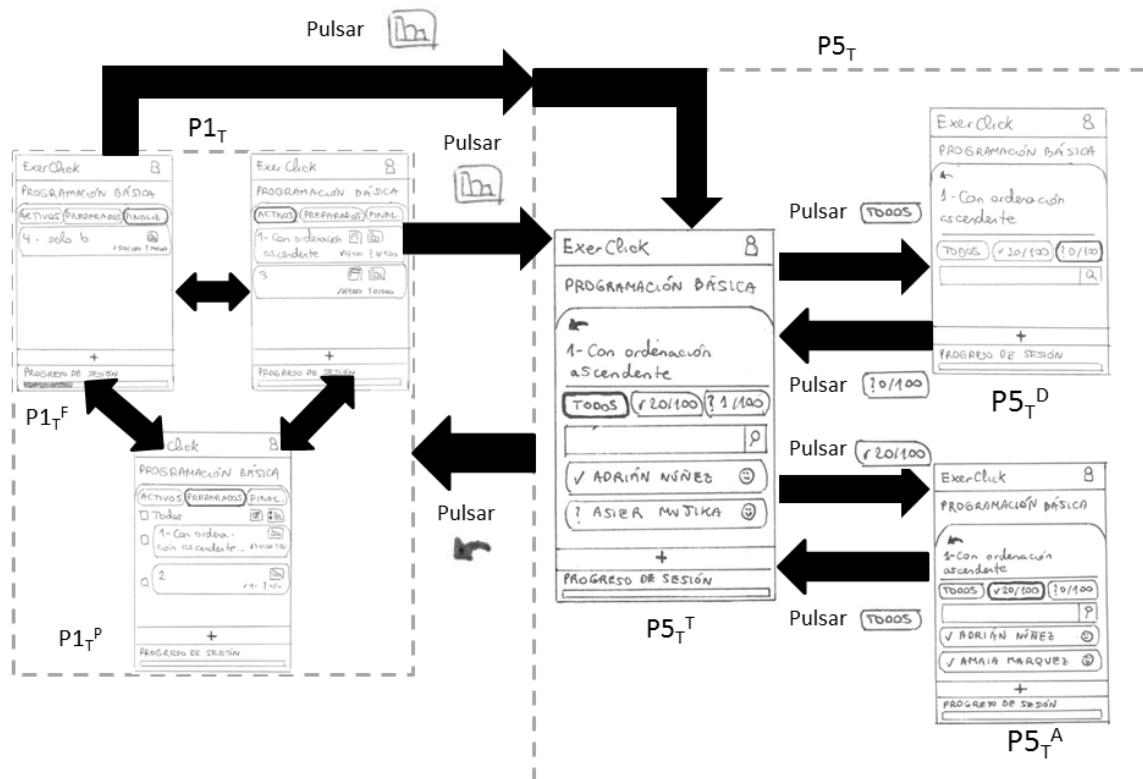


Figura 15: Autómata de estados correspondiente al UO4-T

Desde $P1_T^A$ o $P1_T^F$ siempre accederemos directamente a $P5_T^T$ y desde ésta podemos acceder a $P5_T^A$ y $P5_T^D$ mediante la pulsación de los botones de la parte superior. En los

botones de finalizados y duda aparecen iconos, el número de acabados/dudas respecto al número de alumnos en formato n/m y el mismo dato en formato de porcentaje.

4.2.7. UO5-T + UO6-T: Ver la descripción completa y editar un ejercicio

En este apartado detallaremos el análisis de requisitos de los UOs 5-T y 6-T. El UO6-T es un modelo incremental del UO5-T (al igual que los UOs 1-T y 2-T).

UO5-T

Es posible ver la descripción completa de un ejercicio (UO5T) desde cualquier variante de P1 ($P1_T^A$, $P1_T^F$ o $P1_T^P$), haciendo *click* sobre el contenedor de cualquier ejercicio (sobre la "caja") podremos acceder a P3, tal y como se muestra en la figura 16. La interfaz se carga con datos sobre el ejercicio sobre el que hemos hecho *click* (identificador siempre y opcionalmente cualquier otro dato que hayamos introducido como profesor). Si no hubiera más detalles añadidos aparecerá un mensaje avisando de ello.

Este UO es similar al UO2-S del alumno. Sin embargo, a diferencia del UO2-S (donde se mostraban también los detalles de un ejercicio) en P3 tenemos un botón con un engranaje en la parte superior, al pulsarlo nos llevará a P4 (editar el ejercicio). El enlace "+Más" sigue igual que en el UO2-S, sirve para mostrar más detalles. Más tarde es eliminado.

UO6-T

Accederemos a $P4_T$ mediante $P3_T$ utilizando el botón con un engranaje (figura 17). En esta nueva pantalla podremos editar el ejercicio del que estábamos viendo los detalles en $P3_T$.

En la nueva ventana tenemos el título del ejercicio y la flecha de ir atrás como en P3. En el centro de la ventana tenemos un campo por cada detalle del ejercicio, el campo se cargará con el valor previo (si tuviese). Podemos editar esos valores y pulsar el botón de "Guardar" para que se guarden los cambios. Si pulsamos atrás los cambios no se guardarán.

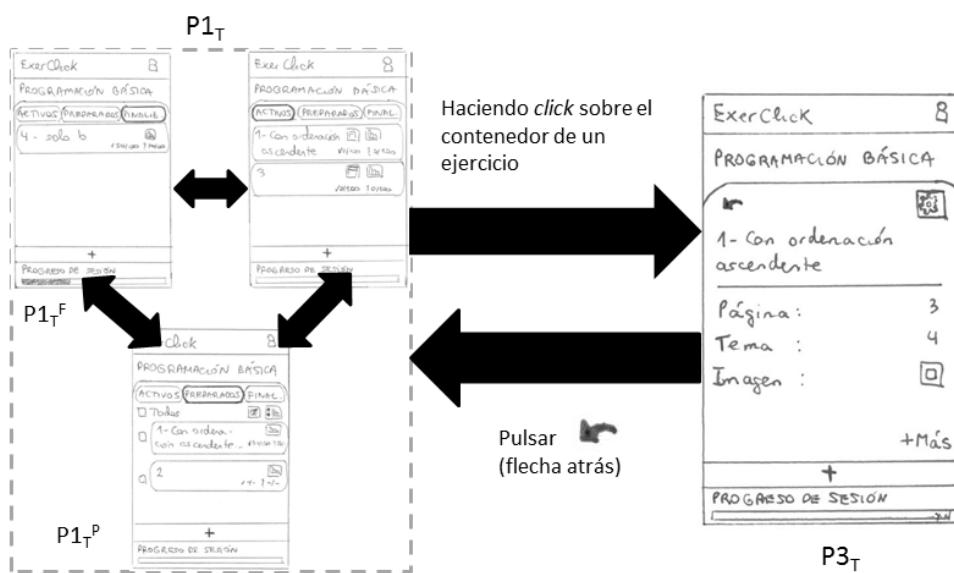


Figura 16: Autómata de estados correspondiente al UO5-T

Más adelante el botón de atrás se elimina y se mantiene el botón de engranaje, pulsar este botón tiene el mismo efecto que la flecha para ir hacia atrás. También se cambia el título de la ventana (donde sale el título del ejercicio) por un campo con el título del ejercicio. Este campo es editable, lo cual permite editar el identificador del ejercicio (este campo no se puede dejar vacío).

El enlace "Más" muestra más detalles que editar (aunque fue eliminado más adelante).

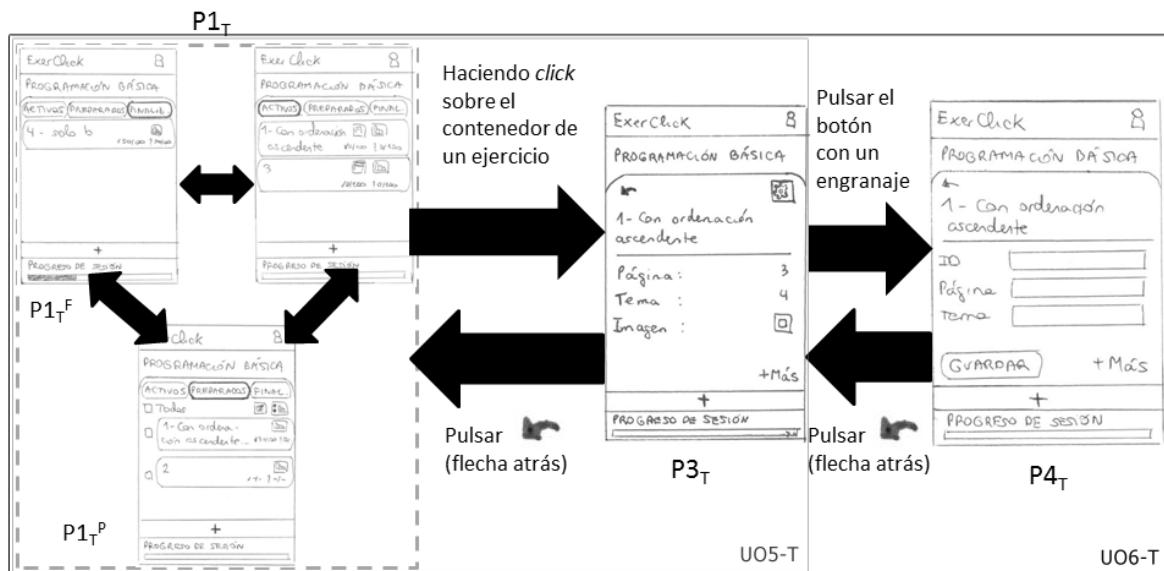


Figura 17: Autómata de estados correspondiente a los UOs 5-T y 6-T

4.2.8. UO8-T + UO9-T + UO10-T: Cerrar sesión, Cambiar el idioma y la asignatura de la aplicación

La última interfaz de la aplicación es P6_T. Esta interfaz es el perfil del profesor, donde se pueden realizar algunas opciones:

- **UO8-T:** Cerrar la sesión abierta (y volver a la pantalla inicial, P0).
- **UO9-T:** Cambiar el idioma de la aplicación (entre Castellano, Euskera, Inglés y Francés).
- **UO10-T:** Cambiar la asignatura activa.

Para acceder al perfil hay que hacer *click* sobre el nombre de usuario o el símbolo de usuario de la parte superior derecha de la aplicación. La pantalla del perfil tiene dos desplegables (para escoger idioma y para escoger asignatura), un campo para mostrar la asignatura activa (de la que se muestran ejercicios en P1_T) y un botón para cerrar sesión. Más tarde en lugar de la flecha hacia atrás que hay en la parte superior se pone un botón con el texto "Ir a clase" para volver a P1_T (es decir, el mismo efecto que la flecha hacia atrás). En el perfil podremos realizar las acciones descritas previamente de la siguiente forma:

- **Cerrar sesión:** Tenemos que pulsar el botón "Cerrar sesión" de la parte inferior. Volveremos a P0, y desde ahí podemos autenticarnos de nuevo.
- **Cambiar de idioma:** En el perfil del profesor aparecerá una opción "Idioma" con un desplegable. Al escoger cualquier opción de este la aplicación cambiará automáticamente de idioma (no hay transición de pantallas).
- **Cambiar de asignatura:** Existe otra opción con desplegable llamada "Asignatura", donde podemos escoger entre cualquier asignatura del profesor. Al escoger una nueva asignatura no se apreciarán más cambios que el texto que aparece sobre el botón de cerrar sesión (donde aparece la asignatura que está seleccionada), sin embargo la "asignatura activa" ha cambiado. Es decir, si volvemos de nuevo a los ejercicios (P1_T) veremos que estamos en la nueva asignatura seleccionada, y, por tanto, aparecerán los ejercicios de esa asignatura.

Todo se resume en el autómata de estados de la figura 18, donde aparecen las transiciones de los 3 UOs.

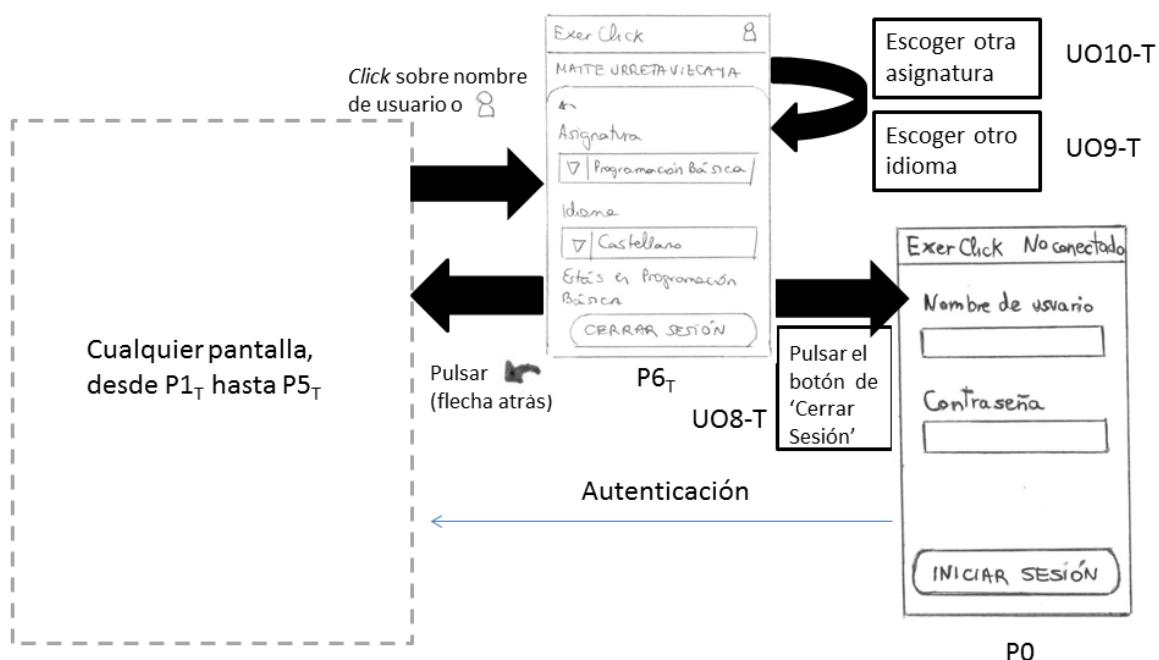


Figura 18: Autómata de estados correspondiente a los UOs 8-T, 9-T y 10-T

Capítulo 5

Diseño e Implementación

En este capítulo tenemos dos apartados importantes: el 5.1 en el que se muestran las interfaces y, en general, el lado cliente y el 5.2 en el que se muestra la lógica de negocio. Al final del capítulo se detalla la estructura de ficheros y carpetas del proyecto en el apartado 5.3 y los requisitos para ejecutar la aplicación en el apartado 5.4.

