



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Aplicación para clase interactiva para tablets ANDROID

Trabajo Fin de Máster

Máster Universitario en Gestión de la Información

Autor: Francisco José Guerra Aroca

Tutor: Miguel Sánchez López

2013 - 2014

Resumen

El presente proyecto tiene por objetivo la creación de una aplicación web para la docencia utilizando módulos y herramientas que permiten la creación de documentos colaborativos mediante el uso de JavaScript. El proyecto a realizar se centra en potenciar la interactividad existente entre un profesor y sus alumnos durante la realización de una clase docente.

Dicha aplicación podrá ser lanzada en el navegador de cualquier dispositivo sea portátil o fijo. Por ello su desarrollo es en lenguajes estándar de programación web como HTML5, CSS3 y JavaScript para que cualquier dispositivo sea capaz de interpretarlo.

Finalmente se elaborará un plan de desarrollo de funcionalidades para dotar de nuevas y mejoradas características a la aplicación en posteriores actualizaciones o versiones de la misma.

Palabras clave: web, interactividad, docencia, aplicación.

Abstract

This project aims at creating a web application using teaching modules and tools that enable the creation of collaborative documents using JavaScript. The project to be realized focuses on enhancing the existing interaction between a teacher and his students during a class.

This application may be launched in the browser of any device, portable or fixed. For that reason its development is in standard web programming languages like HTML5, CSS3 and JavaScript for any device to be able to interpret it.

Finally, a development plan to provide new functionalities and improved features to the app in future updates or versions of it will be developed.

Keywords: web, interactivity, teaching, application.

Índice

1.	Introducción.....	7
2.	Marco teórico.....	11
2.1.	Evolución de la enseñanza	11
2.2.	Nuevas tecnologías en la docencia	13
2.2.1.	Video.....	14
2.2.2.	Informática	15
2.2.3.	Telecomunicaciones.....	16
2.3.	Classroom Presenter 3	17
2.3.1.	Inicio aplicación.....	17
2.3.2.	Rol instructor	18
2.3.3.	Rol estudiante	23
2.4.	Soluciones actuales para la docencia	24
2.4.1.	Modelo Moodle - Sakai.....	25
2.4.2.	Modelo BlackBoard / WebCT	26
2.5.	E-learning.....	27
2.6.	Conclusiones	29
3.	Especificación.....	31
3.1.	Características aplicación.....	32
3.2.	Diseño	35
3.2.1.	Rol Docente.....	36
3.2.2.	Rol Estudiante	38
4.	Implementación.....	41
4.1.	Primeros pasos: Activando la aplicación	41
4.2.	Lenguajes de programación	47
4.3.	Autorización de la aplicación	48
4.4.	Control de usuarios	50
4.5.	Invitar alumno	55
4.6.	Transparencias.....	56
4.7.	Pizarra virtual	58
5.	Conclusiones.....	69
6.	Referencias	71



1. Introducción

En el presente proyecto, se busca generar una herramienta para la enseñanza, mediante la creación y desarrollo de una aplicación web. Esta aplicación tiene como funcionalidad principal el permitir la interactividad entre el profesor y los alumnos asistentes a su clase presencial. Para ello se dotará la aplicación de funcionalidades que permitan al profesor realizar preguntas que los alumnos podrán responder a través de la propia aplicación, elaborar cuestionarios, o que los propios alumnos envíen sus preguntas al profesor durante el desarrollo de la clase.

Para su desarrollo haremos uso de la API de Google conocida como *Google Drive Realtime*¹. Esta es la que utiliza la propia compañía para las ediciones interactivas de archivos en su plataforma de Google Drive, mediante la cual, varias personas pueden editar un archivo al mismo tiempo, y a su vez en tiempo real ver las modificaciones que se van haciendo del mismo.

Lo que se pretende es generar una aplicación donde el profesor tenga todas las posibilidades de edición sobre aquello en lo que esté trabajando, así como la posibilidad de generar nuevos contenidos que enviar a sus alumnos o que estos puedan responder preguntas y ser evaluados en tiempo real.

Por otra parte el alumno tendrá la oportunidad de ver en tiempo real las modificaciones y explicaciones de su profesor en el material docente así como poder responder a preguntas, cuestiones o incluso interactuar con el mismo. Se pretende también que se puedan intercambiar roles, y que pueda llegar el momento en que el alumno tome la palabra y pueda realizar el “rol de profesor” por un determinado periodo.

El funcionamiento de la aplicación será similar al que se utiliza en la aplicación *Google Drive*, en cuanto a lo que se refiere a interactividad y visualizado de la información, pero añadiendo parte de las funcionalidades que permite la aplicación *Classroom Presenter*², que es aquella que se toma como base para el desarrollo de esta nueva aplicación y que ha sido utilizada por docentes de esta propia universidad durante diversos cursos en ordenadores Tablet pc.

Esta aplicación desarrollada por la universidad de Washington, cuya última versión es de 2008, es ideal como modelo de desarrollo del proyecto. Tomando como base su funcionamiento se quiere extrapolar a la época actual. Una época en la cual se está produciendo un gran auge de métodos de docencia online, así como la realización de cursos gratuitos / de pago, de carácter online. Sin ir más lejos, la propia universidad en la que nos encontramos, la Universitat Politècnica de València, tiene presencia online y realiza cursos online como por ejemplo los que podemos encontrar en *miríada*³. Una página que ofrece la posibilidad de realizar cursos de prestigiosas universidades de manera gratuita, proporcionando un certificado de realización del curso, emitido por la propia entidad, una vez superado el mismo.

¹ <https://developers.google.com/drive/realtime/realtime-quickstart>

² <http://classroompresenter.cs.washington.edu>

³ <https://www.miriadax.net/>

Este proyecto busca desarrollar un método de docencia, a través de una aplicación web, que utilizando las bases establecidas en las herramientas que se utilizaban con anterioridad, saque partido de las nuevas posibilidades y métodos de docencia online existentes hoy en día, generándose una unión que proporcione una herramienta docente, pero sin perder la asistencia presencial al aula por parte de los alumnos. De esta manera se podrán realizar métodos de trabajo y formativos distintos, apoyados en las nuevas tecnologías. Donde cada alumno pueda formar parte de la clase con su ordenador personal, Tablet o Smartphone, sin necesidad de instalación de programas. Esto supone una gran ventaja ya que se genera una aplicación tal que puede ser lanzada en cualquier ordenador, sin importancia el tipo de (PC o Mac), o del sistema operativo del dispositivo móvil (Android, iOS, Windows phone, etc.). Todo ello gracias al tratamiento como aplicación web y su ejecución directamente a través del navegador.

Debido al hecho de ser una aplicación web, se deberán desarrollar una serie de restricciones y limitaciones para controlar su acceso y funcionamiento. Esto es, que únicamente aquellas personas relacionadas con la propia docencia se va impartir o con los propios contenidos, puedan acceder a ella, y que su modificación o edición no sea libre para cualquier persona tenga acceso al sistema.

En primer lugar para solucionar este aspecto de acceso a la aplicación, se generaría una base de datos con la información de las personas que van a estar presentes en el curso o clase docente. De esta manera delimitaremos el número de personas con acceso directo al contenido que vayamos a generar en dicha página, así como también se indicará el rol principal de cada uno de ellos, diferenciando entre “profesor” y “alumno”. Para ello una vez tuviéramos esta información almacenada se generarían accesos para cada una de las personas, mediante la utilización de un usuario y contraseña de validación, o mediante invitaciones de acceso a la página gracias a las propias funcionalidades de la API de Google Drive.

De esta forma, tendremos también la información de los alumnos pertinentes del curso, la cual nos será de utilidad ya que podremos directamente vincularla a las respuestas, preguntas, o interacciones que vayan realizando durante la actividad docente. Por tanto si un alumno responde un cuestionario propuesto por el profesor a través de la aplicación, la información con las respuestas quedará también almacenada para una posterior visualización y análisis por parte del responsable docente si fuera necesario. Con todo ello, podríamos generar una base de datos que podría llegar a alcanzar la información suficiente para que se pueda autoevaluar a los alumnos participantes de la clase, o como un gran apoyo de cara al profesor a la hora de asignación de notas y resolución de los actos evaluativos.

Otro factor importante es la forma en la que mostrar la información y como el usuario “estudiante” la recibirá. Se parte con la idea de generar dos entradas de información, diferenciadas en dos paneles.

Por una parte, se recibirán en tiempo real los contenidos y ediciones que va realizando el profesor sobre el material docente con el que esté trabajando. Este panel lo consideraríamos de visualización de la información. Y por otra tendríamos el panel de interacción directa, donde el alumno recibiría las preguntas recibidas del profesor realizadas durante la clase para poder responderlas directamente, o incluso podría generar él mismo preguntas que llegarían a la pantalla del profesor, y que ningún otro alumno podría ver ya que se enviarían directamente al usuario “profesor”.

En definitiva, esta aplicación no busca en ningún caso convertirse en una herramienta para la impartición directa de curso o docencia online, aunque pudiera llegar a convertirse mediante las funcionalidades y herramientas de apoyo o servidores necesarios para tal fin. Lo que se busca es una herramienta de ayuda al profesor en su docencia, dándole nuevas posibilidades y nuevos métodos de trabajo, pero sin olvidar el trabajo presencial y directo con el alumno que toda clase debe tener. Se busca tener un contacto más directo con el alumno en lo que a explicación de contenidos y resolución de los mismos se refieren. Para que este, esté más implicado e involucrado en la clase, de una manera directa y además interactiva.

2. Marco teórico

Antes de comenzar con la metodología del proyecto realizado, su diseño, implementación, funcionalidades y todo aquello que lo describe, conviene ponerse en situación de como se ha llegado a la etapa actual en la que es posible realizar dicho proyecto.

Dado que en esta aplicación se dan unión por una parte las aplicaciones de docencia por ordenador tradicionales, y por otra, las nuevas tecnologías y herramientas web que han permitido el desarrollo de la aplicación. Es importante realizar una presentación de estas últimas para poner en situación el estado actual de las mismas.

Presentaremos la aplicación que sirve como base y punto de partida en el proyecto que se desarrolla, *Classroom Presenter 3*, una aplicación pionera en lo que a docencia online se refiere.

Así mismo, hay que tener en cuenta lo que hoy en día existe en el mercado, cuales son las soluciones existentes en el panorama actual, y el lugar en el que se situaría nuestra aplicación dentro de dicho mercado.

Finalmente es importante remarcar el auge de la “docencia online”, mediante cursos y universidades online, que se conoce como *E-learning*. Ver los motivos y las razones por las cuales se está convirtiendo en una herramienta cada vez más utilizada por estudiantes para su aprendizaje, y porque cada vez más personas confían en ella, analizando sus ventajas, pero también sus inconvenientes.

2.1. Evolución de la enseñanza

Allá por el año 2000 AC, en la región de Mesopotamia (lo que hoy se conoce como Irak), es donde se tiene constancia de lo que conocemos como educación⁴. De hecho ya alrededor del 3100 AC el pueblo sumerio (parte de los habitantes de Mesopotamia), desarrollaron las primeras técnica de escritura y poco después la escritura cuneiforme, utilizada para representar alrededor de 600 sílabas y números.

En esta época, los niños asistían a las escuelas, normalmente cercanas a los templos, debido principalmente al poder que tenían en aquel entonces los sacerdotes en todo el panorama educativo, y estaba diseñada precisamente para conducirlos en su vocación de futuros sacerdotes o escribas. Por aquel entonces se crean también las bibliotecas.

⁴ http://www.ehowenespanol.com/primeros-rastrros-educacion-mesopotamia-info_288999/





Figura 1. Piedra con símbolos de escritura de la época Mesopotámica

Sería más adelante el pueblo egipcio el que introdujo nuevas conductas en la educación. Por una parte se permitió a las niñas el acceso a la educación, aunque solo iban a la escuela si las tareas propias de su género lo permitían. Por otra, los niños recibían entrenamiento en la profesión de su padre. Toda aquella persona que pudiera costearse los estudios, podían ingresar en la escuela, pues para ellos la educación era algo muy respetable⁵. Otro avance muy importante es que, además del ámbito religioso y profesional, las escuelas comenzaron a enseñar materias como ciencias, matemáticas, y arquitectura.

El pueblo griego por su parte, únicamente volvió a permitir la educación a los varones., excepto en Esparta, aunque si bien esta, se prestaba poca atención a actividades culturales o saber leer y escribir, su principal preocupación se centraba en las actividades físicas⁶. Era a partir de los siete años cuando los niños comenzaban a ir a clase, donde se incluían actividades tales como la gimnasia, la lectura, escritura y aprenderse versos y fragmentos de poetas. También se convirtió en algo muy importante la música. Es en esta cultura donde nace el concepto de “educación” o como allí era conocido *Paideia*. Para ellos entendido como el proceso de crianza de los niños mediante la transmisión de valores (ser) y saberes técnicos (hacer) inherentes en la sociedad.

El pueblo romano llevaba a cabo su educación al margen. Era una educación de carácter más familiar, con orientación laboral de tipo campesino y consagración al servicio del estado. Aquí los educadores son los padres, especialmente el padre. La jornada escolar empezaba al alba. El contenido de la enseñanza era: lectura, escritura y algo de cálculo. En cuanto a la escuela como edificio, se situaba junto a los pórticos del foro o plaza pública. La escuela primaria era llamada *ludus litterarius*, y el maestro, *ludu-magister*⁷.

⁵ <http://www.iccc.es/2008/01/la-educacion-en-el-antiguo-egipto>

⁶ http://es.wikipedia.org/wiki/Historia_de_la_funcion_del_docente

⁷ <http://www.profesorenlinea.cl/universalhistoria/Roma/RomaEducacion.htm>



Figura 2. Maestro y alumno en la época romana

Ya en la edad media, sería la iglesia la que tomaría la responsabilidad de la enseñanza. En cuanto al desarrollo en educación superior, fueron sin embargo musulmanes y judíos los que desempeñaron el papel principal, interviniendo entre los conceptos de enseñanza de la antigua Grecia, y los estudiosos europeos.

Durante el siglo XVII se produce un cambio donde la ciencia sustituye la rama religiosa como motor de la docencia y nacerán diversos tipos de escuelas, algunos que ya llegan a nuestros días como la escuela municipal (pública), particular (privada), u otras como las pías o lasalianas.

En la edad moderna, se crean en España las Escuelas normales, siendo la primera instaurada en 1838 para varones, y 1858 para mujeres. Estas escuelas, son centros educativos dirigidos exclusivamente a la formación de maestros. Aquí nacería lo que se conoce como la figura de los profesores en la actualidad.

Pero que habría sido de estos profesores, y sus enseñanzas de no haber sido por las herramientas que a lo largo del tiempo se han ido desarrollando para favorecer sus métodos y estimular y ayudar a los alumnos en sus estudios.

2.2. Nuevas tecnologías en la docencia

Al inicio de la educación, dada por profesores tal y como los conocemos en la actualidad, es decir desde el siglo XIX, todos los materiales y herramientas eran materiales curriculares en papel o impresos, como libros de texto, cuadernillos de ejercicios, exámenes, etc.

Con las nuevas tecnologías, se generan nuevas herramientas de información que podríamos dividir en tres grandes grupos: video, informática y telecomunicación⁸. Estos tres grupos forman lo que se conoce como Tecnologías de la Información y la Comunicación (TIC). Aunque dentro de este grupo de tecnologías, no es que todas sean precisamente novedosas, pues por ejemplo el

⁸ http://campus.usal.es/~teoriaeducacion/rev_numero_06/n6_art_bustillo.htm

teléfono data del 1876, pero dada su nueva integración con otras técnicas y herramientas, se sitúa a en plena vanguardia, con sistemas de videoconferencias y charlas por voz como Skype⁹.

Estas nuevas tecnologías nos permiten:

- Se permite una educación sincrónica.
- Sirven para la captura y adquisición de información, así como de su posterior procesamiento y análisis.
- Favorecen el trabajo cooperativo o en grupo.
- Uso de los recursos que nos ofrecen los nuevos ordenadores y herramientas hardware, y software de la informática.
- Permite interconectar un espacio cerrado como la clase de un instituto, colegio, o universidad con el mundo exterior sin necesidad de desplazamientos.
- Nuevos sistemas de educación y métodos formativos como las clases a distancia o los exámenes online.
- Globalización y aumento del alcance de los contenidos docentes generados.

2.2.1. Video

Un vídeo educativo es un medio didáctico que facilita el descubrimiento de conocimientos y la asimilación de éstos. Además, puede ser motivador para el alumnado pues la imagen en movimiento y el sonido pueden captar la atención de ellos. En origen estos videos Beta, VHS o 2000 con sus correspondientes cintas de duración fija, pero eso era antes, ahora con las nuevas tecnologías todo es más sencillo y rápido.