5.1. M-2: Interfaces o lado del cliente

En este apartado se mostrarán las interfaces desarrolladas partiendo de los modelos de requisitos (M-1) de cada objetivo y algunos elementos comunes de todos los objetivos.

5.1.1. Interfaz de autenticación de usuario

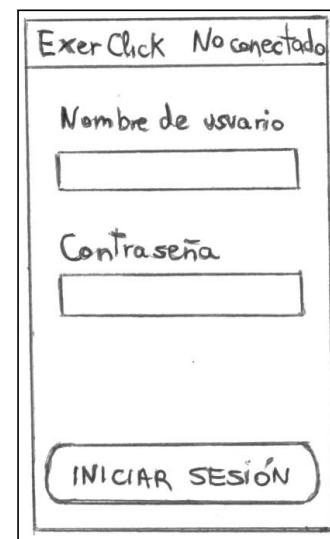
A modo de ejemplo se añade el prototipo en papel correspondiente a esta pantalla (figura 19a) para que se aprecie la diferencia de estilo.

La pantalla de autenticación de usuario (index.html) de la figura 19a es la primera que se muestra al iniciar la aplicación. Esta pantalla nos redirige a teacher.html o student.html (vista del profesor y del alumno, respectivamente) dependiendo del rol del usuario con el que nos identifiquemos (el rol está definido en la base de datos).

qClick [5] y *exerClick* comparten la misma base de datos y el sistema de autenticación de ambos es similar. Por ello, se decidió utilizar el mismo sistema de autenticación de



(a) Diseño (M-2)



(b) Prototipo en papel (M-1)

Figura 19: Pantalla de autenticación de usuario

qClick en exerClick y se reutilizó la mayor parte del código.

5.1.2. Interfaz del alumno

La pantalla del alumno (figura 20) le muestra a este el estado actual de la clase: los ejercicios activos, su progreso en esta sesión y la media de progreso de la clase.

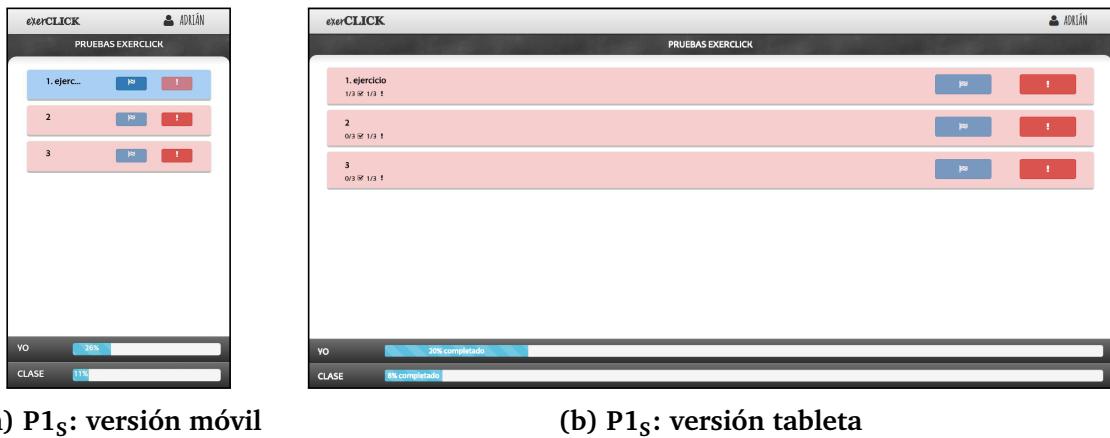


Figura 20: P1_s: vista del alumno

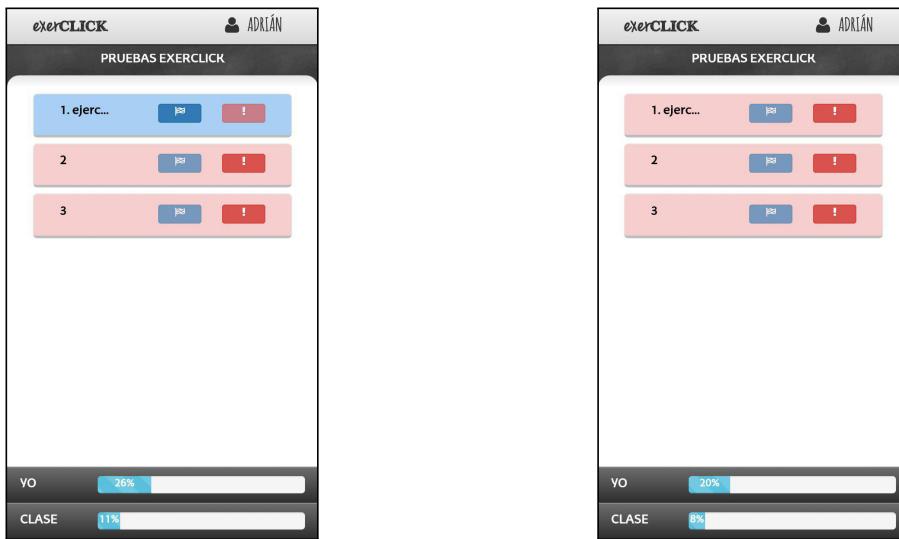
UO1-S: Responder a un ejercicio

En la figura 21 se muestra la pantalla principal del alumno, (P1_s). En la barra superior aparece el nombre de usuario e inmediatamente debajo el nombre de la asignatura de la que hay una sesión activa.

Las cajas son ejercicios activos en esa asignatura. Para que el estado de los ejercicios sea más visible las cajas tienen colores: el rojo es para una duda y el azul para un ejercicio acabado. Dentro de las cajas aparece el nombre de los ejercicios a la izquierda y a la derecha dos botones para marcar el estado en el que nos encontramos con cada ejercicio.

En la figura 21a cada uno de los ejercicios tiene un estado (finalizado o con duda) denotado por el botón que está activado (el otro al estar desactivado tiene menos opacidad). También se puede apreciar en el color del contenedor del ejercicio, si está en azul es que está marcado como finalizado (botón azul también iluminado y rojo apagado) y análogamente si el contenedor es rojo.

Para cambiar el estado de un ejercicio (UO1-T) debemos darle al botón activado/iluminado (es decir, quitar la marca que habíamos hecho) para que el ejercicio se quede sin estado (nada marcado, la caja toma un color gris). Cuando la caja está gris ambos



(a) P_{1S}: vista principal, se muestran los ejercicios (b) P_{1S}: cambio en el estado de un ejercicio

Figura 21: P_{1S}: cambio en el estado de un ejercicio

botones están iluminados y podemos, por tanto, marcar un nuevo estado (el que queramos). Se puede ver parte de la transición en la figura 21: el ejercicio "1º ejercicio" está marcado como acabado y en la figura 21b está marcado como duda al igual que el resto (habiéndolo estado sin marca previamente).

UO2-S: Ver detalles de un ejercicio

Al pulsar sobre un ejercicio podremos ver la pantalla P₂ del alumno. Los detalles del ejercicio sobre el que hemos pinchado se cargan en una pestaña, tal y como se ve en la figura 22. Tenemos un botón para ir atrás y cerrar la pestaña en la parte superior, junto al identificador del ejercicio. En el cuerpo de la pestaña aparecen los detalles de uno en uno, podemos bajar gracias al scroll para ver todos los detalles.

Un añadido respecto al M-1 es que P_{2S} es una ventana solapada a la interfaz P_{1S}, y por tanto se puede hacer click fuera de ella. Al hacerlo se cerrará automáticamente la ventana. Daremos por hecho de aquí en adelante que cualquier ventana solapada a una interfaz, que tenga un espacio fuera de ella para hacer click, puede cerrarse al hacer click fuera de la ventana.

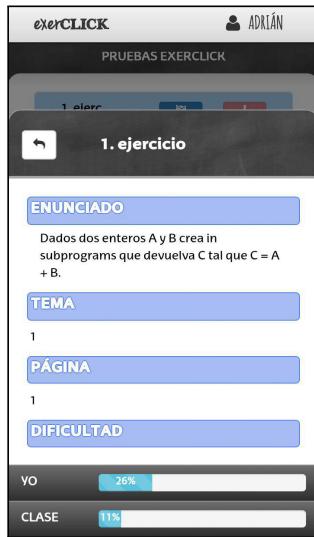


Figura 22: P2_S: ver los detalles de un ejercicio

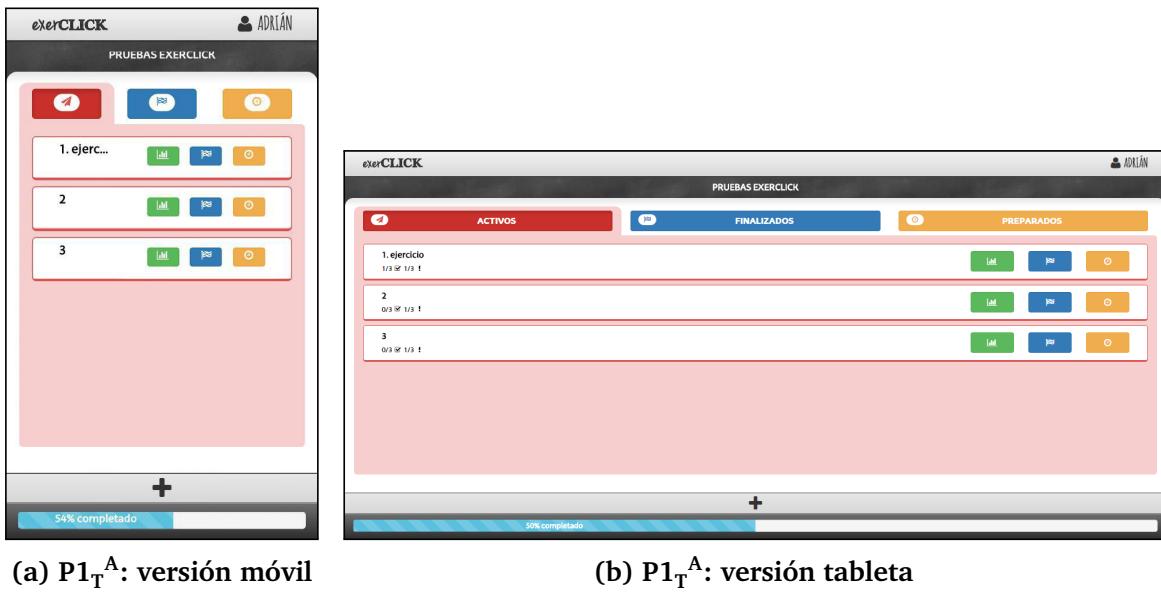
5.1.3. Interfaz del profesor

Cuando nos identificamos como profesor, si hay una clase activa, accederemos directamente a esta pantalla (teacher.html) que aparece en la figura 25. Además sirve como base de muchos UOs del profesor, ya que hay que pasar obligatoriamente por ella. En ella se muestra en la parte superior el nombre del usuario identificado y justo debajo el nombre de la asignatura activa.

En el centro tenemos tres botones de colores y una lista de ejercicios (si la hubiera). La lista de ejercicios que aparece al cargar esta interfaz por primera vez es la de ejercicios activos propuestos en clase (que los alumnos pueden ver). Estos ejercicios corresponden al botón rojo. Pulsando el botón azul se mostrarán los ejercicios finalizados y pulsando el botón amarillo los preparados (son ejercicios guardados que se quieren lanzar).

En cada caja de ejercicios veremos el nombre del ejercicio en la parte izquierda y en la derecha tres botones. El botón verde sirve para mostrar las estadísticas (UO4-T) en la pantalla P5_T y los otros dos cambian dependiendo del tipo de ejercicios que se muestren. Su función es cambiar el tipo del ejercicio (UO3-T). Así, un ejercicio activo, al pulsar el botón azul se cambiará a finalizado.

Finalmente, en la parte de abajo hay dos barras. La primera, con un '+', sirve para mostrar la pestaña para crear ejercicios nuevos (UO1-T, UO2-T y UO1). La barra inferior a esta muestra el progreso en la sesión, es decir: cuántos ejercicios han sido dados por

Figura 23: P1_T^A: Vista del profesor

finalizados respecto a los lanzados.

5.1.4. Interfaz del perfil del profesor

Es la interfaz de profile.html (figura 24), y la vista de profesor de inicio después de autenticarse si no hay ninguna sesión activa en marcha (no hay clase). Desde esta pantalla podemos cambiar algunas opciones como la asignatura activa (aparece un mensaje de cuál es la asignatura activa más abajo) o el idioma de la aplicación. El botón de "Ir a clase" nos devuelve a la pantalla P1_T (teacher.html) con la asignatura activa escogida. En la parte inferior hay un botón para cerrar sesión y volver a la pantalla de autenticación (P0).

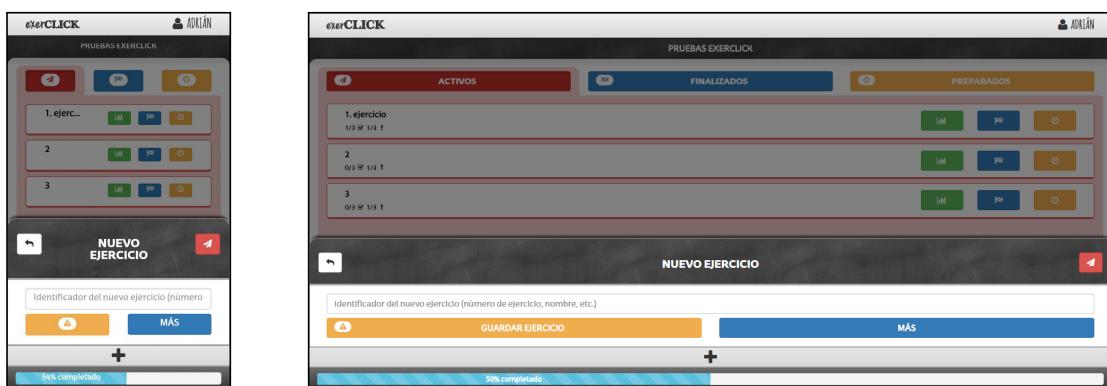
UO1-T: Crear-Lanzar un ejercicio simple

Al pulsar el botón con símbolo de '+' (plus/más) de la parte inferior de la interfaz del profesor se desplegará la pestaña que aparece en la figura 25a. En la parte superior tenemos la el botón con una flecha hacia atrás para cerrar la pestaña, el identificador del ejercicio y el botón para lanzar el ejercicio. Este botón sólo estará activo si estamos en clase, de otro modo sólo podremos guardar el ejercicio.



Figura 24: Perfil del profesor

En la parte central hay un campo para añadir el identificador del ejercicio (obligatorio). Hay dos botones debajo de este campo: para guardar el ejercicio como preparado y otro para mostrar la versión avanzada de crear ejercicios (con más detalles, UO2-T) con la pantalla P2_T^I.



(a) Interfaz del UO1-T: Crear-Lanzar un ejercicio simple

(b) Versión grande (tablet) de la pantalla del UO1-T (P2_T)Figura 25: Pantallas del UO1-T (P2_T)

UO2-T: Crear-Lanzar un ejercicio detallado

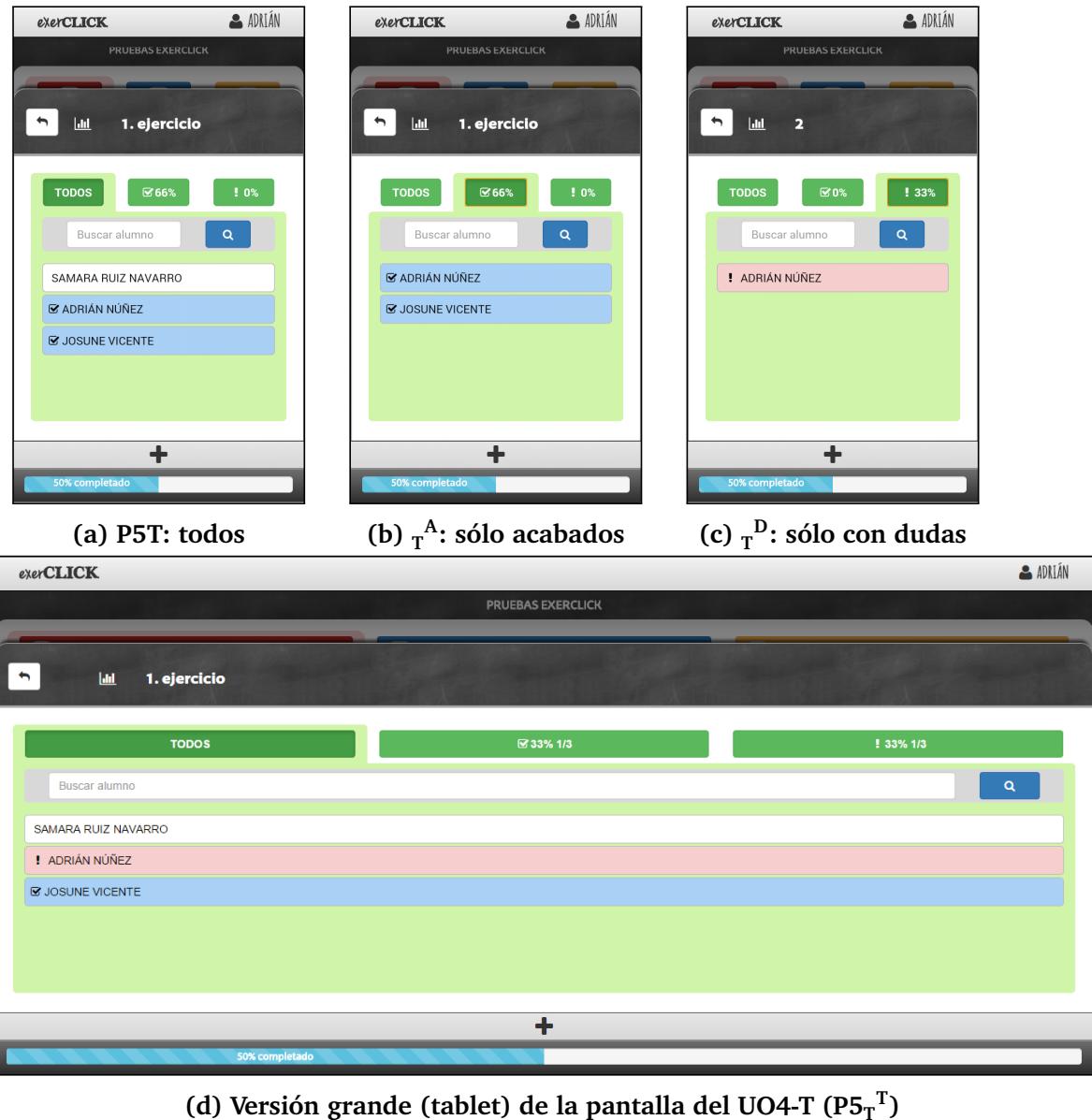
En la pantalla de la figura 26, correspondiente al P2_T¹, se añaden más detalles a parte del identificador (opcionales todos menos este último). Al pulsar sobre el botón "Menos" volveremos a P2_T.



Figura 26: Interfaz del UO2-T: Crear-Lanzar un ejercicio detallado

UO4-T: Ver estadísticas de un ejercicio

En la figura 27 vemos todas las pantallas asociadas a las estadísticas de los ejercicios. La parte superior de la nueva pestaña de esta pantalla es similar a la de crear nuevos ejercicios. En la parte central hay 3 botones verdes, un buscador y una lista de alumnos. Al cargar esta pestaña se cargarán todos los alumnos involucrados en el ejercicio (esto corresponde al primer botón). Los otros dos botones muestran todos los alumnos que han marcado el ejercicio como acabado y todos aquellos que han marcado una duda (en ese orden). El buscador nos permite buscar por nombre y/o apellido a cualquier persona (busca por subcadenas, así que no es necesario introducir el nombre o apellido completo) e irá actualizando la lista de alumnos.

Figura 27: Pantallas del UO4-T (P5_T)

UO5-T: Ver la descripción completa de un ejercicio

La pantalla para ver los detalles de un ejercicio ($P3_T$) se muestra en la figura 28. En la parte superior vemos un botón azul con engranajes que nos lleva a la pantalla $P4_T$ ($UO6-T$). En la parte central veremos los detalles del ejercicio que estamos viendo (si hubiera alguno). Si un ejercicio no tiene detalles aparecerá un mensaje indicando que no existen detalles asociados al ejercicio como en la figura 28b.

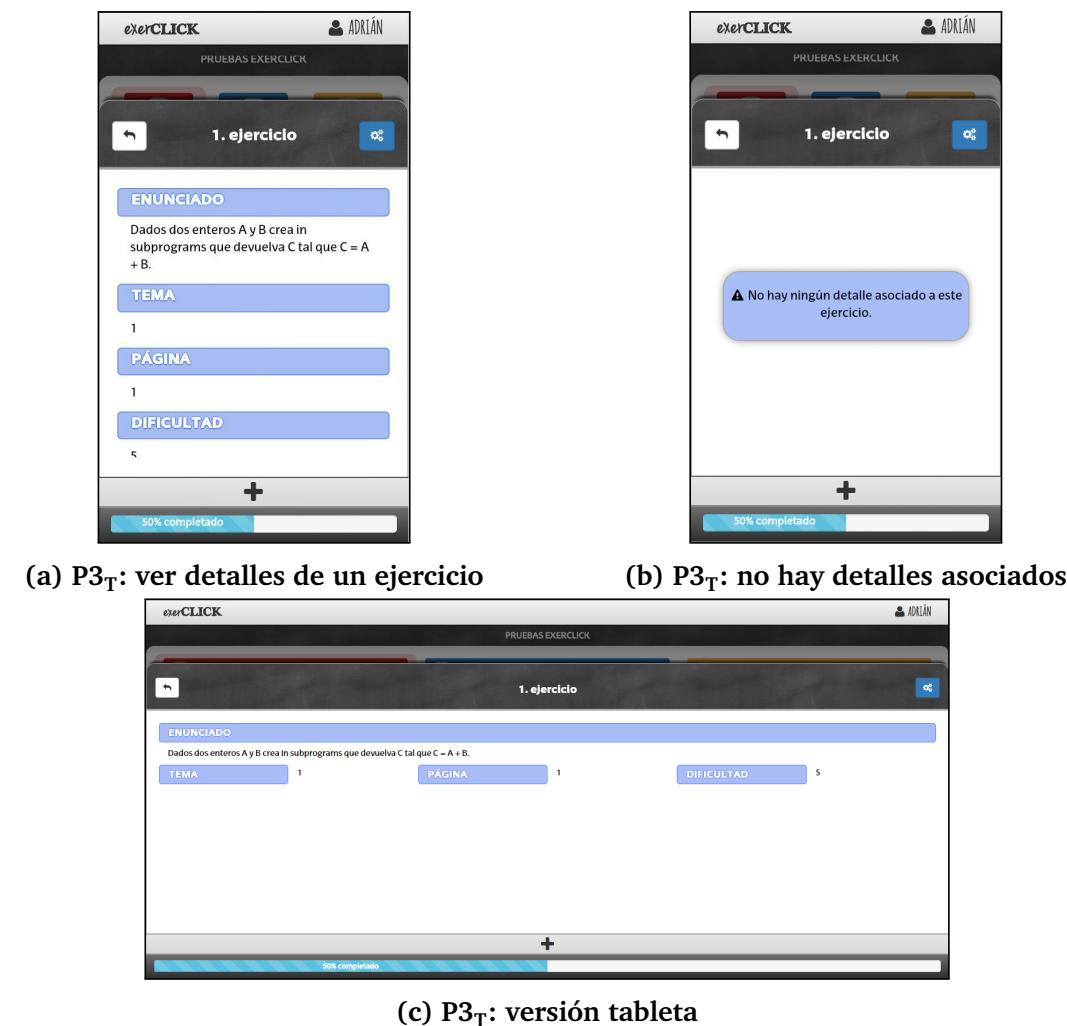


Figura 28: Pantallas del UO5-T ($P3_T$)

UO6-T: Editar un ejercicio

Para editar un ejercicio accederemos a la pantalla P4_T (figura 29). Para llegar a esta pantalla hay que pulsar sobre el botón de los engranajes en la pantalla P3_T. Los detalles que antes (en P3_T) eran texto plano ahora están dentro de un campo editable (incluido el identificador del ejercicio). Si volvemos a pulsar el botón de engranajes no se guardarán los cambios y volveremos a P3_T. Si pulsamos en el botón "Guardar cambios" que hay en la parte de abajo se guardarán y volveremos a P3_T con los datos modificados.

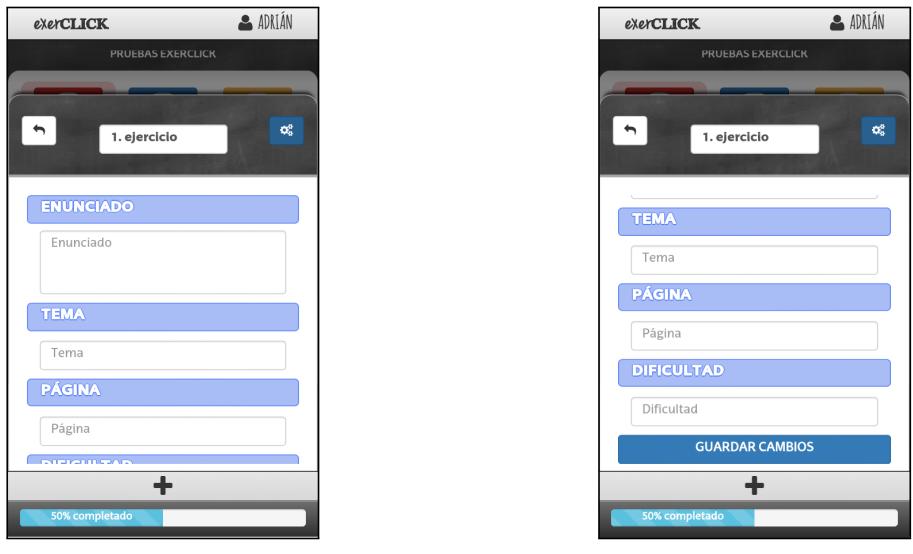


Figura 29: Pantallas del UO6-T (P4_T)

5.1.5. Uso de iconos mediante Font Awesome

Font Awesome [3] es un sitio web generado mediante un repositorio de Github. De este sitio se han obtenido todos los iconos de la aplicación: desde los iconos de los botones hasta el ícono de usuario que aparece junto al nombre de usuario.

Su uso es sencillo, para añadir cualquier ícono basta con añadir código como el de este ejemplo (para añadir el ícono del avión de papel de los ejercicios activos):

```
1 <i class="fa fa-paper-plane"></i>
```

Además podemos aumentar el tamaño del ícono añadiendo clases del tipo fa-2x, fa-3x, etc. (aumentan por 2 y por 3 el tamaño del ícono, respectivamente). También existe la opción de adaptarlo al texto que tiene cerca con fa-fw y otras tantas opciones que no se han llegado a utilizar en el proyecto (rotaciones, añadir un marco de prohibido encima, etc.). Se pueden encontrar ejemplos en la propia página.

5.2. M-3: Lógica de negocio o lado del servidor

En esta sección hablaremos sobre el código del lado del servidor (escrito en PHP) desde el apartado 5.2.1 hasta el 5.2.10 y sobre la base de datos en el apartado 5.2.11.

5.2.1. Configuración

Código 5.1 Autenticación del usuario.

```

1 <?php
2   define('HOST', 'xxxx');
3   // Database username
4   define('USER', 'xxxx');
5   // Database password
6   define('PASSWORD', 'xxxx');
7   // Database name
8   define('DATABASE', 'xxxx');
9
10  define("CAN_REGISTER", "any");
11  define("DEFAULT_ROLE", "member");
12 ?>

```

Este archivo de configuración define datos de autenticación en la base de datos. Se añade y utiliza en cada fichero mediante el siguiente código:

Código 5.2 Autenticación del usuario.

```

1 include 'mysqli-config.php';
2 $mysqli = new mysqli(HOST, USER, PASSWORD, DATABASE);
3
4 if ($mysqli_connect_errno()) {

```

```

5     echo "Failed to connect to MySQL: " .
6     mysqli_connect_error();
}

```

Este trozo de código se añade siempre al inicio de cada código PHP en el servidor. Al final se cierra la base de datos.