Se pueden crear videos educativos de manera muy sencilla, pero para ello hay que realizar una investigación previa, y un esquema sobre lo que pretendemos comunicar y cómo queremos hacerlo. Hay que pensar si utilizaremos el sonido, si quitaremos el sonido ambiente, usaremos música de fondo, o los aspectos técnicos como la cámara que se utilizará, la resolución del video e incluso que software se utilizará para editar el video o el lugar donde se subirá el video.

Y el video no es un sistema en desuso precisamente, grandes compañías como Google, lo utilizan hoy en día para realizar la explicación de contenidos nuevos, como por ejemplo donde se explica el sistema *realtime*, con lo que se trabajará para desarrollar la aplicación. Lo que llamaríamos un video tutorial.

⁹ <http://www.skype.com/es/>

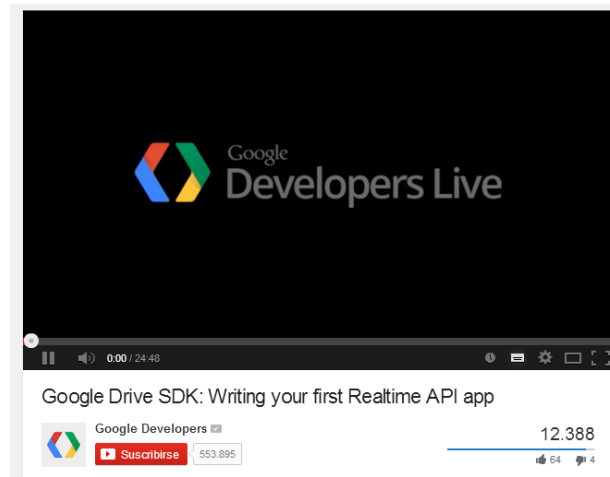


Figura 3. Video explicativo Google drive SDK¹⁰

2.2.2. Informática

Desde hace poco menos de 20 años, la informática está jugando un papel cada vez mayor en la educación, y en cualquier ámbito en general del mundo. Además su influencia viene en varios aspectos de la educación:

1. Es un tema propio de enseñanza a todos los niveles educativos. No solo es su utilización a nivel docente, si no que en si es una materia de conocimiento obligatorio a cada vez niveles más tempranos de educación.

Hoy en día, incluso desde los 3 y los 6 años de edad, algunos alumnos de educación infantil, comienzan a obtener los primeros conocimientos generales a través de los ordenadores, así como nociones básicas de su funcionamiento que les permiten desplazarse por unos contenidos adaptados a sus cualidades, y a sus edades tempranas.

No es hasta edades de educación superior donde la propia informática se da como materia docente y se comienza un estudio más profundo de sus características por parte del alumno, más que una mera herramienta de aprendizaje y formación ya presente en todas las escuelas del primer mundo, y cada vez más presente en las escuelas de otros países menos desarrollados donde este tipo de tecnologías abren unas puertas inmensas a nuevos medios y conocimientos educativos para los alumnos.

2. Proporciona nuevas herramientas y recursos para resolución de problemas, cuestiones, proyectos, y elementos evaluativos:
 - 2.1. Programas de enseñanza lineal o programada. Aquellos que guían al alumno a través de la psicología conductista y donde se mueve al alumno siempre por un camino para lograr los objetivos planteados. Este tipo de programas serían por ejemplo los que se utilizan con niños en edades tempranas para que aprendan a multiplicar, los colores, el abecedario, etc.

¹⁰ <https://www.youtube.com/watch?v=Mlaov3J4hDA>

- 2.2. Simuladores. Donde se generan unas situaciones y condiciones concretas para facilitar la utilización o aprendizaje de ciertas materias.
- 2.3. Programas de inteligencia artificial. Donde se interactúa directamente con el ordenador, viendo como este responde a nuestras órdenes, o el código que le facilitamos.

Este tipo de programas, y algunos otros, no suelen ser de una única categoría, sino que un mismo programa puede estar compuesto por diversas de estas.

3. Es un medio administrativo para el propio docente, que le permite controlar de una manera más automatizada y con un mejor soporte aquella información sobre sus alumnos, temarios, apuntes de clase, exámenes, guías docentes, etc.

Plataformas virtuales para el almacenamiento de información como Dropbox¹¹, de gestión del aprendizaje como Sakai¹² son muy utilizadas hoy en día por gran parte del personal docente en su día a día educativo como herramientas administrativas o de gestión. Estos son solo algunos ejemplos ya que ahora mismo las posibilidades que ofrece la informática son muy amplias.

2.2.3. Telecomunicaciones

Pese a que la informática es el medio que por cantidad y utilización sea ahora mismo el más presente dentro del grupo de las 3 tecnologías, es probablemente las telecomunicaciones el que mayor evolución está teniendo hoy en día, ofreciendo soluciones de interconectividad entre usuarios (el patrón de las TIC ahora mismo, la interconectividad entre usuarios) impensables hasta hace poco, servicios de videoconferencias, de streaming o de interconectividad e intercambio directo de información, entre otros. Todos ellos tienen eso sí, características que los engloban:

- Utilizan algún medio o tecnología de la comunicación
- Une individuos o grupos situados en cualquier parte del mundo.
- Metodología directa, en muchos casos con participación activa de todos los integrantes de la misma.

La principal razón por la cual no está tan integrada en el sistema educativo (mayor cuanto más superiores son los estudios), es sin duda su coste, que con la proliferación de redes de comunicación, RDSI, satélites, o programas de software (aunque en su mayoría de pago pero cada vez a precios más razonables), harán que este coste vaya bajando poco a poco. Si bien es cierto que este tipo de tecnologías se pueda pensar que son gratis, y que cualquier persona las

¹¹ <https://www.dropbox.com/>

¹² <http://sakaiproject.org/>

utiliza hoy en día, el tratarse de un uso docente donde están incluidos pueda ser grupos de docenas de personas o cantidades superiores a 10 o 15 personas, hace que las empresas propietarias de estos servicios hagan pagar un precio por el uso de las mismas para este tipo de grupos, sobre todo sino para su utilización, si no para su velocidad y estabilidad principalmente.

Con todos estos avances en las tecnologías, surge a principios del siglo XXI, una de las aplicaciones más novedosas y con mayores funcionalidades para los docentes de aquella época. Pionera en ofrecer interconectividad entre profesores y alumnos, fue una de las aplicaciones que marcó el inicio de una rama de la docencia que se viene siguiendo desde aquel entonces, donde profesor y alumno están cada vez más conectado a través de la TIC, y dándose incluso caso, donde ya la propia presencia del alumno en clase no es necesaria, debido a herramientas o plataformas para la docencia puramente online.

No es todavía el caso de la herramienta que tomamos como base para este proyecto, Classroom presenter 3, puesto que si viene sí que se puede seguir la docencia a través de la conexión TCP, únicamente se pueden seguir imágenes, y no voz como podría ser ahora por lo cual no se podrían recibir explicaciones de lo que se realiza sobre las transparencias, además posee ciertas limitaciones de conexiones, especialmente si se realizan en una red abierta. Pero quitando los problemas lógicos que puede tener hoy en día dicha aplicación contra otras más actuales, lo que importa son las funcionalidades que introduce de manera novedosa y que queremos poder desarrollar ahora también en esta nueva aplicación pero otro desarrollo y funcionamiento.

2.3. Classroom Presenter 3

2.3.1. Inicio aplicación

En esta aplicación, lo primero se nos plantea es la opción de elegir el rol que seamos realizar dentro de la misma. Por una parte está la posibilidad de hacerlo conectado a la red como:

- Estudiante
- Instructor
- Visualización pública

Este es el formato que se utiliza para dar clases, donde el profesor toma el rol de instructor y los estudiantes su susodicho. La visualización pública quiere decir que utilizas un aparato como visualizador de lo que vas a realizar, por ejemplo conectarlo a un proyector directamente para que la salida vaya directamente a dicho aparato. De esta forma por ejemplo puedes tener una aplicación corriendo en el proyector y otra en local en un ordenador propio de tal forma que puedas intercambiar lo que se ve en una y la otra, y que las otras personas no puedan ver el trabajo se va realizando en el pc local hasta que no se hace 'publico'.