5.2.2. Identificación de usuario

El código 5.3 se ejecuta en la interfaz P0 al intentar autenticarse. No se puede tener más de una sesión abierta en diferentes dispositivos, sólo la última sesión es la que se puede utilizar. En la base de datos proporcionada teníamos la tabla *usersession* para almacenar las sesiones y mantener ese control. En esta tabla guardamos el ID de usuario global (almacenado en la tabla *fosUser*) y el identificador de sesión generado por PHP. Además, dependiendo del rol del usuario se le llevará a la interfaz teacher.html o student.html.

Código 5.3 Autenticación del usuario.

```

1 $name = test_input($_GET['Username']);
2 $pass = test_input($_GET['Password']);
3
4 // Check if username exists
5 $statement = 'SELECT * FROM fos_user WHERE username=' .
6     $name . '"';
7 $result = $mysqli->query($statement);
8
9 // Only 1 result, otherwise error
10 $num = mysqli_num_rows($result);
11 if($num != 1){
12     die();
13 }
14 $row = mysqli_fetch_assoc($result);
15 if(!isPasswordValid($row['password'], $pass, $row['salt'])){
16     die();
17 }
18 // Mandatory call for using $_SESSION array
19 session_start();
20 $_SESSION['name'] = $row['username'];

```

```
21 // Check for the role of the user
22 if (strpos($row['roles'], 'ROLE_STUDENT') !== false) {
23     $_SESSION['role'] = "ROLE_STUDENT";
24 }
25 if (strpos($row['roles'], 'ROLE_TEACHER') !== false) {
26     $_SESSION['role'] = "ROLE_TEACHER";
27 }
28 }
29
30 // Save the user id
31 $_SESSION['general_id'] = $row['id'];
32
33 // Find id based in role
34 switch($_SESSION['role']) {
35 case 'ROLE_STUDENT':
36     $statement = 'SELECT id, name, surname1, surname2 FROM
37         student WHERE fosUser=' . $row['id'] . '';
38     $next_location = 'student.html';
39     break;
40 case 'ROLE_TEACHER':
41     $statement = 'SELECT id, name, surname1, surname2 FROM
42         teacher WHERE fosUser=' . $row['id'] . '';
43     $next_location = 'teacher.html';
44     break;
45 }
46
47 $result = $mysqli->query($statement);
48 if (!$result) {
49     die('The query has encounter a problem: ' .
50         mysqli_error($mysqli));
51 }
52
53 // Only 1 result, otherwise error
54 $num = mysqli_num_rows($result);
55 if ($num != 1){
56     session_destroy();
57     die('No role asigned.');
58 }
```

5.2.3. Cargar la interfaz principal (P1 del profesor y P1 del alumno)

Es importante al cargar la interfaz saber si tenemos clase o no. Se utiliza principalmente el código 5.4 para ello. Si no hay ninguna guardada en la sesión abierta buscamos si en el día y hora en la que nos encontramos hay alguna clase. Si la hay guardamos el identificador de su *attendanceclass*. En caso de que haya una clase activa simplemente obtendremos datos de ella (hora de inicio, de final y día).

Si nos hemos autenticado con el rol de profesor también obtendremos todas sus asignaturas, de modo que podamos cargarlas en el perfil del profesor más adelante.

Código 5.4 Obtener información inicial para la carga de la interfaz.

```

1 if(!isset($_SESSION['class'])) {
2     $now = date('H:i');
3     $day = date('Y-m-d');
4     // Find all classes previous to the actual time
5     $statement = 'SELECT * FROM attendanceclass WHERE day
6         <= "' . $day . '" OR (day = "' . $day . '" AND
7         endHour < "' . $now . '") ORDER BY day DESC,
8         startHour DESC';
9
10
11    $class = -1;
12    while($row = mysqli_fetch_array($result)) {
13        // For the next class previous to the actual time
14        // find a session given by the logged teacher
15        $statement = 'SELECT session.* FROM session INNER
16            JOIN groupteacher ON session.ctGroup =
17                groupteacher.ctGroup WHERE session.id = "' .
18                $row['session'] . '" AND groupteacher.teacher=
19                "' . $_SESSION['id'] . '"';
20        if(!$session = $mysqli->query($statement)){
21            die('The query has encounter a problem: ' .
22                mysqli_error($mysqli));
23        }

```

```

18     $group = mysqli_fetch_array($session);
19     if(mysqli_num_rows($session) != 0){
20         break;
21     }
22 }
23 // Id of the session (next class)
24 $class = $row['id'];
25 } else {
26     $class = $_SESSION['class'];
27     $statement = 'SELECT * FROM attendanceclass WHERE id=
28         "' . $_SESSION['class'] . '"';
29
30     if(!$result = $mysqli->query($statement)){
31         die('The query has encounter a problem: ' .
32             mysqli_error($mysqli));
33     }
34
35     $row = mysqli_fetch_array($result);
36     $start = date('H:i', strtotime($row['startHour']));
37     $end = date('H:i', strtotime($row['endHour']));
38     $day = $row['day'];
39 }
40
41 if($_SESSION['role'] == 'ROLE_TEACHER') {
42     $statement = 'SELECT subject.id, subject.name AS
43         subject, subject.acronym, ctgroup.name, ctgroup.id
44         AS group_id FROM subject INNER JOIN ctgroup ON
45         ctgroup.subject = subject.id INNER JOIN
46         groupteacher ON groupteacher.ctgroup = ctgroup.id
47         WHERE groupteacher.teacher= "' . $_SESSION['id'] . '
48         " ORDER BY subject.name ASC';
49
50     if(!$subjects = $mysqli->query($statement)) {
51         die('The query has encounter a problem: ' .
52             mysqli_error($mysqli));
53     }
54     $subj = array();
55     while($row = mysqli_fetch_array($subjects)) {
56         $subj[] = array('name' => $row['subject'], ,
57             'acronym' => $row['acronym'], 'group' => $row[,
58                 'name'], 'group_id' => $row['group_id'], 'id'
59                 => $row['id']);
60     }
61 }
62 }
```

El segundo elemento principal al cargar las interfaces es la carga de los ejercicios (como se puede ver en el código 5.5). Se obtienen los ejercicios en los que estamos involucrados (porque estamos matriculados en esa asignatura como alumno o porque la impartimos como profesor). Después, independientemente del rol del usuario autenticado, obtenemos para cada ejercicio que hemos obtenido de la base de datos: el número de alumnos que lo han marcado como acabado, el número de ellos que han marcado una duda y el número de alumnos total involucrados en el ejercicio.

Código 5.5 Cargar los ejercicios del tipo pasado por parámetro.

```

1 if($_SESSION['role'] == 'ROLE_TEACHER') {
2     $statement = 'SELECT exercise.id, description FROM
3         exercise LEFT JOIN ctgroup ON exercise.ctGroup =
4             ctgroup.id INNER JOIN groupteacher ON groupteacher.
5                 ctgroup = ctgroup.id '
6                     'WHERE type = "' . $_GET['Type']
7                         ' AND ctgroup.id = "' .
8                             $_SESSION['group_id'] . '"'
9                             AND ctgroup.subject = "' .
10                            $_SESSION['subject_id'] . '"'
11                            AND groupteacher.teacher = "' .
12                                $_SESSION['id'] . '"';
13 } else if($_SESSION['role'] == 'ROLE_STUDENT') {
14     $statement = 'SELECT exercise.id, description, state
15         FROM exercise INNER JOIN ctgroup ON ctgroup.id =
16             exercise.ctGroup INNER JOIN groupstudent ON
17                 groupstudent.ctgroup = exercise.ctgroup '
18                     'INNER JOIN exercisestate ON
19                         exercise.id = exercisestate.
20                             idexercise WHERE type = "'
21                             Active" AND exercisestate.
22                             idstudent = "' . $_SESSION['
23                                 id'] . '" AND ctgroup.subject
24                                     = "' . $_SESSION['subject_id
25                                         '] . '" AND exercise.ctGroup
26                                         = "' . $_SESSION['group_id'] .
27                                             '" AND groupstudent.student
28                                                 = "' . $_SESSION['id'] . '"';
29 }
30
31 $result = $mysqli->query($statement);

```

```

10 $exercises = array();
11 while ($row = $result->fetch_assoc()) {
12     $statement = 'SELECT * FROM exercise INNER JOIN
13         exercisestate ON exercise.id = exercisestate.
14         idexercise INNER JOIN ctgroup ON ctgroup.id =
15         exercise.ctGroup ,
16             'WHERE exercise.id = "' . $row['
17             id'] . '" AND state = "
18             Finished" AND ctgroup.id = "' .
19             $_SESSION['group_id'] . '"
20             AND ctgroup.subject = "' .
21             $_SESSION['subject_id'] . '",'
22             ;
23     $result2 = $mysqli->query($statement);
24
25     $statement = 'SELECT * FROM exercise INNER JOIN
26         exercisestate ON exercise.id = exercisestate.
27         idexercise INNER JOIN ctgroup ON ctgroup.id =
28         exercise.ctGroup ,
29             'WHERE exercise.id = "' . $row['
30             id'] . '" AND state = "
31             Question" AND ctgroup.id = "' .
32             $_SESSION['group_id'] . '"
33             AND ctgroup.subject = "' .
34             $_SESSION['subject_id'] . '",'
35             ;
36     $result3 = $mysqli->query($statement);
37
38     $statement = 'SELECT * FROM exercise INNER JOIN
39         attendanceclass ON exercise.launched =
40         attendanceclass.id INNER JOIN
41         attendanceclassstudent ON attendanceclassstudent.
42         attendanceclass = attendanceclass.id WHERE exercise
43         .id = "' . $row['id'] . '"';
44     $result4 = $mysqli->query($statement);
45
46     $row['nofinished'] = mysqli_num_rows($result2);
47     $row['noquestions'] = mysqli_num_rows($result3);
48     $row['num'] = mysqli_num_rows($result4);
49
50     $exercises[] = array('exercise' => $row);
51 }

```

Ambas interfaces principales también comparten las barras de progreso de la clase. Para obtener la información para cargarlas se ejecuta el código 5.6 en el caso del profesor y el 5.7 en el caso del alumno. Nos devuelven el porcentaje de carga para cada barra mediante simples divisiones.

Código 5.6 Obtener información para la carga de la barra de progreso del profesor.

```

1 $statement = 'SELECT * FROM exercise INNER JOIN ctgroup ON
    ctgroup.id = exercise.ctGroup ' .
    'WHERE ctgroup.id = "' .
    $_SESSION['group_id'] . '" .
    AND ctgroup.subject = "' .
    $_SESSION['subject_id'] . '" .
    AND exercise.type != "Ready" .
    ;
2
3 $result = $mysqli->query($statement);
4 $total = mysqli_num_rows($result);
5
6 $statement = 'SELECT * FROM exercise INNER JOIN ctgroup ON
    ctgroup.id = exercise.ctGroup ' .
    'WHERE ctgroup.id = "' .
    $_SESSION['group_id'] . '" .
    AND ctgroup.subject = "' .
    $_SESSION['subject_id'] . '" .
    AND exercise.type = "Finished"
    " ;
7
8 $result = $mysqli->query($statement);
9 $finished = mysqli_num_rows($result);

```

Código 5.7 Obtener información para la carga de la barra de progreso del estudiante.

```

1 $statement = 'SELECT * FROM exercise INNER JOIN ctgroup ON
    ctgroup.id = exercise.ctGroup INNER JOIN exercisestate
    ON exercisestate.idexercise = exercise.id ' .
    'WHERE ctgroup.id = "' .
    $_SESSION['group_id'] . '" .
    AND ctgroup.subject = "' .
    $_SESSION['subject_id'] . '" .
    AND exercisestate.idstudent =
    " . $_SESSION['id'] . '"';
2
3 $result = $mysqli->query($statement);

```

```
4 $total = mysqli_num_rows($result);
5
6 $statement      = 'SELECT * FROM exercise INNER JOIN
7   ctgroup ON ctgroup.id = exercise.ctGroup INNER JOIN
8   exercisestate ON exercisestate.idexercise = exercise.id
9   ,
10
11          'WHERE ctgroup.id = "' . $_SESSION['group_id'] . '" .
12          AND ctgroup.subject = "' . $_SESSION['subject_id'] . '" .
13          AND exercisestate.idstudent =
14          "' . $_SESSION['id'] . '" .
15          AND exercisestate.state = "Finished";
16
17 $result = $mysqli->query($statement);
18 $finished = mysqli_num_rows($result);
19
20
21 $studentprogress = ($total == 0) ? 0 : intval($finished *
22   100 / $total);
23
24
25 $statement = 'SELECT * FROM attendanceclassstudent INNER
26   JOIN attendanceclass ON attendanceclass.id =
27   attendanceclassstudent.attendanceclass INNER JOIN
28   session ON attendanceclass.session = session.id INNER
29   JOIN ctgroup ON ctgroup.id = session.ctGroup INNER JOIN
30   exercise ON exercise.ctGroup = ctgroup.id INNER JOIN
31   exercisestate ON exercisestate.idexercise = exercise.id
32   ,
33
34          'WHERE ctgroup.id = "' . $_SESSION['group_id'] . '" AND ctgroup.subject
35          = "' . $_SESSION['subject_id'] . '" .
36          "' ;
37
38 $result = $mysqli->query($statement);
39 $total = mysqli_num_rows($result);
40
41
42 $statement = 'SELECT * FROM attendanceclassstudent INNER
43   JOIN attendanceclass ON attendanceclass.id =
44   attendanceclassstudent.attendanceclass INNER JOIN
45   session ON attendanceclass.session = session.id INNER
46   JOIN ctgroup ON ctgroup.id = session.ctGroup INNER JOIN
47   exercise ON exercise.ctGroup = ctgroup.id INNER JOIN
48   exercisestate ON exercisestate.idexercise = exercise.id
49   ,
50
51          'WHERE ctgroup.id = "' . $_SESSION['group_id'] . '" AND ctgroup.subject
52          = "' . $_SESSION['subject_id'] . '" .
53          "' ;
```

```

19          'WHERE ctgroup.id = "' . $_SESSION['
20          group_id'] . '" AND ctgroup.
21          subject = "' . $_SESSION['
22          subject_id'] . '" AND
23          exercisestate.state = "Finished"';
24 $result = $mysqli->query($statement);
25 $finished = mysqli_num_rows($result);

```

5.2.4. Responder a un ejercicio (UO1-S)

Un alumno puede marcar un ejercicio como acabado o con una duda. Este nuevo estado se pasa por parámetro y se procesa con el código 5.8.

Si el ejercicio estaba con una duda marcada y la quitamos inmediatamente debemos guardarla en la base de datos como una duda resuelta (estas dudas resueltas se muestran en las estadísticas de los ejercicios finalizados). Si el ejercicio ya tenía una duda resuelta anteriormente por el estudiante se actualiza y se guarda la última.

Código 5.8 Responder a un ejercicio.

```

1 $idexercise = $_GET['Idexercise'];
2 $idstudent = $_SESSION['id'];
3 $state = $_GET['State'];
4
5 $statement = 'SELECT state FROM exercisestate WHERE
6     idexercise = "' . $idexercise . '" AND idstudent = "' .
7     $idstudent . '"';
8 $result = $mysqli->query($statement);
9 $row = $result->fetch_assoc();
10 if($row['state'] == 'Question' && ($state == 'Nothing'
11     $state == 'Finished')) {
12     $statement = 'SELECT * FROM exercise_solved_questions
13         WHERE idexercise = "' . $idexercise . '" AND
14             idstudent = "' . $idstudent . '"';
15     $result = $mysqli->query($statement);
16
17     $day = date('Y-m-d');
18     $time = date('H:i:s');
19     if(mysqli_num_rows($result) == 1) {
20         $statement = 'UPDATE exercise_solved_questions
21             SET day = "' . $day . '" AND time = "' . $time

```

```

        . . . WHERE idexercise = '' . $idexercise . '';
        AND idstudent = '' . $idstudent . '';
    $mysqli->query($statement);
} else {
    $statement = 'INSERT INTO
        exercise_solved_questions (idexercise,
        idstudent, day, time) VALUES ("' . $idexercise
        . '", "' . $idstudent . '", "' . $day . '",'
        . $time . ')';
    $mysqli->query($statement);
}
$stmt = 'UPDATE exercisestate SET state = '' . $state
        . '' WHERE idexercise = '' . $idexercise . '' AND
        idstudent = '' . $idstudent . '';
$result = $mysqli->query($stmt);

```

5.2.5. Ver detalles de un ejercicio (UO2-S y UO5-T)

Al cargar la interfaz P3 (profesor) o P2 (alumno) para ver los detalles de un ejercicio en concreto se cargan a su vez varios datos. Para obtener estos datos se usa el código 5.9. Mediante el identificador del ejercicio se obtienen todos los campos de un ejercicio y se devuelven. De esta forma desde javascript se pueden añadir los detalles que no sean nulos o vacíos.

Código 5.9 Ver detalles de un ejercicio.

```

1 $statement = 'SELECT * FROM exercise WHERE id = '' . $_GET[
    'Id'] . '';
2 $result = $mysqli->query($statement);
3 $row = $result->fetch_assoc();

```

5.2.6. Crear-Lanzar un ejercicio (UO1-T y UO2-T)

El código 5.10 implementa la función de lanzar ejercicios, ya sean simples o avanzados (con más detalles). En principio se añade un ejercicio en la base de datos con todos sus campos siempre. Cuando creamos un ejercicio simple solo le pasamos (como detalle) el identificador. Al crear un ejercicio avanzado se le pueden pasar más detalles.

Código 5.10 Crear-lanzar un ejercicio.

```

1 $statement = 'INSERT exercise(ctGroup, type, launched,
2   description, statement, topic, page, difficulty) VALUES
3   ("' . $_SESSION['group_id'] . '", "", "' . $_GET['State'] .
4   '", "' . $_SESSION['attendanceclass'] . '", "", "' . $_GET['
5   Description'] . '", "' . $_GET['Statement'] . '", "", "' .
6   $_GET['Topic'] . '", "", "' . $_GET['Page'] . '", "", "' . $_GET
7   ['Difficulty'] . '")';
8 $mysqli->query($statement);
9
10 $idexercise = $mysqli->insert_id;
11
12 $statement = 'SELECT student FROM groupstudent WHERE
13   groupstudent.ctGroup = "' . $_SESSION['group_id'] . '"';
14 $result = $mysqli->query($statement);
15
16 while ($row = $result->fetch_assoc()) {
17   $statement = 'INSERT exercisestate (idstudent,
18     idexercise, state) VALUES (' . $row['student'] . ' ,
19     "' . $idexercise . '", "Nothing")';
20   $mysqli->query($statement);
21 }
```

5.2.7. Cambiar detalles de un ejercicio (UO3-T y UO6-T)

El código 5.11 actualiza el estado de un ejercicio con los parámetros que se le pasan, entre ellos el tipo. Dependiendo de los parámetros pasados este código servirá para implementar el UO3-T (cambiar el tipo de un ejercicio) o el UO6-T (editar un ejercicio). Si sólo le pasamos el tipo ejecutará el código correspondiente al UO3-T (línea 2). De otro modo ejecutará la parte correspondiente al UO6-T (línea 13).

Código 5.11 Cambiar el tipo de un ejercicio.

```

1 if(!isset($_GET['Description']))    $_GET['Description'] ==
2   null) {
3
4   $statement = 'SELECT type FROM exercise WHERE id = "' .
5   . $_GET['Id'] . '"';
6   $result = $mysqli->query($statement);
7   $row = $result->fetch_assoc();
8   if($row['type'] == 'Ready') {
```

```

7      $statement = 'UPDATE exercise SET type = "' .
     $_GET['Type'] . '", launched = "' . $_SESSION[
      'attendanceclass'] . '" WHERE id = "' . $_GET[
      'Id'] . '"';
8  } else {
9      $statement = 'UPDATE exercise SET type = "' .
     $_GET['Type'] . " WHERE id = '" . $_GET['Id'] .
     . '"';
10 }
11 }
12 } else {
13
14     $statement = 'UPDATE exercise SET description = "' .
     $_GET['Description'] . '", statement = "' . $_GET[,
Statement'] . '", topic = "' . $_GET['Topic'] . '',
page = "' . $_GET['Page'] . '", difficulty = "' .
     $_GET['Difficulty'] . '" WHERE id = "' . $_GET['Id'] .
     ] . '"';
15 }

```

5.2.8. Ver estadísticas un ejercicio (UO4-T)

El código 5.12 ejecuta una consulta sobre la base de datos para obtener para un ejercicio en concreto el nombre completo de cada alumno que está involucrado y su estado en él (si lo ha acabado o tiene una duda). Además permite filtrar por nombre o apellido gracias al Key pasado por parámetro (utilizando el operador LIKE de SQL). El resultado será mostrado en la interfaz P5.

Código 5.12 Obtener estadísticas de un ejercicio.

```

1 $statement = 'SELECT name, surnames, state FROM student
   INNER JOIN exercisestate ON student.id = exercisestate.
   idstudent WHERE exercisestate.idexercise = "' . $_GET[,
   Id'] . '" AND state LIKE "' . $_GET['State'] . '" AND (
   name LIKE "' . $_GET['Key'] . '" OR surnames LIKE "' .
   $_GET['Key'] . '")';
2 $result = $mysqli->query($statement);
3
4 $statistics = array();
5 while ($row = $result->fetch_assoc()) {
6     $statistics[] = array('statistic' => $row);

```

7 }

Al visualizar las estadísticas de un ejercicio, en cada pestaña se ven los porcentajes de alumnos que han marcado el ejercicio como acabado o que han marcado una duda. Para obtener esos porcentajes se llama al código 5.13.

Código 5.13 Obtener porcentajes para las estadísticas e un ejercicio.

```

1 $data = array();
2
3 $statement = 'SELECT * FROM exercisestate INNER JOIN
4     student ON exercisestate.idstudent = student.id WHERE
5     exercisestate.idexercise = ' . $_GET['Id'];
6 $result = $mysqli->query($statement);
7 $total = mysqli_num_rows($result);
8 $data['total'] = $total;
9 $result->free();
10
11 $statement = 'SELECT * FROM exercisestate INNER JOIN
12     student ON exercisestate.idstudent = student.id WHERE
13     state LIKE "Finished" AND exercisestate.idexercise = ' .
14     $_GET['Id'];
15 $result = $mysqli->query($statement);
16 $finished = mysqli_num_rows($result);
17 $data['finished'] = $finished;
18 $result->free();
19
20 $statement = 'SELECT * FROM exercisestate INNER JOIN
21     student ON exercisestate.idstudent = student.id WHERE
22     state LIKE "Question" AND exercisestate.idexercise = ' .
23     $_GET['Id'];
24 $result = $mysqli->query($statement);
25 $question = mysqli_num_rows($result);
26 $data['question'] = $question;

```

Cuando un ejercicio se da por finalizado, al mostrar sus estadísticas veremos una diferencia en el apartado de dudas. Se mostrarán 3 pestañas extra de filtro: mostrar todas las dudas en el ejercicio, mostrar las que no han sido resueltas o mostrar las que alguna vez fueron resueltas (pero luego se marcaron como acabado o quedaron sin marcar). Para poder obtener las estadísticas de esos ejercicios se han utilizado 3 consultas SQL diferentes, algo más complejas. El código 5.14 devuelve una de las 3 dependiendo del valor del parámetro *Question_State* que le pasemos.

Código 5.14 Mostrar todas las dudas, las no resueltas o las resueltas.

```

1 if($_GET['Question_State'] == 'All') {
2     $statement = 'SELECT DISTINCT * FROM student INNER
3         JOIN exercisestate ON student.id = exercisestate.
4             idstudent WHERE exercisestate.idexercise = "' .
5                 $_GET['Id'] . '" AND exercisestate.state = "
6                     Question" AND (name LIKE "' . $_GET['Key'] . '" OR
7                         surnames LIKE "' . $_GET['Key'] . '")';
8 } else if($_GET['Question_State'] == 'Solved') {
9     $statement = 'SELECT DISTINCT * FROM student INNER
10        JOIN exercisestate ON student.id = exercisestate.
11            idstudent INNER JOIN exercise_solved_questions ON
12                exercisestate.idexercise =
13                    exercise_solved_questions.idexercise AND
14                    exercise_solved_questions.idstudent = student.id
15                    WHERE exercisestate.idexercise = "' . $_GET['Id'] .
16                        '" AND exercisestate.state = "Finished" AND (name
17                            LIKE "' . $_GET['Key'] . '" OR surnames LIKE "' .
18                                $_GET['Key'] . '")';
19 } else if($_GET['Question_State'] == 'NotSolved') {
20     $statement = 'SELECT DISTINCT * FROM student INNER
21         JOIN exercisestate ON student.id = exercisestate.
22             idstudent LEFT JOIN exercise_solved_questions ON
23                 exercisestate.idexercise =
24                     exercise_solved_questions.idexercise AND
25                     exercise_solved_questions.idstudent = student.id
26                     WHERE exercisestate.idexercise = "' . $_GET['Id'] .
27                         '" AND exercisestate.state = "Question" AND (name
28                             LIKE "' . $_GET['Key'] . '" OR surnames LIKE "' .
29                                 $_GET['Key'] . '") AND exercise_solved_questions.id
30                         IS NULL';
31 }
32 $result = $mysqli->query($statement);
33
34 $statistics = array();
35 while ($row = $result->fetch_assoc()) {
36     $statistics[] = array('statistic' => $row);
37 }
```

5.2.9. Cambiar de idioma (UO9-T)

Para cambiar el idioma se llama a este pequeño trozo de código y se devuelve el resultado.

Código 5.15 Cambiar el idioma de la aplicación.

```
1 $_SESSION[ 'lang' ] = $_GET[ 'Language' ];
```

5.2.10. Cambiar de idioma (UO10-T)

Se cambian los parámetros de la sesión para reflejar que vamos a cambiar la asignatura activa. Esta asignatura es la que se muestra cuando volvemos a clase desde el perfil (de la que se muestran ejercicios). Los parámetros que definen la asignatura (*Subject, Id, Group_Id* y *Group*) vienen de la interfaz, donde son almacenados gracias a los atributos "data-*" de HTML5.

Código 5.16 Cambiar la asignatura activa.

```
1 $_SESSION[ 'subject' ] = $_GET[ 'Subject' ];
2 $_SESSION[ 'subject_id' ] = $_GET[ 'Id' ];
3 $_SESSION[ 'group_id' ] = $_GET[ 'Group_id' ];
4 $_SESSION[ 'group' ] = $_GET[ 'Group' ];
```

5.2.11. Base de datos

El diseño de la base de datos se ha realizado considerando la base de datos utilizada en PresenceClick (ver figura 35 del apéndice C). En ella hay muchas tablas que no nos han interesado usar en este proyecto, se han utilizado sólo unas pocas.

Se han incorporado unas tablas nuevas a la base de datos ya existente para la gestión de ejercicios tal y como se muestra en la figura 34 del apéndice C), donde aparecen algo tablas como *student* de PresenceClick relacionadas a tablas nuevas de *exerClick*. Las tablas añadidas son:

- **exercise:** contiene los detalles de un ejercicio, el grupo al que pertenece y el estado del ejercicio.
- **exercisestate:** guarda el estado de un ejercicio de un alumno concreto.

- **exerciseattendanceclass:** relaciona un ejercicio con la sesión de ejercicios (*attendanceclass*) en la que se lanzó el ejercicio.
- **exercise_solved_questions:** se guarda el momento (día y hora) en el que una duda de un alumno en un ejercicio ha sido resuelta.

Se utiliza el prefijo "exercise" para denotar que son relativas a *exerClick*.

5.3. Estructura del proyecto

En este apartado se revisara la estructura del proyecto, es decir, los ficheros que componen el proyecto y su estructura de carpetas. En la figura 30 se muestra la raíz del proyecto:

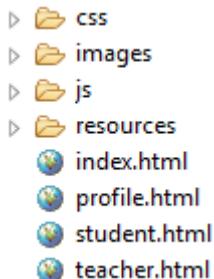


Figura 30: Estructura de archivos: raíz del proyecto

El proyecto cuenta con 4 carpetas principales: css, js, images y resources. Además contiene 4 ficheros HTML:

- **index.html:** funciona como pantalla de identificación de usuario y permite redirigir a las vistas principales (teacher.html o student.html) dependiendo del rol del usuario identificado.
- **student.html:** es la vista del alumno.
- **teacher.html:** es la vista del profesor.
- **profile.html:** pantalla de perfil del profesor, únicamente accesible desde teacher.html.