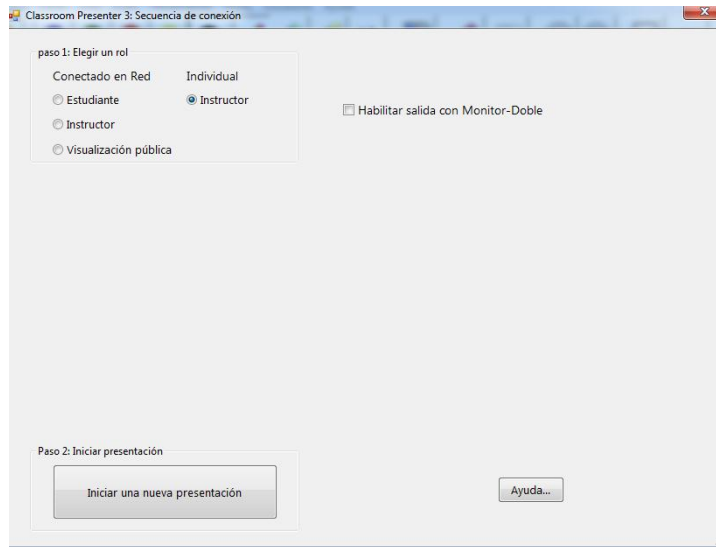


Figura 4.- Inicio aplicación

2.3.2. Rol instructor

En este caso se toma el rol del ‘Profesor’ o instructor, para ello el mismo debe iniciar una conexión con el servidor TCP para que los alumnos se puedan conectar a las transparencias. También podemos hacerlo de manera ‘Individual’ sin conectarnos a la red, para hacer las modificaciones y pruebas queramos hacer directamente sin tener que conectarnos a la red.

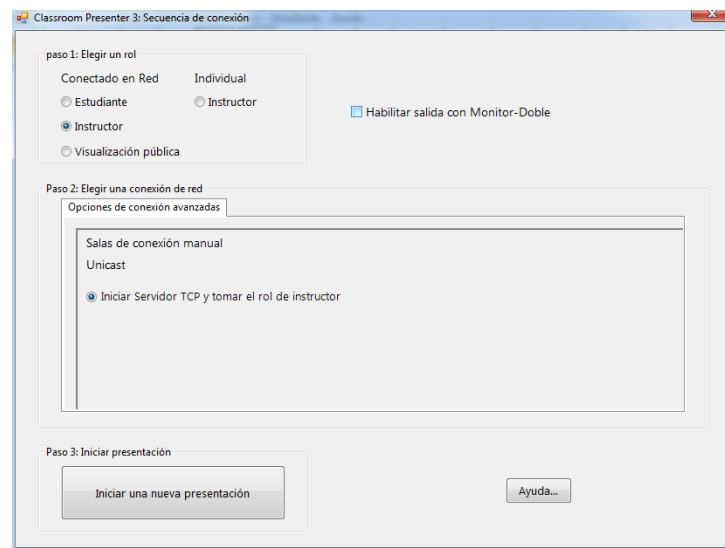


Figura 5.- Modo Instructor

A continuación, esta es la pantalla que se nos abre en primer lugar (sin abrir ninguna diapositiva directamente). Como se aprecia el menú está formado por una barra de menú, y botones de acción.

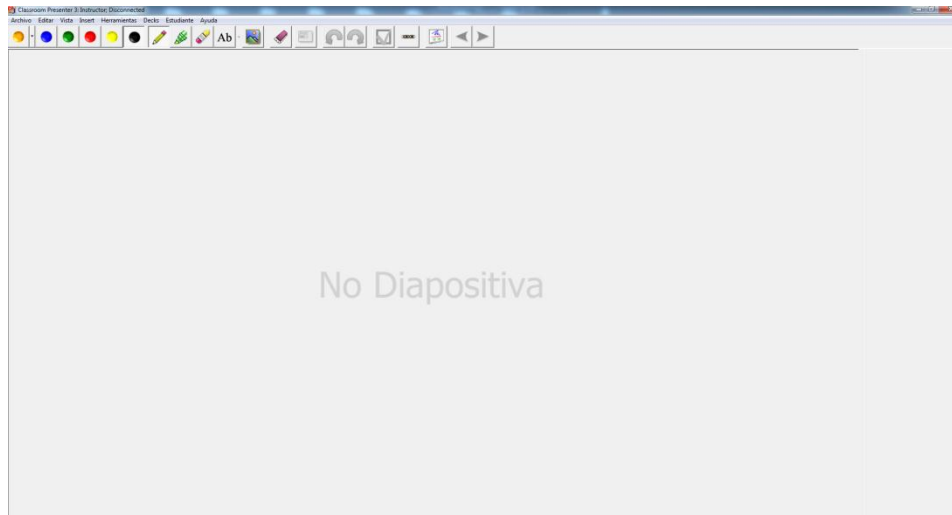


Figura 6.- Vista aplicación

Menú:

Archivo

Mediante esta opción tenemos la posibilidad de abrir nuevas diapositivas ‘Decks’, guardarlas, o incluso exportarlas a otro formato. También podemos guardar las diapositivas con los cambios le hayamos realizado mediante las herramientas que más adelante se verán.

Editar

La única posibilidad de edición dentro la de herramienta es la de retroceder a una acción anterior o avanzar a una acción posterior.

Vista

Con este menú se tiene posibilidad de por un lado, modificar el lugar donde aparecerán las diapositivas una vez cargadas (si en la parte superior, inferior, derecha o izquierda de la vista), así como el ancho de la tira de diapositivas, tanto de vista propia como la que los estudiantes conectados podrán ver. También permite acceder al modo pantalla completa.

Insert

Lo que permite es únicamente insertar imágenes dentro de la transparencia, para ello da la posibilidad de insertarla desde distintos orígenes. Desde archivo, capturadas previamente mediante el teclado, copias, o desde una cámara conectada.

Herramientas

Este es el menú donde encontramos la mayor parte de opciones de configuración, específicamente dentro del submenú ‘Opciones...’, los otros dos sirven por orden para organizar los envíos que realizan los estudiantes, y como para editar el formato en que se formulan las encuestas en la aplicación

Por defecto el sistema presenta una creación de encuestas con 4 opciones (ABCD), que se puede editar:

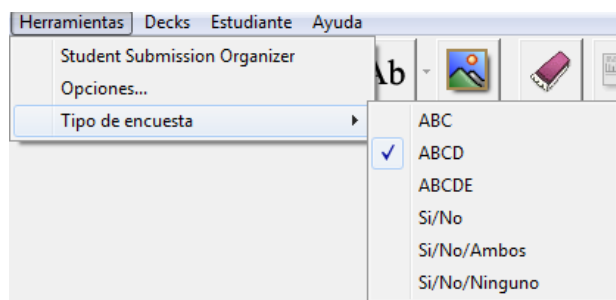


Figura 7.- Tipos de encuesta

Opciones

General

- Habilitar una salida con monitor doble (que ya se da nada más iniciar la aplicación)
- Tratamiento de diapositivas
- Idioma
- Reestablecer opciones predeterminadas
- IP emitida
- Lapicera
- Guardado de archivos
- Delay de transmisión de visualización de las imágenes a los estudiantes (retardo de la emisión).
- Formato de impresión: Páginas por hoja.

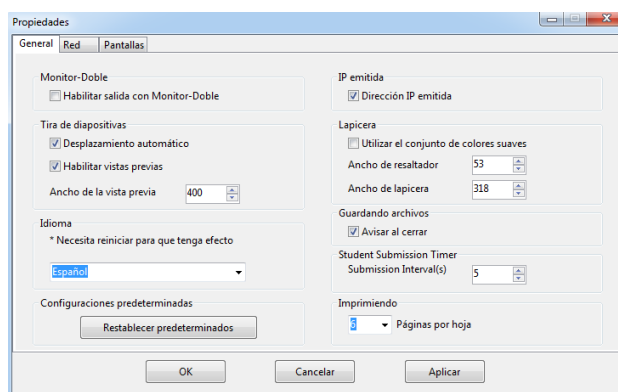


Figura 8.- Opciones – General

Red

Estas opciones son para ajustar la red a la cual se conectan los alumnos.

Pantallas

Estas son opciones son para la configuración de un doble monitor o extender la visualización a otro monitor conectado directamente al ordenador.

Deck

Este menú sirve para el control del estilo de las transparencias así como del slider de las mismas. También da la opción de crear una transparencia en blanco, como si de una pizarra se tratase.

Estudiante

Para controlar el tipo de interacción con el estudiante. Por una parte permite habilitar o no que estos puedan enviar documentos (o transparencias modificadas) al profesor. Esto sirve por ejemplo para pruebas evaluativas donde el profesor lanza una pregunta que los alumnos resuelven y le envían al mismo. Si no se desea recibir nada simplemente se deshabilita. También está la opción de ‘Habilitar votación rápida’ que solo se controla cuando se lanza una pregunta a los alumnos.