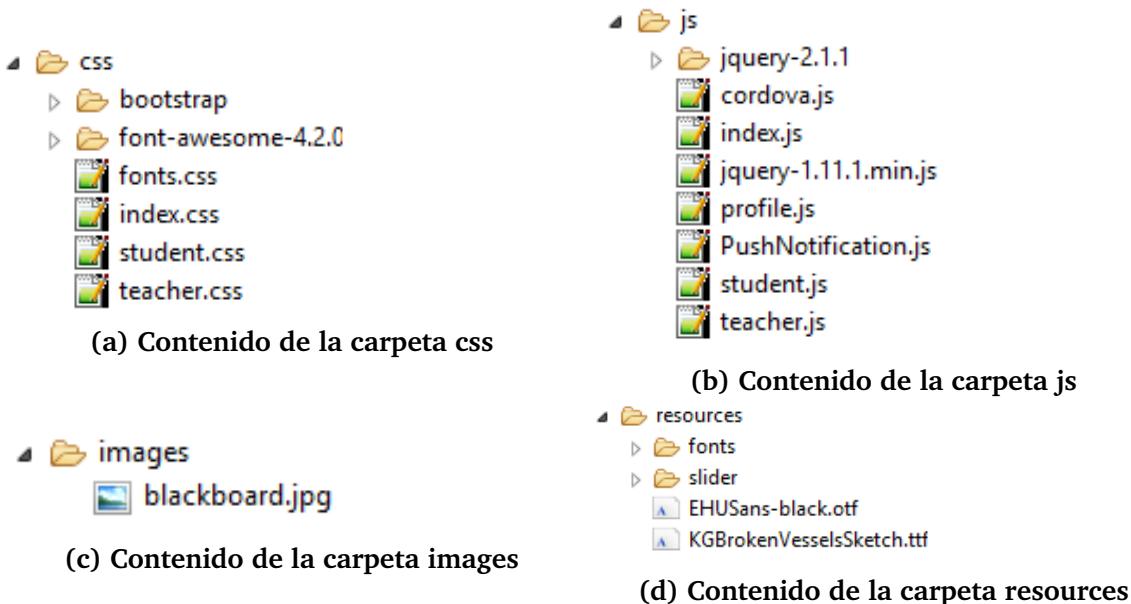


Figura 31: Contenido de las carpetas principales de la raíz del proyecto

Estos ficheros muestran las interfaces principales de la aplicación. Dentro de las otras carpetas encontramos más ficheros.

Los archivos de CSS sirve para darle estilo a las interfaces. Cada interfaz tiene un archivo .css asociado (excepto profile.html, que comparte teacher.css con teacher.html). Además, se añaden dos carpetas extra: la que necesitamos para Boostrap [2] y para usar los iconos de Font Awesome [3].

Los ficheros Javascript de la carpeta JS le dan dinamicidad a la aplicación. Controlan los clicks, hacen aparecer las pestañas nuevas y cargan contenido dinámicamente mediante AJAX. Cada interfaz (fichero HTML) tiene asociado su propio fichero JS. Además se incluye el fichero cordova.js y los ficheros necesarios para utilizar JQuery.

La carpeta images contiene las imágenes utilizadas en el proyecto y la carpeta resources contiene otros elementos utilizados en el proyecto (en este caso fuentes utilizadas).

5.4. Requisitos de dispositivos para su ejecución

La aplicación está pensada para el uso en cualquier dispositivo móvil con los sistemas operativos Android e iOS. No está pensada para tamaños de pantalla excesivamente pe-

queños, donde probablemente la aplicación se vería incorrectamente.

En Android la versión mínima requerida es la 4.0 (API 15). En versiones 5.0 o superior no se garantiza que funcione correctamente.

Capítulo 6

Conclusiones y Líneas futuras

exerClick es la aplicación para el seguimiento de ejercicios en el aula desarrollada en este proyecto. Su objetivo principal es dar una visión más real de lo que hacen los alumnos (tanto una visión global del grupo como individual), de modo que el docente pueda ofrecer un aprendizaje más adaptado e individualizado, aunque los grupos de alumnos sean muy grandes.

Se planteó al inicio como una aplicación web, al igual que su antecesor, qClick [5], una aplicación de pregunta-respuesta en un entorno docente. Se decidió cambiar a una aplicación móvil finalmente, conllevando el cambio en la tecnología principal: *Apache Cordova*. Las tecnologías principales (tecnologías web) se mantuvieron intactas gracias a Cordova: HTML5, CSS3 y Javascript. Esto permitió realizar una aplicación móvil manteniendo las tecnologías acordadas al inicio del proyecto.

El proyecto partió de algo sencillo como idea inicial (objetivo de las primeras iteraciones): una aplicación sencilla de lanzar un ejercicio simple, que el alumno envíe su *feedback* (el estado en el que se encuentra su ejercicio) y el profesor pueda visualizar esa respuesta (tanto de forma individual como grupal, aportando una visión global). Sin embargo, durante la captura de requisitos inicial se vio que había muchas ideas sobre la mesa que podrían ser interesantes, algunas descartadas por requerir quizás demasiado tiempo o ser demasiado ambiciosas. Esto, aun así, aportó gran inspiración para el desarrollo de la aplicación, pudiendo añadir más funcionalidad de forma incremental.

El proyecto fue lento al principio debido a la obligación del aprendizaje de la tecnología *Apache Cordova* (y la adaptación de ésta a móviles) y Responsive Web Design. En esta etapa se intentó perfeccionar el objetivo más básico de lanzar un ejercicio, especialmente su interfaz, que había que adaptar muy bien a móviles. Se quería crear también

algo reutilizable, de modo que las siguientes interfaces fueran más sencillas de crear. Esto creo un *boom* importante en las fases finales del proyecto, donde se consiguieron crear muchas interfaces y cumplir muchos objetivos en un corto periodo de tiempo.

Desde el inicio del proyecto se pensó en tener alumnos evaluando la aplicación, dando un *feedback* más cercano y constante. Debido a lo tarde que apareció una versión útil de la aplicación (que los usuarios pudieran manejar) los alumnos no probaron la aplicación hasta fechas tardías. Incluso hasta poder generar el fichero APK sólo se hicieron pruebas en un único móvil en el que estaba instalada la aplicación (en el anexo B se pueden ver los resultados de esa prueba). Después se les pasó el fichero APK para poder probarla en sus propios móviles, de cara a ver como se adaptaba la aplicación a diferentes dispositivos y de tener un *feedback* más continuo. En este punto no se llevaron acabo evaluaciones presenciales, sólo se recibían opiniones de pruebas por petición del autor o por iniciativa propia de los alumnos. Pese a que no se llevó acabo como fue planeado, las aportaciones recibidas por los usuarios fueron vitales para ciertos errores o para mejorar aspectos importantes de la aplicación.

6.1. Objetivos alcanzados

Como se ha comentado, hubo muchas ideas descartadas al inicio por ser demasiado ambiciosas o salirse del alcance previsto del proyecto. Entre los objetivos seleccionados, se han conseguido llevar a cabo satisfactoriamente todos excepto uno, que se comenta en el apartado 3.4.1.

Objetivos del Alumno

UO1-S: *Responder a un ejercicio.* El alumno quiere indicar que ha acabado o que tiene dudas con un ejercicio que el profesor ha propuesto. Para ello, se ha pensado en que el alumno puede marcar el ejercicio con uno de los dos estados. A modo de *feedback*, el profesor recibirá el estado (acabado o con dudas) de ese alumno.

UO2-S: *Ver detalles de un ejercicio.* El alumno quiere ver más detalles sobre un ejercicio disponible y activo en la sesión. Al pinchar sobre un ejercicio puede ver cualquier detalle extra que el profesor haya decidido añadir.

Objetivos del Profesor

UO1-T: *Crear-Lanzar un ejercicio simple.* El profesor quiere proponer un ejercicio rápidamente, sin escribir mucho. Ésto resulta especialmente útil cuando se tiene claro cuales

son los ejercicios que se mencionan (se sabe el tema, la página... el contexto de qué ejercicios se están realizando es claro) y cuando se quieren lanzar rápidamente sin perder mucho tiempo.

UO2-T: *Crear-Lanzar un ejercicio detallado.* El profesor quiere proponer la realización de un ejercicio preparado previamente o con bastantes detalles. Al contrario que en el UO1-T, en este caso el profesor se los puede preparar fuera de una sesión, incluso llegando a inventarse un ejercicio.

UO3-T: *Cambiar el tipo de ejercicio.* El profesor desea cambiar un ejercicio del tipo que tiene a otro cualquiera (de activo a finalizado, por ejemplo). Fundamental cuando se desea lanzar un ejercicio que estaba guardado mientras se preparaba mejor, cuando queremos dar un ejercicio por finalizado o cuando nos confundimos en cualquier cambio y queremos deshacerlo.

UO4-T: *El profesor desea ver qué tal le ha ido a la clase en general o a un alumno en un ejercicio.* Este UO es quizá de los más importantes, ya que es el que nos muestra todo el feedback recibido de los alumnos. Podemos ver el estado de un ejercicio para cada alumno o ver globalmente (en porcentajes) el progreso de la clase.

UO5-T: *Ver la descripción completa de un ejercicio.* Un profesor quiere ver la descripción completa de un ejercicio (identificador, enunciado, página, tema, etc.).

UO6-T: *Editar un ejercicio.* El profesor desea editar los atributos de un ejercicio. Quizá se haya equivocado al crear el ejercicio en algún detalle o quiera añadir más detalles. Especialmente útil para los ejercicios que se han guardado y se quieren preparar mejor.

UO8-T: *Cerrar sesión.* El profesor quiere cerrar su sesión activa.

UO9-T: *Cambiar el idioma de la aplicación.* El profesor desea cambiar el idioma con el que lee la aplicación. Un requisito fundamental de cara a la internacionalización fue que la aplicación estuviera en 4 idiomas: castellano, euskera, inglés y francés.

UO10-T: *Cambiar de asignatura.* El profesor, que tiene más de una asignatura, quiere cambiar de una asignatura x a otra asignatura y.

6.2. Líneas futuras y Propuestas de mejora

exerClick fue pensada como parte de un proyecto mayor, *pClick* y a su vez *PresenceClick*, que integrará ésta y otras aplicaciones. Pasaría a tener un módulo en *PresenceClick*, de modo que algunas opciones de *exerClick* que quizás necesiten de un entorno de trabajo más apropiado (como un ordenador) podrían exportarse a ese módulo.

Además de ésto, se pensaron más ideas relacionadas a *exerClick* que quedaron en el tintero. La siguiente lista aporta todas esas ideas y otras que surgieron durante o después del desarrollo del proyecto:

- Evaluar a un alumno en un ejercicio (UO7-T). Este objetivo se marcó como una posibilidad y resultaba interesante. Para un ejercicio se permitía asignarle a un alumno una valoración del 1 al 5 (usando emoticonos de tristeza o sonrisa de diferentes grados) al igual que hace *PresenceClick*. El docente se acercaría a un alumno, vería el estado en el que se encuentra con el ejercicio y si al docente le pareciera correcto podría darle una valoración (no visible para el alumno, es algo personal del docente).

Otro objetivo diferente sería que el propio alumno pudiera valorarse a sí mismo, de modo que cuando un profesor da un ejercicio por finalizado el alumno puede introducir una valoración sobre sí mismo.

- Mejorar la autenticación queda abierta para varias mejoras:
 - Mejorar el control de errores como que no haya conexión (con un aviso), por ejemplo.
 - Que a un alumno se le reconozca si se ha identificado en la clase (utilizando el sistema de reconocimiento por tarjeta) y en caso contrario enviar un mensaje de error.
- Poder descargar un PDF con los detalles de los ejercicios, de modo que para ejercicios con muchos detalles se tuviera una visualización más sencilla.
- Borrar ejercicios por completo. Esta idea se pensó en realizarla desde *PresenceClick* por seguridad, por eso no se ha implementado.
- Permitirle al alumno acceder a su propio perfil con las posibilidades de cerrar sesión y cambiar de idioma (al igual que el profesor). Debido a la semejanza entre pantallas no llevaría mucho trabajo añadir esta mejora.

- Mejorar el uso de Responsive Web Design en pantallas grandes. Se ha usado RWD para adaptar el diseño a cualquier pantalla, enfocándose en una pantalla de un dispositivo móvil. En tablets, por ejemplo, el diseño deja muchos huecos libres sin aprovecharse. Sería interesante aprovechar esos espacios libres para añadir más información. También se podría plantear el diseño de otra manera para aprovechar mejor las pantallas grandes.

Toda la información reunida con exerClick que alimente la base de datos proporcionará datos muy interesantes para futuros usos. Si bien por si sola la información de exerClick no parece demasiado útil, una vez que juntamos esta y la información reunida por los otros módulos tendremos una base de datos con muchísima información para realizar todo tipo de procesos. Podríamos realizar incluso predicciones sobre el rendimiento de los alumnos hacia el final del curso, por ejemplo (lo que viene siendo *Machine Learning*). Se puede usar la información para ver como se podría mejorar una asignatura respecto a entregables, clases, etc. Hay bastantes posibilidades futuras, pero *exerClick* se dejó en este punto como un Trabajo de Fin de Grado, con posibilidades de mejora hacia el futuro.

6.3. Lecciones aprendidas

Apache Cordova vs Aplicación Nativa

- Basándome en mi experiencia durante el proyecto y en algunas opiniones recogidas me ha parecido que la tecnología de Apache Cordova intenta abarcar demasiado. Las aplicaciones que se generan se vuelven lentas y con fallos. Para un proyecto que quiera un diseño interesante, sin embargo, Cordova es una herramienta sencilla de usar. También para aquellos que no tengan conocimientos de las tecnologías de Android y quieran crear una aplicación esta herramienta es estupenda.

Para desarrollar una futura aplicación, en lo personal, tendría en cuenta crear una aplicación nativa de Android. El desarrollo de la interfaz ha sido la parte más costosa de esta aplicación, cosa que con una aplicación nativa es sencillo y robusto, adaptable a cualquier pantalla fácilmente (con Cordova hay que romperse más la cabeza). A parte de eso señalaría también los siguientes motivos:

- La falta de documentación de Cordova frente a la gran comunidad de Android.
- Gestores de errores o *plugins* fáciles de instalar y utilizar en Android con Cordova se vuelven una pesadilla.

En resumen, escoge Cordova si:

- No tienes conocimientos de Android.

- Quieres realizar una aplicación visualmente atractiva con tus conocimientos web.
- La aplicación es ligera y no va a tener demasiada funcionalidad.