Las otras dos opciones tienen que ver con lo que ve el alumno. Si se habilita ‘Forzar navegación vinculada’, el alumno siempre verá lo mismo que el profesor, mientras que si se desactiva tiene libre movimiento para realizar acciones y visualizar transparencias. Mediante ‘StudentNavigationType’ es también controlar lo que el alumno puede y no poder ver o hacer.

Ayuda

Finalmente el menú de ayuda característico en cualquier aplicación. La particularidad principal es la ‘Dirección IP’ esta es de ayuda cuando un alumno no se puede conectar directamente a las transparencias porque la propia aplicación no encuentra el servidor del tutor, entonces puede conectarse directamente mediante la dirección IP que el tutor puede encontrar en dicho submenú de ayuda y proporcionársela al alumno.

Botonera

Además del menú de herramientas, tenemos también a disposición una botonera de acciones que podemos realizar directamente sobre las transparencias o sobre los estudiantes que están accediendo a las transparencias.

Botones de escritura

La primera sucesión de botones se refiere a los botones que permiten realizar trazos y escribir texto sobre las transparencias.

El primero de ellos nos permite elegir un color personalizado con que escribir y los cinco posteriores vienen predefinidos (azul, verde, rojo, amarillo y negro). Los siguientes dos para seleccionar el tipo de trazo, si fino de lapicero y o más grueso como un rotulador. El siguiente nos permite borrar contenido, y el último nos permite insertar cuadros de texto.

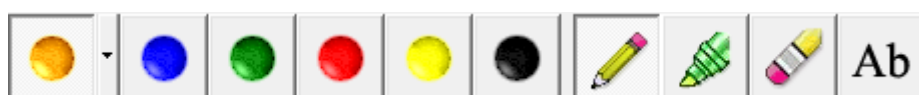


Figura 9.- Botonera escritura

Botones de edición

Estos botones permiten hacer acciones rápidas de edición del contenido que tenemos en pantalla. El primero de ellos permite insertar imágenes, el siguiente permite eliminar todo el contenido insertado previamente en la transparencia, y el tercero reajusta el zoom de la transparencia al tamaño original de esta. Finalmente las últimas dos son para rehacer y deshacer las acciones que previamente hayamos realizado.

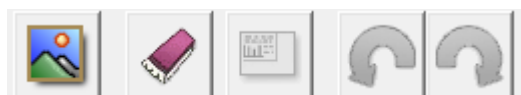


Figura 10.- Botonera de edición

Botones acción con el estudiante

Algunos de estos botones permiten realizar algunas acciones que ya tenemos en el menú previo pero de manera más directa. El primero de los aquí presentes nos permite (si tenemos unas transparencias activas), lanzar una encuesta (que podemos configurar en el menú previamente presentado) para que los alumnos respondan en un espacio de tiempo que vienen definido hasta que el instructor vuelve a utilizar dicho botón. Por otro lado lo que permite el segundo de los botones es los estudiantes vean de manera sincronizada lo que el profesor esta también viendo (navegación sincronizada).



Figura 11.- Botones de acción estudiante

Botones de transparencias

En primer lugar podemos cambiar las transparencias que tenemos cargadas por una pizarra en blanco para poder escribir en cualquier momento sobre ella como si fuera la clásica pizarra de una clase normal, y alternar con las transparencias cuando el profesor desee. Los botones con forma de flecha son para pasar a una ventana anterior o posterior respectivamente.



Figura 12.- Botones de transparencias

2.3.3. Rol estudiante

La mayoría de opciones y elementos son los mismos que ya se tenían en el rol de instructor, la diferencia principal radica en una nueva sección de la botonera, específica para este rol, donde se dan las opciones de sumisión de preguntas y resoluciones a los cuestionarios y encuestas:

- *Botones específicos para los estudiantes*

Cuando entramos como estudiante, este tiene una serie de botones específicos, principalmente preparados para la resolución de las encuestas que los profesores pueden lanzar. Con los botones de respuesta podemos indicar la solución proceden que creemos al problema planteado por el profesor, en el formato que corresponda como estén planteadas las preguntas.

Por su parte, el primer botón de esta botonera, lo que indica es que estamos habilitados a realizar envíos al profesor, de manera que este pueda recibirlos y ver nuestro trabajo, la respuesta a un problema, dudas personas o preguntas sobre la materia por ejemplo. Si no estuviera activado, esta casilla aparecería marcada con un símbolo de prohíbo y por tanto no se podría realizar ningún envío.

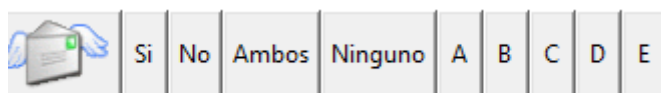


Figura 13.- Botonera de estudiantes

2.4. Soluciones actuales para la docencia

Según una noticia publicada en Europa Press sobre el informe UNIVERSITIC de 2011: *“El 90% por ciento de los profesores y de los estudiantes de las universidades españolas utilizan ya plataformas de docencia virtual, mientras que los centros universitarios disponen de un ordenador por cada 10 alumnos.”*¹³

Hay que tener en cuenta que este informe de 2011, se corresponde con datos de 2010, es decir que son datos de hace 4 años, con lo cual cabe suponer que este porcentaje habrá aumentado todavía durante estos años. Pero en dicha noticia hay todavía más datos a tener en cuenta.

“Las TI proporcionan soporte a la docencia presencial, de manera que la mitad de las aulas están equipadas con un proyector multimedia y ofrecen a todos sus estudiantes conexión a Internet. Las universidades ponen a sus estudiantes alrededor de 80.000 ordenadores, lo que supone un ratio de casi 1 ordenador para cada 10 estudiantes. Además, el 90 por ciento del PDI y de los estudiantes ya utilizan la plataforma docencia virtual institucional.”

Estos datos vienen a reflejar que cada vez más, las TIC (no solamente las “Tecnologías de la información”, sino también las de comunicación, ya que no podríamos hablar de unas sin el avance de las otras), están presentes en la docencia niveles universitarios, y cada vez un mayor número de docentes se apoya en ellas. Incluso el ratio de ordenadores por alumnos comienza a ser destacable, llegando a poderse realizar ya clases prácticas o laboratorios, donde los alumnos tienen un ordenador propio o por grupos para desarrollar los trabajos de la asignatura, o seguir la docencia de la misma.

Un último dato a destacar sería que *“el 90 por ciento de las universidades disponen de un presupuesto propio y diferenciado para las TI y que a pesar de la crisis, se mantiene el presupuesto dedicado a las TI”*. Esto hace destacar la prioridad y esfuerzo de las instituciones en desarrollar este tipo de herramientas, no quedando meramente como complementos para la docencia, sino como elementos imprescindibles a día de hoy y con un poder y evolución cada vez mayor.

Tal y como se comenta en el informe, el 90 por ciento de PDI y de los estudiantes utilizan la plataforma de docencia virtual institucional. Este tipo de plataformas son imprescindibles hoy en día en aquellas universidades e instituciones que trabajan con ellas, por las posibilidades y el material almacenado en ellas, que agiliza y permite nuevas soluciones docentes. No son herramientas para la transmisión de conocimientos o de ayuda directamente, pero sí que son un soporte fundamental para el correcto funcionamiento de las asignaturas o materias.

Dos de las principales plataformas de docencia virtual serían Moodle¹⁴, conocida por sus versatilidad, comodidad así como de trabajar con código abierto lo cual da enormes facilidades para nuevos desarrollos de módulos y herramientas, y Sakai también de código abierto, pero con

¹³ <http://www.europapress.es/portaltic/movilidad/software/noticia-90-profesores-estudiante-universitarios-utilizan-plataformas-docencia-virtual-20111006161958.html>

¹⁴ <https://moodle.org/?lang=es>

un enorme equipo de trabajo dedicado a su desarrollo, y puesto en funcionamiento en diversas universidades como la UPV, y otras prestigiosas universidades en todo el mundo¹⁵.

Por otra parte se encuentran las plataformas comerciales, llevadas por compañías de software, con código cerrado y desarrollo únicamente dentro de la propia empresa. Destaca entre estas Blackboard / WebCT.

2.4.1. Modelo Moodle - Sakai

Son plataformas de aprendizaje, de software libre por lo que su utilización y distribución son gratuitas, además de código abierto por lo cual están en constante evolución gracias a los avances y nuevos módulos desarrollados tanto por el personal de la empresa como por programadores anónimos o colaboradores de la plataforma.

Ofrece ventajas para todos aquellos profesores que quieren enriquecer su trabajo en el aula con el apoyo de un entorno virtual de aprendizaje. Sus principales características, ya que ambos podríamos situarlos en un funcionamiento muy parecido y sin diferenciar una de la otra a nivel general son¹⁶:

- Como expositor de contenidos formativos para ofrecer a los estudiantes apuntes, documentación, recursos, etc.
- Como espacio de encuentro para sus alumnos/as, no sólo para el seguimiento de los contenidos del curso sino también como lugar de debate y red social de aula (protegida y segura) gracias a la utilización de los foros, chat, correo y mensajería, entre otros.
- Como espacio de trabajo en el que los alumnos/as además de acceder a recursos y documentación, podrán también acceder, realizar y entregar sus tareas al profesor/a.
- Como espacio de trabajo colaborativo. Posibilidad de crear y organizar grupos de trabajo.

¹⁵ <http://www.20minutos.es/noticia/1104579/0/>

¹⁶ <http://formacionprofesorado.educacion.es/index.php/es/materiales/270-moodle-plataforma-de-aprendizaje?showall=1>

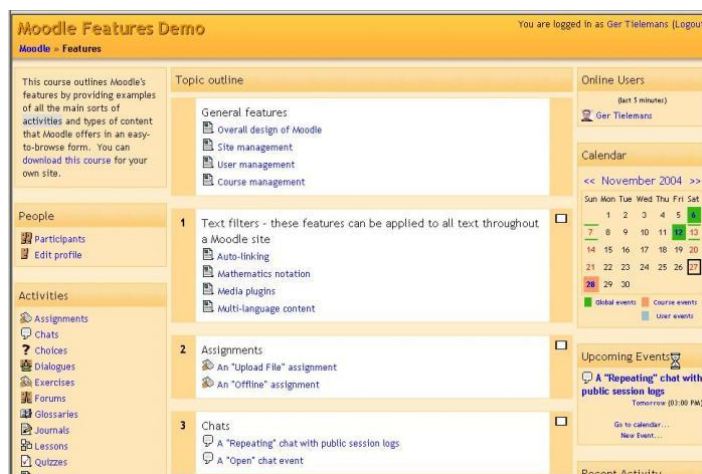


Figura 14.- Vista principal plataforma en Moodle

2.4.2. Modelo BlackBoard / WebCT

En 2005, Blackboard había desarrollado y licenciado aplicaciones de programas empresariales y servicios relacionados a más de 2.200 instituciones en más de 60 países. Estas utilizan entre otras funcionalidades, el programa de aprendizaje en línea (*e-learning*), que les proporciona el programa.

A nivel educativo, se encuentra presente en distintas instituciones relacionadas con la educación como el Servicio Nacional de Aprendizaje (SENA) de Colombia o universidades como la Nacional de Colombia, o la Universidad de Puerto Rico.

Dispone de diversos módulos y herramientas en los que destacan:

- Módulo de Contenidos: Herramienta donde se organizan los materiales didácticos del curso.
- Herramientas de Comunicación: Foro, Correo interno y Chat.
- Herramientas de Evaluación: Exámenes, Trabajos y Autoevaluación.
- Herramientas de Seguimiento y Gestión de Alumnos.



Figura 15.- Vista principal plataforma en BlackBoard

Como ya se ha dicho es de código cerrado y por tanto hay que pagar por sus derechos de uso y utilización, al contrario que ocurría con Moodle y Sakai. Además estas dos últimas poseen un sistema y manejo ligeramente más sencillo que Blackboard, pero por su parte esta tiene unos módulos y herramientas más desarrolladas y fiables al estar testeados y generados por el personal de la empresa y haber pasado los filtros hasta ser abiertos al público.

Pero es gracias a estos servicios, independientemente de su modelo, que se abre un nuevo tipo de educación, desconocido hasta hace algunos años, realizado íntegramente de manera online, gestionados mediante este tipo de herramientas y conocido como E-learning.

2.5. E-learning

El aprendizaje electrónico o E-learning, es la educación a distancia completamente virtualizada, y que se realiza a través de plataformas o centros de E-learning cuyo funcionamiento es 100 % online como puede ser Miriada X, aunque también puede darse el caso de otros de carácter mixto como Massaber¹⁷ o universidades cuyo funcionamiento es de este mismo tipo y están especializadas en este tipo de docencia como la UNED¹⁸.

En este tipo de plataformas se ofrecen desde Masters y Posgrados, hasta cursos específicos de materias determinadas. Este último caso, es el que más en auge esta hoy en día, dando la posibilidad a los alumnos de acceder a necesidades concretas sin necesidad de estudiar una carrera entera para unos conocimientos específicos. Se ahorran los costes de este tipo de carreras extensas y además de conocimientos que van más allá de lo que el alumno puede necesitar.

Si se toma como modelo la plataforma Miriada X, este es el tipo de cursos que ofrece, materias específicas, impartidos en muchos casos por universidades y centros de prestigio que dan confianza al alumno sobre su calidad y la validez de los conocimientos adquiridos en los

¹⁷ <http://www.massaber.es/>

¹⁸ http://portal.uned.es/portal/page?_pageid=93,1&_dad=portal&_schema=PORTAL

mismos. Además este formato online tiene otros beneficios para los alumnos como son el ahorro de tiempo y coste de desplazamientos, la autogestión de horarios y la especialización

Esta formación no es solamente demanda por personas con necesidades específicas concretas de las materias que cursan, sino que también hay gran parte de alumnos que lo toman como una aportación de contenidos para complementar su formación en el área que más les interesa.

El funcionamiento por regla general suele ser el primer lugar, que el alumno elige el curso o incluso cursos (no hay norma que impida al alumno cursar varios al mismo tiempo generalmente), y se le da acceso a una intranet en la plataforma donde tiene acceso al temario teórico, una agenda con la programación, ejercicios prácticos y de autoevaluación. A su vez existe una figura de profesor que es el encargado de resolver las dudas pueda ir teniendo el alumno y de seguir su evolución en el curso.

Es en este punto de su funcionamiento donde enlazamos con un punto anterior sobre las nuevas tecnologías de la docencia. Para la realización de este tipo de curso, especialmente en lo que se refiere a como se visualizan los contenidos, en muchas plataformas se suelen presentar mediante video tutoriales, y las dudas y charlas con los profesores de soporte de la plataforma, se tiene la posibilidad de realizarse mediante video conferencias, incluso algunas veces, se realizan clases que se retransmiten mediante streaming, donde el profesor realiza su explicación en un aula, y los alumnos tienen acceso a la retransmisión, pudiendo llegar a realizar preguntas en vivo al profesor mediante chat o voz.

Es aquí donde podemos apreciar como las tecnologías han revolucionado la docencia como la conocíamos hace algunos años. Si en este tipo de herramientas y soluciones existentes en la actualidad, este tipo de aprendizaje y de plataformas no podrían existir y seguiría únicamente dándose una docencia presencial de aula, donde no podríamos asistir a tales grados de especialidad de las materias, o el número de alumnos sería muy superior al que podría contener una clase, o personas con pocos recursos o falta de posibilidad de desplazamiento, pudieran acceder a la educación.

Otra de las ventajas que presenta el E-learning es la autorregulación de los horarios, en los que tú decides para ajustarlos a tu propio planing. Para ello también está el profesor de soporte, que te ayude en ello y preparar un proceso formativo a un ritmo óptimo.

Finalmente habría que destacar lo que podría considerarse como una ventaja, pero también en un inconveniente según la plataforma y es la certificación de los conocimientos adquiridos. Si vamos a plataformas donde los cursos los imparte universidades como tal, o centros de renombre, el certificado que nos acredite los conocimientos vendrá con el nombre del centro que los ha realizado, pero en el caso de que estos curso se den en plataformas donde no se conoce el centro emisor, o son cursos propios de la plataforma o de una persona en particular, dar un certificado de conocimiento es más complicado. No todos los centros disponen de acreditaciones reguladas y todos queremos demostrar en el curriculum lo que hemos aprendido para el mercado laboral.

2.6. Conclusiones

Una vez visto desde los orígenes de lo que se conoce como educación, las herramientas utilizadas, el avance y utilización de las tecnologías de la información y la comunicación de manera fundamental en los centros docentes, podemos ya centrar donde entraría nuestra aplicación y que hueco ocuparía dentro de este nuevo mundo de interactividad y métodos online.