Y crea una aplicación nativa en Android si:

- Va a tener muchas funcionalidades.
- Vas a manejar muchas animaciones o efectos.
- Quieres crear algo simple (sin meterte en todo el tema Cordova) o algo realmente amplio (resulta más cómodo para no tener tantos problemas).

Generar un apk con Apache Cordova

Para generar un apk necesitaremos usar un IDE de desarrollo, en este caso se usó Eclipse Luna. Crearemos un nuevo proyecto en Android con el código que tenemos (si no estábamos trabajando ya sobre un IDE) tal y como se muestra en la figura 32.

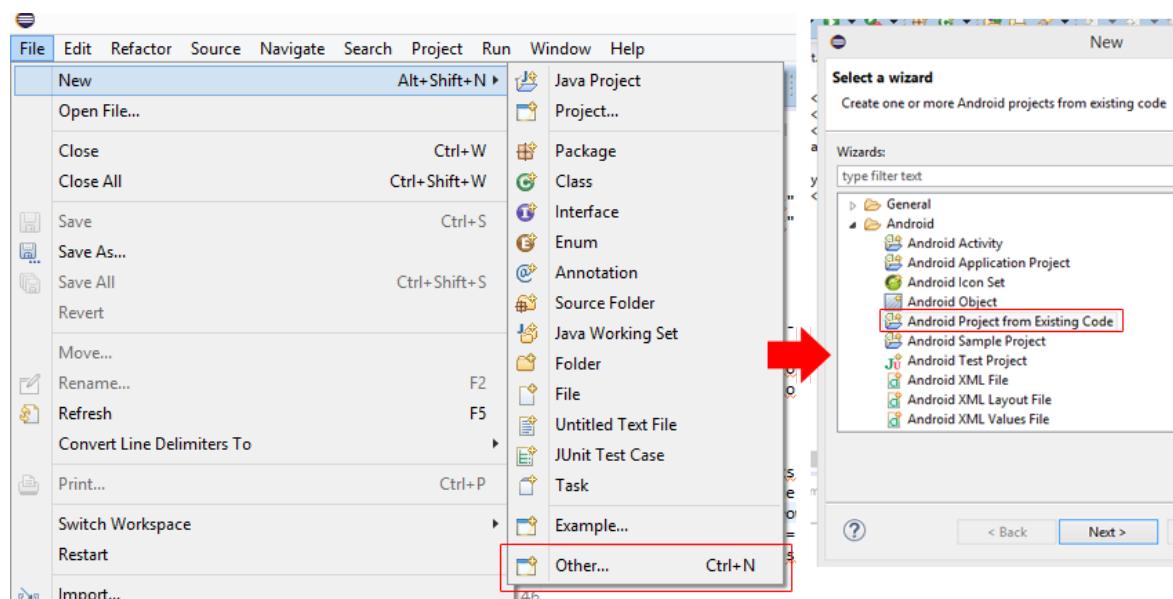


Figura 32: Primer paso: crear un proyecto nuevo

En el proyecto nuevo debemos meter nuestro código en la ruta "assets/www" (si no existe cread las carpetas necesarias). También necesitaremos el archivo cordova.js que podemos conseguir en <https://github.com/apache/cordova-js>. Deberíamos tener algo parecido a la estructura de la figura 33.

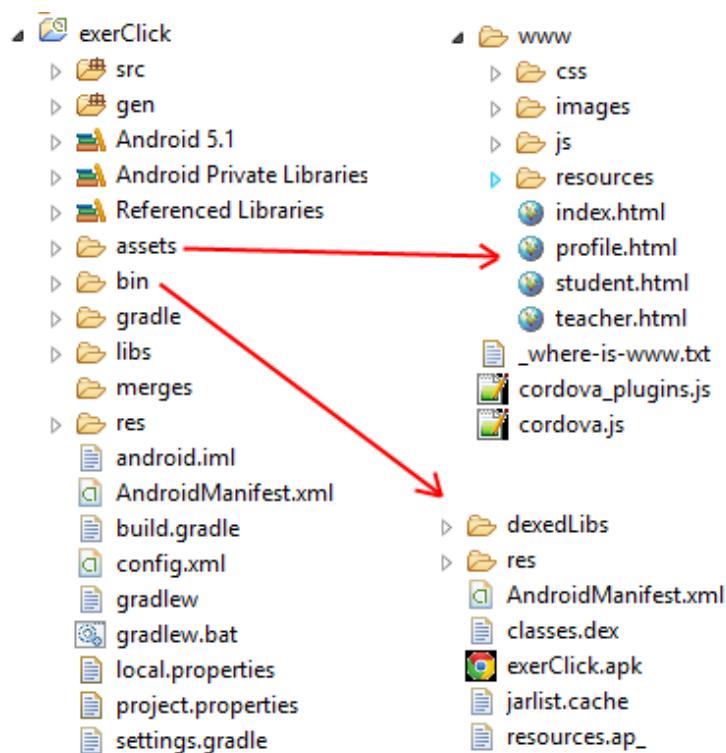


Figura 33: Segundo paso: tener la estructura y archivos correctos

Finalmente debemos ejecutar el proyecto (aunque no llegue a cargarse el emulador, para hacerlo rápido podemos anular el proceso cuando se nos pida escoger el emulador). Una vez ejecutado se habrá creado el apk y podremos encontrar en la carpeta bin de nuestro proyecto de Eclipse (figura 33).

Bibliografía y Referencias

- [1] Isabel Fdez. de Castro Begoña Losada Maite Urretavizcaya. “*N_InterMod*: Una Propuesta de notación de Diálogo enriquecida para el desarrollo ágil de aplicaciones interactivas”. En: *Ibergarceta publicaciones, S.L.* (2010). (visto 9-10-14).
- [2] *Bootstrap. The most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.* <http://getbootstrap.com/>. (visto 1-12-14).
- [3] *Font Awesome.* <http://fortawesome.github.io/Font-Awesome/>. (visto 1-12-14).
- [4] *Manifiesto Ágil.* <http://www.agilemanifesto.org/iso/es>. (visto 24-3-15).
- [5] Leire Ozaeta. *qClick, Sistema móvil de Pregunta-Respuesta en el aula.*
- [6] *The Apache Software Foundation.* <https://cordova.apache.org/>. (visto 1-12-14).

Apéndice A

Actas de Reunión

Reunión de Trabajo 1

Fecha: 8 de octubre de 2014

Hora de inicio: 12:30

Hora de finalización: 13:45

Presentes: Maite Urretavizcaya, Adrián Núñez

Temas tratados durante la reunión:

- Inicio del proyecto: presentación del proyecto y de la metodología de trabajo a seguir durante el desarrollo del mismo.
- Acordar tareas a realizar antes de la siguiente reunión.

Resumen de la reunión:

- Se presenta ExerClick: la aplicación web para gestión de ejercicios en el aula. Será una aplicación accesible desde dispositivos pequeños como el teléfono móvil de un alumno hasta dispositivos con pantallas más grandes como las de un ordenador que puede haber en el aula.
- La primera idea general de ExerClick es que sea una aplicación que puedan manejar tantos alumnos como profesores (dos roles definidos). En principio la idea es que los ejercicios se realicen dentro del aula. Los profesores podrán proponer ejercicios

en la aplicación para que los alumnos los realicen. Los alumnos durante o después de la realización del ejercicio podrán responder a la propuesta del profesor: si lo han terminado o han tenido dudas, si están atascados, etc. De esa forma el profesor puede realizar un seguimiento más cercano, rápido y sencillo del alumnado.

- La filosofía de la aplicación es que tenga mucha funcionalidad pero en pocos Clicks.
- Se buscan 4 factores fundamentales en la aplicación:
 - El uso de tecnologías actuales: HTML5, CSS3, PHP, MySQL y Symphony.
 - La simplicidad de la aplicación (en pocos clicks se deben de poder realizar muchas cosas).
 - Que sea internacional: internamente estará escrito en inglés (variables, comentarios, etc.) con el fin de que pueda llegar a ser código libre accesible a cualquiera. Además se quiere presentar en 4 idiomas (castellano, euskera, inglés y francés).
 - Uso de la tecnología Responsive Web Design (RWD). La aplicación se quiere adaptar a cualquier dispositivo.
- Presentación y explicación general sobre la metodología de desarrollo InterMod a utilizar. Intermod es una metodología ágil, basada en modelos y centrada en los usuarios.
- Se identifica el equipo de trabajo:
 - Adrián Núñez, el alumno.
 - Maite Urretavizcaya, la profesora.
 - Juan Miguel López y Begoña Losada, parte del grupo GaLan, que actuarán como validadores.
 - Usuarios finales, tanto profesores como alumnos, que actuarán como validadores.

Acordado para la siguiente reunión:

- Inicio de la fase previa al desarrollo de la aplicación: recopilación de información sobre cualquier tema de interés para el proyecto.
- Generar ideas, prototipos, etc. para tener una visión más concreta del tipo de aplicación que se quiere hacer.

Reunión de Trabajo 2

Fecha: 21 de octubre de 2014

Hora de inicio: 11:00

Hora de finalización: 12:25

Presentes: Maite Urretavizcaya, Samara Ruiz, Adrián Núñez

Temas tratados durante la reunión:

- Presentación de las ideas pensadas en la fase inicial del proyecto.
- Partiendo de la discusión de las ideas las decisiones más importantes han sido:
 - Crear una aplicación para móviles en lugar de una aplicación web.
 - Iniciar el proyecto teniendo en cuenta que la aplicación sólo va a usarse dentro de clase.

Resumen de la reunión:

- Se comienza explicando las ideas reunidas para la aplicación (en el documento de ideas acumuladas por parte del alumno durante las semanas anteriores), de forma que se tenga más claro el tipo de aplicación que se quiere realizar.
- Samara Ruiz se une a la reunión.
- Se comentan más ideas y unos bocetos iniciales. Comienza la discusión de la que salen las siguientes ideas:
 - En lugar de una aplicación web se realizará una aplicación para móviles.
 - Se limitará de momento el uso de la aplicación a cuando el docente y el alumnado estén en una sesión lectiva (excluyendo algunas funciones).
 - Se deja como pendiente un sistema de valoración de dificultad de los ejercicios por parte del alumno y de valoración del grado de satisfacción en la resolución de ejercicios de los alumnos por parte del profesor. Ambas usando el sistema que se usa en PresenceClick.
- Se proponen ideas para simplificar los bocetos iniciales. El fin es tener una versión más simple, con menos botones y pantallas (se pretende realizar todo en pocos clicks). Se proponen más ideas respecto a la interfaz y queda pendiente el realizar nuevos bocetos.

- Se da inicio a la iteración 1 del proyecto como una iteración larga con dos objetivos: responder a un ejercicio (por parte del alumno, UO-1S) y proponer un ejercicio (por parte del profesor, UO-1T). Se escoge como primer UO (Objetivo de Usuario) el de responder a ejercicios por parte del alumno (UO-1S). Los nuevos bocetos serán parte del prototipo en papel para el UO.
- Se comparte el documento de ideas general con Maite Urretavizcaya y Samara Ruiz.

Acordado para la siguiente reunión:

- Corregir el documento de ideas para el proyecto con lo decidido/propuesto en la reunión.
- Iniciar la iteración 1 con la fase de Análisis y Captura de requisitos. Enviar cuanto antes unos prototipos en papel resultado de esta fase para ser validados antes de la siguiente reunión.

Reunión de Trabajo 3

Fecha: 3 de noviembre de 2014

Hora de inicio: 8:40

Hora de finalización: 10:30

Presentes: Maite Urretavizcaya, Samara Ruiz, Adrián Núñez

Temas tratados durante la reunión:

- Discusión sobre los prototipos en papel desarrollados: definir las funcionalidades y diseño.

Resumen de la reunión:

- Se decide separar el UO-1T en dos UOs: una cuando haya una sesión activa y otra cuando no. Las denotaremos como UO-1T y UO-2T.
- Se decide que el nombre del profesor/alumno no es tan importante y que es mejor añadir el nombre de la asignatura. En lugar del nombre aparecerá un ícono que dará acceso al perfil del usuario donde aparecerán las siguientes opciones:
 - Ver el nombre del usuario.
 - Escoger la asignatura para el caso del profesor. Las asignaturas se ordenan por orden de proximidad horario (la siguiente clase será la primera de la lista).
 - Cerrar sesión.

Además, cuando un profesor inicia sesión, si tiene más de una asignatura y ninguna sesión actualmente, será redirigido al perfil para escoger alguna asignatura.

- Los ejercicios se identificarán por lo menos con un string identificativo (único para cada sesión, pero no único para una asignatura, pues identificadores como “1” pueden repetirse muchas veces a lo largo de una asignatura). Este string puede contener un sólo un número como al principio o cosas más detalladas como “1 - con ordenación”, “1 - con x”, “2.3”, etc.
- Se decide que una vez se inicia sesión, si no hay una clase en ese momento se irá abrirá la pantalla principal en la pestaña de ejercicios preparados. En caso de que haya clase se abrirá en la pestaña de ejercicios activos.
- Para los botones de lanzar se usará el ícono del avión de papel.

- Los botones de editar/borrar ejercicios deben estar separados para que no haya ningún problema por darle sin querer a borrar.
- En lugar de que haya un botón para ver el ejercicio completo se harán los ejercicios clickables, de modo que al pinchar sobre un ejercicio te muestre los detalles del ejercicio (nos ahorramos un botón).
- Se piensa en un “Modo Ordenar” para poder ordenar manualmente la lista de los ejercicios activos o preparados para que aparezcan con un orden concreto. El objetivo es mostrar que ejercicios se desea que se realicen primero.
- En la lista de ejercicios, cada ejercicio tendrá un pequeño indicador de cuánta gente lo ha dado por acabado y cuanta tiene dudas en el ejercicio como vista previa.
- Un color para identificar el estado de los ejercicios, como primer acercamiento: rojo (activo), amarillo (preparado) y azul (finalizado).
- Al crear un nuevo ejercicio si su identificador ya existe tenemos dos posibilidades, queda pendiente ver cuál es la mejor:
 - Añadir al final del identificador un número como en el caso: 1 que pasaría a ser 1(2), por ejemplo.
 - Mostrar un mensaje de error.
- Cuando acaba una sesión los ejercicios activos pasan al estado preparados, quedando guardados todos los avances realizados durante la sesión.
- Se ha pensado en dejar la parte de eliminar completamente ejercicios a Presence-Click, entrando como administrador.

Acordado para la siguiente reunión:

- Desarrollar unos prototipos en papel con lo acordado durante la reunión, a limpio.