Como ya se ha comentado previamente, lo que se busca no es una nueva herramienta o plataforma para realizar cursos online o la docencia puramente online. Principalmente porque no vamos a implementar (por ahora) ningún elemento en nuestra aplicación que permita seguir las propias explicaciones que el docente pueda realizar de los contenidos educativos, ni de manera telefónica, ni mediante retransmisión por video o streaming.

Lo que se busca es proporcionar una herramienta a modo de pizarra y proyector, el alumno pueda ver directamente el material docente con el que trabaja el profesor desde su propio navegador web desde el ordenador en el que esté conectado, y que además pueda responder o preguntar a este último, tal como podría ser hoy en día levantarse a la pizarra a resolver un problema o realizar una explicación a los compañeros, pero todo ello desde la pantalla del navegador.

Además al tratarse de una aplicación en tiempo real, el alumno debe seguir las explicaciones para seguir el ritmo de la clase, no pudiendo entrar a posteriori y entonces obtener todos los datos que se ha perdido previamente. No es un repositorio, o un contenedor de información, es una aplicación de soporte docente para la realización de las clases teóricas, una nueva herramienta con la que favorecer la docencia y otorgar nuevas posibilidades, tanto a alumnos, como profesor.

No se busca generar un nuevo modelo de educación, o una plataforma para la gestión educativa, sino más bien interconectar el alumno con el profesor, más allá de elementos tradicionales de un aula. Lo que se busca es que tanto la pizarra, como el proyecto, como incluso los exámenes, estén dentro de la pantalla del navegador web. Un objetivo grande, pero que el trabajo necesario, la voluntad y la dedicación necesaria, a día de hoy, no es tan imposible como podría pensarse hace unos años, y ahí lo que busca nuestro proyecto.



3. Especificación

Esta aplicación se enmarca dentro de las necesidades de la Universitat Politècnica de València para adaptar una aplicación software utilizada en Tablets pc llamada Classroom Presenter 3 a las nuevas tecnologías, y que se pueda generar una aplicación de características similares, pero que pueda funcionar en cualquier dispositivo.

Hasta ahora se viene utilizando la aplicación Classroom Presenter 3, dentro de un aula específica habilitada, que cuenta con Tablets pc con la aplicación ya instalada y preparada para el correcto funcionamiento y desarrollo de la clase. Debido al tiempo transcurrido desde su puesta en marcha, hay ciertos errores y comportamientos que se deben intentar solucionar, pero que dado su código cerrado al tratarse de un producto comercial no se puede trabajar.

Pero el principal problema radica en los dispositivos donde se ejecuta. Para estos también pasan los años, sus prestaciones van quedando atrás en comparación a las de los nuevos dispositivos, y además incluso son ya más caros que otros de nueva fabricación. Es por ello que empieza a aparecer el problema o la necesidad de renovación de estos dispositivos por otros más nuevos, y en estos tiempos de crisis conviene que sea por dispositivos cuyas características y funcionalidades puedan ser más amplias, como por ejemplo podrían ser Tablets con sistema operativo iOS o Android. Pero en estas nuevas Tablets, la aplicación software utilizada hasta ahora no sería posible seguir utilizándola y por lo tanto habría que buscar una nueva solución.

Es aquí donde se produce el inicio de la necesidad que da origen a la aplicación realizada. Para solucionar el problema de la ejecución con independencia del dispositivo hay varias opciones, pero todas y cada una pasan por dos caminos:

- Programación específica, adaptada a cada uno de los sistemas operativos, de tal manera que se tengan que generar aplicaciones idénticas pero con distinta programación dadas las características y lenguajes de programación de cada uno de los dispositivos, ya sean iOS, Android, PCs, Windows Phone, etc. Esto si bien daría lugar a una aplicación más potente y que podría sacar partido a todas las funcionalidades de cada sistema, no daría por otro lado a tener que replicar la aplicación en diversos lenguajes, con un sobrecoste elevado.
- Programación multiplataforma, apoyada en lenguajes HTML5 para elaborar las interfaces mediante un diseño Web, CSS para definir la presentación y el estilo de cada una de estas interfaces, JavaScript para controlar las funciones a realizar dentro de la aplicación y que son lo que dará funcionalidad a la misma, y también PHP para establecer conexiones entre el terminal y el servidor y obtener los datos que necesitamos a la hora de mostrar información que el usuario solicite.

De esta manera podemos generar una aplicación, que lanzaremos directamente en el navegador con independencia del dispositivo, y que funcionará de una manera correcta en todos y cada uno de ellos.

La aplicación se podría dividir en tres partes en su implementación. Una primera parte donde se plantea que se quiere hacer en la aplicación, que características se desea que tenga y que se pueda realizar con ella. Una segunda que sería el diseño de la aplicación, acorde a lo que se busca representar y a las características y servicios que se desea prestar. Finalmente una última parte presentar el modelo de datos en el cual se sustentará la aplicación y que le dará consistencia.

3.1. Características aplicación

Lo que buscamos en primer lugar sería algo que permitiera lo que es la funcionalidad básica de la aplicación que estamos tratando de actualizar y esto sería la interconectividad entre alumno-profesor. Para ello se debería dotar de una serie de herramientas e interfaces que favorecieran dicha interconexión. Otra cosa a tener en cuenta es que esta aplicación debe ser posible ejecutarse en cualquier dispositivo, y por tanto la forma en la que se debe mostrar la información es importante, así como el diseño de las propias interfaces de tal manera que incluso en dispositivos más pequeños como un Smartphone se pudiera llegar a trabajar correctamente sobre la misma. Lo primero a realizar sería ver cuáles son las características y funcionalidades que se desea tenga la aplicación. Es evidente, que generar todas las que aporta la aplicación Classroom presenter 3 en un primer momento sería algo más costoso y que requería de más tiempo y personas invertidas en el proyecto así que hay que centrarse en unos objetivos prioritarios concretos y de ahí ir pasando a otros más secundarios.

Una manera de enumerar lo que nuestra aplicación busca es utilizar lo que se conoce como *estándar MosCoW*¹⁹. Este estándar se utiliza para realizar una lista de requerimientos de la aplicación los cuales se dividen en 4 categorías: Must, Should, Could y Would.

- **Must:** Son los requerimientos esenciales que deben ser incluidos para que el producto o servicio funcione.
- **Should:** Son aquellos que pueden ser integrados para dar mayor beneficio.
- **Could:** Son los que sí existe tiempo se podrían integrar, pero su ausencia no afecta significativamente al funcionamiento del producto o servicio.
- **Would:** Son aquellos que se tiene previsto integrar en un futuro, ya sea debido a que no se han podido integrar en este momento por falta de recursos económicos, tiempo, o la tecnología necesaria. Son los que se llamaría un plus, o extra.

De esta forma, podemos poner los requerimientos que queremos cumpla nuestra aplicación por categorías:

¹⁹ <http://www.bizengine.com/moscow-method/>

1. MUST:

- Panel de administración:
 - Control de la clase: si se desea empezar algo, o cargar algo previamente preparado o ya realizado.
 - Control de los alumnos: Controlar los alumnos que están dentro de la aplicación, y diferenciando entre aquellos que están en ese momento y los que no lo están
- Panel de clase:
 - Transparencias: Interfaz con un espacio considerable donde el profesor cargará las transparencias y los alumnos podrán visualizarlas directamente sin uso del proyecto.
 - Preguntas-Respuestas: Interfaz principal de interconectividad donde el profesor tendrá oportunidad de generar preguntas y cuestiones a sus alumnos, y a su vez estos tendrán la oportunidad de resolverlas.

El profesor podrá ver en cualquier momento lo que sus alumnos han respondido a las preguntas que ha realizado, y además quien y quien no lo ha hecho.

Además los alumnos tendrán también la opción de enviar sus propias preguntas o cuestiones relacionadas con la clase al profesor para que este pueda responderlas o solucionarlas cuando considere oportuno.

2. SHOULD:

Si se quiere maximizar la interactividad entre usuarios de la aplicación lo que se debería buscar es crear herramientas o generar funciones para potenciar este tipo de interacción, si se toma como ejemplo las funcionalidades que tenía la aplicación base del proyecto entonces:

- Botonera de escritura: Que permita al docente nuevos elementos y escribir o dibujar sobre las transparencias a modo de pizarra para resaltar elementos, añadir anotaciones, etc.
- Opciones: Poder deshabilitar la interconectividad en un momento dado, o deshabilitar que los alumnos puedan realizar preguntas.
- Control transparencias: Que los alumnos vean directamente las transparencias en las que se encuentra el docente a modo de streaming de lo que esta este último explicando para que así se pueda controlar que los alumnos están directamente viendo lo que el profesor está explicando y no otra cosa distinta.