Reunión de Trabajo 4

Fecha: 11 de noviembre de 2014

Hora de inicio: 11:00

Hora de finalización: 11:40

Presentes: Maite Urretavizcaya, Samara Ruiz, Adrián Núñez

Temas tratados durante la reunión:

- Planteamiento de la funcionalidad del los UO-1, UO-2, UO-3 y UO-4 mediante los prototipos en papel y los autómatas de estados (fin de M-1).
- Inicio de M-2 (implementación de la interfaz).

Resumen de la reunión:

- Se muestran los prototipos en papel junto a sus correspondientes autómatas de estados (dos: uno del profesor y otro del alumno).
- Se acuerda que cada UO tendrá el siguiente contenido:
 - Nombre del UO.
 - Descripción breve.
 - Autómata de estados individual del UO.
 - Pantallas necesarias para el UO.
- Se visualiza, por cada UO (del 1 al 4), qué pantallas debería usar y cómo debería ser la interacción con el usuario para cumplir el UO.
- Se plantean las siguientes ideas/propuestas:
 - Vista alumno:
 - Cuando el alumno responde a un ejercicio, el botón que ha marcado (ya sea que ha acabado o que ha tenido una duda) se resaltará. También es posible que se le cambien los bordes al contenedor en el que está el ejercicio.
 - Siguiendo con los mismos botones: añadir un lapso de tiempo antes de que la respuesta del alumno cuente dentro de las estadísticas por si se hubiera equivocado pulsando el botón.

- Añadir dos barras de progreso de la sesión: una del alumno y otra de la clase (para poder ver su progreso en comparación con el resto de compañeros).
- Vista profesor:
 - Los iconos para ejercicio acabado o alumno con dudas deben ser iguales para ambas vista: profesor y alumno.
 - En vez de usar estadísticas del tipo “20/100 han acabado este ejercicio” interesa usar porcentajes (20%), ya que para otros valores puede que no quede tan claro el porcentaje real.
 - En la parte de dudas, en estadísticas, se mostrarán sólo los alumnos que tienen en el momento dudas, no los que tuvieron anteriormente.
 - Cuando el ejercicio está finalizado se propone añadir diferentes estadísticas (nuevo UO que queda como pendiente): gente que ha tenido duda en algún momento, dudas resueltas, gente que se quedó con dudas, gente que terminó el ejercicio y tiempo de realización del mismo.
- Para esta iteración se acuerda que, para simplificar las funcionalidades que vamos a añadir (no queremos añadir todas las funcionalidades de golpe), omitiremos detalles del diseño original que más tarde serán añadidos:
 - Pestaña de ejercicio preparados.
 - Opciones avanzadas de creación de un ejercicio.
 - Guardar un ejercicio (para que se envíe a preparados). Sólo se podrá lanzar. Si bien es posible que existan más elementos a omitir, estos no fueron identificados en el momento de la reunión.
- Se acuerda finalmente iniciar con la implementación de la interfaz mediante las tecnologías HTML y CSS. Además, se usará PhoneGap para portar estas tecnologías a las plataformas móviles (en principio Android, iOS y Windows Phone).

Acordado para la siguiente reunión:

- Implementación de la interfaz en HTML y CSS y adaptación a móviles de esta mediante PhoneGap.

Reunión de Trabajo 5

Fecha: 13 de enero de 2015

Hora de inicio: 11:50

Hora de finalización: 12:30

Presentes: Maite Urretavizcaya, Samara Ruiz, Juan Miguel López, Adrián Núñez

Temas tratados durante la reunión:

- Demostración y discusión del M-2 del UO1-T.

Resumen de la reunión:

- Se empieza mostrando el M-2 del UO1-T desarrollado en un Samsung Galaxy S5 al grupo. La aplicación ha sido añadida al dispositivo mediante USB, no se ha conseguido una apk funcional para instalar la aplicación en cualquier dispositivo.
- De la próxima reunión en adelante se enviarán capturas de pantalla del modelo para agilizar la reunión y que todos los participantes tengan claro el escenario.
- Aspectos de diseño a mejorar:
 - Los botones rápidos que aparecen en las cajas de los ejercicios deben ir siempre a la derecha, alineados con el identificador del ejercicio (aunque haya que acortarlo mucho, se supone que debe ser una clave corta). De esta forma la caja ocupa menos. Además, al pulsar el ejercicio para verlo completo (más adelante) se podrá ver el identificador entero.
 - Alguna forma más visual de identificar un ejercicio en el que haya especialmente un número significativo de dudas (color de fondo, algo llamativo, texto de otro color, etc.).
- Se acuerda realizar finalmente el M-3 del UO1-T. Además, se acuerda seguir con el UO1-S (los 3 modelos) para poder realizar así una evaluación más funcional (con el profesor pudiendo lanzar un ejercicio y el alumno respondiendo a este). También se plantea realizar el UO4-T (ver estadísticas de un ejercicio) para poder tener una aplicación verdaderamente funcional.

Acordado para la siguiente reunión:

- M-3 del UO1-T.

- M-1, M-2 y M-3 del UO1-S y UO4-T.
- Obtener una apk funcional de la aplicación.

Reunión de Trabajo 6

Fecha: 27 de enero de 2015

Hora de inicio: 9:30

Hora de finalización: 10:10

Presentes: Maite Urretavizcaya, Adrián Núñez

Temas tratados durante la reunión:

- Análisis de la validación de los M-2 de UO1-T, UO4-T y UO1-S.

Resumen de la reunión:

- Intercambiar los iconos de ejercicios activos (señal de alerta) y de ejercicios preparados (avión de papel).
- Modificar los colores de los botones: rojo siempre activo, azul finalizado y amarillo preparado. Por ejemplo, el de lanzar un nuevo ejercicio (que salía con el avión de papel en amarillo) debería cambiar a rojo.
- Colocar 3 botones por cada ejercicio para poder mover los ejercicios entre los estados activo, finalizado y preparado cuando se deseé.
- Hacer la barra superior más grande y la parte del nombre de la asignatura y el fondo tipo pizarra más pequeño.
- Usar las fuentes EHU Sans para el idioma castellano y EHU Serif para euskera.
- A la vista de alumno añadirle la barra extra de progreso de la clase. También añadirle a los ejercicios colores de fondo dependiendo de sus estados.

Acordado para la siguiente reunión:

- Mejorar lo comentado y continuar con el proyecto.
- Actualizar la lista de UOs con información actualizada.

Reunión de Trabajo 7

Fecha: 6 de febrero de 2015

Hora de inicio: 11:30

Hora de finalización: 12:20

Presentes: Maite Urretavizcaya, Samara Ruiz, Adrián Núñez

Temas tratados durante la reunión:

- Demostración de los M-3 de los UO1-T, UO3-T, UO4-T y UO1-S.

Resumen de la reunión:

- Se les enseña a los asistentes los M-3 de los UO1-T, UO3-T, UO4-T y UO1-S. La funcionalidad básica (o ciclo principal de la aplicación) de enviar ejercicios y que los alumnos respondan y se refleje esta respuesta en la vista de profesor no está completa.
- Se deciden realizar las siguientes modificaciones:
 - Modificar las barras de progreso: colocar un texto más corto, luego la barra y finalmente el porcentaje, todo en una misma línea para que no ocupen tanta altura (esto se aplica para las barras de progreso de la vista de profesor y para las de la vista de estudiante).
 - El tamaño de la barra superior y de la zona de título de la asignatura hacerlos más grandes para que se vean mejor.
- La internacionalización de la aplicación (tenerla disponible en 4 idiomas) queda fijada como objetivo a corto plazo para ir pensando y planificando para añadir. Se dan algunas ideas basadas en qClick y PresenceClick del modo de traducir el texto.
- Se comentan algunas modificaciones para hacer a la memoria (se dejan por escritas en un documento previamente preparado por la directora del proyecto que le es entregado al alumno).

Acordado para la siguiente reunión:

- Seguir intentando generar el archivo .apk.
- Buscar la internacionalización de la aplicación.

Apéndice A

- Conseguir que el ciclo principal de la aplicación funcione.
- Corregir la memoria y el diseño de la aplicación con lo comentado durante la reunión.

Apéndice B

Actas de Pruebas

Pruebas con el Equipo 2

Fecha: 23 de febrero de 2015

Presentes: Mikel Balduciel, Xabier Zabala, Alex Magadán, Adrián Núñez

Conclusiones:

- El icono de bandera que simboliza finalizar un ejercicio es poco intuitivo. La idea aportada sería usar una bandera de tipo meta, con diseño de cuadrados blancos y negros alternados.
- El icono de alerta que simboliza un ejercicio preparado es también poco intuitivo, no se aportan ideas al respecto.
- Se da la idea de cambiar de asignatura pulsando sobre el nombre de la asignatura que se está viendo en la ventana madre (donde se muestran los ejercicios). Se hace poco intuitivo ir al perfil para hacer este cambio según se ha comentado.
- Se comenta que se debería de hacer más intuitivo el “Ir al perfil”.

Apéndice C

Bases de datos

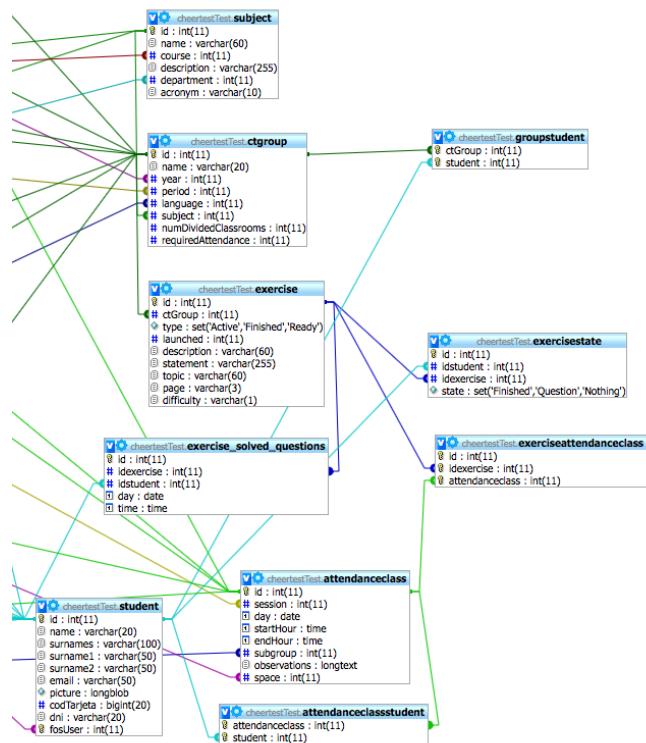


Figura 34: Parte de la base de datos de exerClick

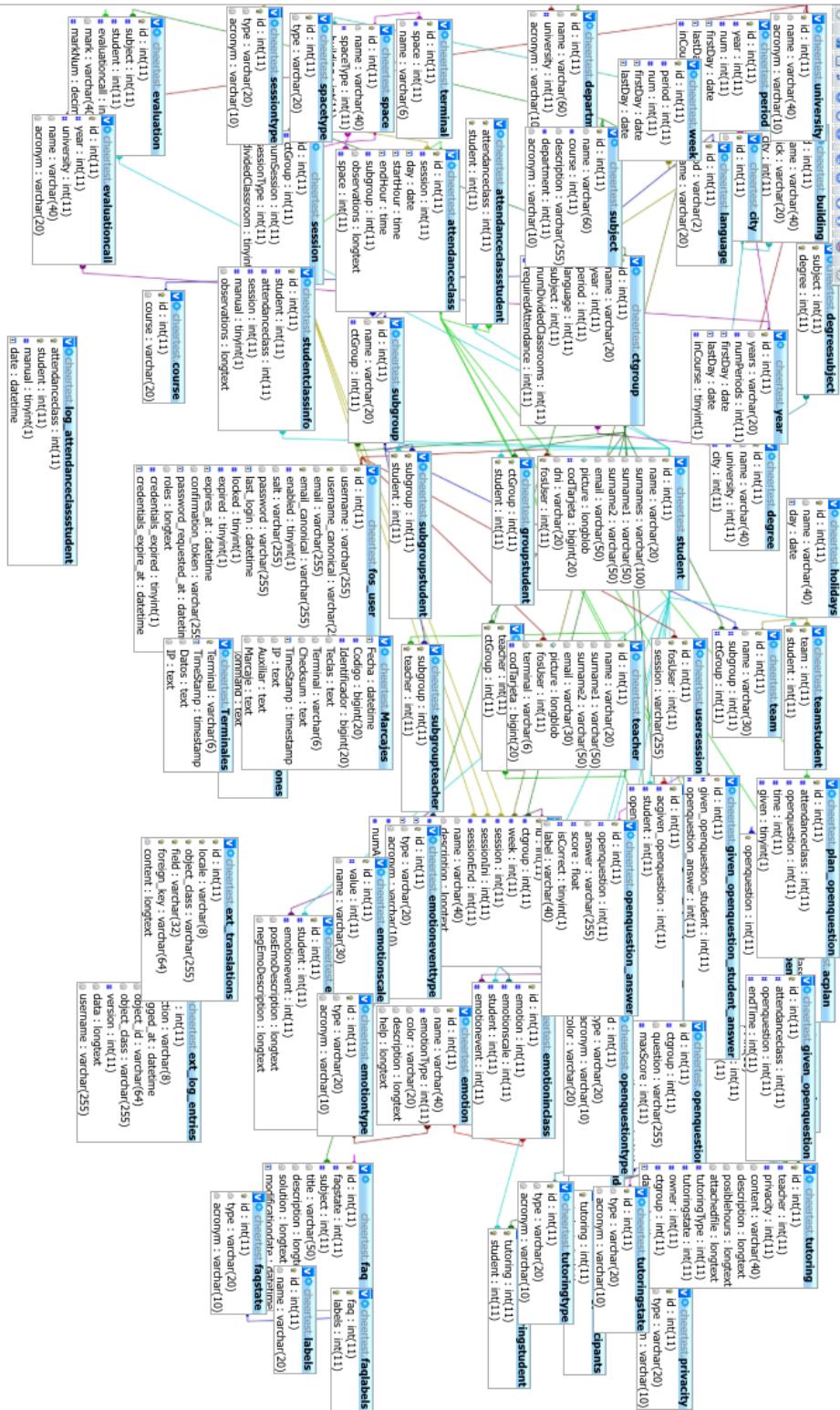


Figura 35: Base de datos de PresenceClick