- Gestión datos por base de datos: Las respuestas y preguntas realizadas por los alumnos, podrían ser almacenadas en una base de datos externa para que el docente a posteriori pudiera ver lo que los alumnos han respondido a sus preguntas sin tener que volver a acceder a la clase (si la ha guardado) y ver ahí las respuestas de sus alumnos

3. COULD:

Existen otros elementos que pueden ayudar sobre todo a una mayor utilidad de la herramienta. No modifican su función principal, sino que le añaden nuevos elementos para convertirla en algo más potente:

- Deshacer – Rehacer acciones: Tener la posibilidad de rehacer o deshacer acciones que se hayan hecho en cualquier momento, como modificar transparencias, o deshacer el envío de una pregunta a los alumnos.
- Intercambio de roles: En un momento dado puede que el profesor quiera que el alumno sea el que realice la explicación y en tal caso cederle a el alumno el control de las transparencias y que sea este quien las controle mientras explica algún punto.
- Formar grupos: Organizar alumnos en grupos a los que o bien plantear conjuntamente trabajos o que respondan preguntas entre todos los integrantes del mismo. En vez de realizar envíos individuales, se hace un único envío a todos los integrantes del grupo cuya respuesta pueden editar en tiempo real entre todos ellos. También para la evaluación de las preguntas de tal forma que una misma nota, fuera para todo el grupo.
- Calendario: Incluir un calendario con las tareas programadas y el horario de las clases para conocimiento de los alumnos con las fechas de entregas.

4. WOULD:

Muchos de las funciones aquí contempladas no se plantean incluir en un primer momento, porque o bien tendrían ciertos inconvenientes debido a la tecnología empleada o bien al correcto uso que se pudiera hacer las mismas:

- Idiomas: Habilitar que se puedan traducir automáticamente los contenidos o textos de la aplicación (incluso aquellos en tiempo real) mediante herramientas o aplicaciones de traducción automática. Tiene el inconveniente que podría darse el caso que las traducciones no sean fieles a la idea original, pero es una función interesante de estudiar su implantación.

- Chat: Habilitar funciones de chat entre personas dentro de la aplicación, ya sea de profesor a estudiante, o entre estudiantes, aunque esto último podría conllevar problemas como distracciones por parte de los alumnos o un uso indebido.
- Mensajería interna: Envío y almacenamiento de correo interno, para lo cual haría falta implementar dicho servidor de correo, y ponerlo en funcionamiento lo que es un coste excesivo de tiempo actualmente.
- Repositorio de material: Generar un repositorio al cual pudieran acceder los alumnos mediante la aplicación y descargarse los contenidos (transparencias, apuntes, etc.), material extra o todo aquello que el profesor desee subir al mismo. Estudiar las posibilidades de este repositorio o la integración con otras plataformas utilizadas en la universidad como *Poliformat*²⁰
- Almacenamiento de trabajos: Habilitar un espacio de Google Drive (o utilización de *Poliformat*) donde los alumnos pudieran subir sus trabajos directamente a través de la aplicación, y de esta manera el profesor tenerlos almacenados en un espacio común y relacionado con la clase. También se podría habilitar el feedback con las respuestas o comentarios del docente sobre los trabajos, y que llegarán directamente a los alumnos a través de la aplicación.

3.2. Diseño

Antes de ponerse manos a la obra con el desarrollo de la aplicación hay que tener claro qué es lo que se quiere ofrecer, pero también como y de qué manera hacerlo. En primer lugar se han mostrado las características que tendrá la aplicación una vez finalizado su desarrollo. Hay que tener en cuenta que su desarrollo estará por una parte terminado cuando sea funcional la herramienta, y por otro se llegaría a un desarrollo completo de la misma si se ejecutarán todas las características que se ha planteado tenga la aplicación. Por tanto nos encontramos que va a ser una aplicación con un desarrollo y diseño inicial para una funcionalidad básica, y que estará abierto para una evolución en la misma para habilitar las nuevas herramientas y funciones a medida vaya siendo posible su desarrollo.

Por ello, una vez claras las características, vamos a proceder a ver como mostrarlas. En este caso se ha buscado un diseño enfocado a que los usuarios de la aplicación, con independencia del rol que ejerzan, tengan todos los elementos de la misma en una misma interfaz principal, sin necesidad desplazarse entre páginas para ver más contenidos. También para el desarrollo futuro de nuevos elementos no indicados en el diseño inicial, se tomará el mismo patrón de diseño y se introducirán en la parte inferior de la página, para que únicamente sea necesario un desplazamiento vertical de la misma para ver todos los contenidos.

Como es lógico habrá ciertas diferencias en los elementos presentes con dependencia del rol que ejerza la persona que está visualizando la aplicación.

²⁰ <https://poliformat.upv.es/>

Para el diseño se ha utilizado una aplicación para la generación de prototipos mediante mockups llamada *Pencil*²¹, mediante la cual se han elaborado las pantallas correspondientes a los dos roles existentes en la aplicación y como deberían quedar distribuidos los elementos dentro de la interfaz de una página web, ya que hay que recordar al ser una aplicación web, se visualizará a través de una navegador.

3.2.1. Rol Docente

El rol docente debe tener unas herramientas propias de gestión de la aplicación, así como una visualización de los contenidos adaptada a lo que serán sus preferencias.

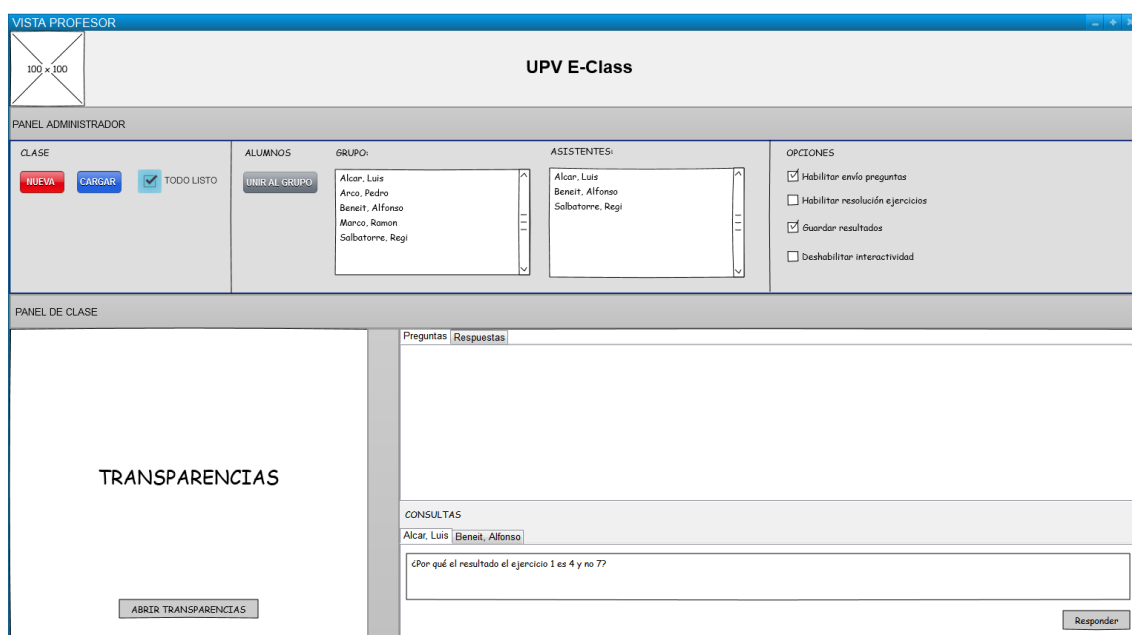


Figura 16.- Diseño preliminar rol docente para la aplicación

Se compone de una primera parte cabecera donde aparece el logo de la aplicación y el título y que será igual también en el rol de alumnos. La segunda parte está formada por un panel de administración con las herramientas necesarias para llevar un control de la aplicación y de gestionar la misma. Se divide en tres grupos,

- **Clase:** Donde se permite crear una nueva o bien abrir una que haya sido ya preparada previamente. Hay que tener en cuenta que contenidos como las preguntas o las transparencias se utilizaran en la clase se pueden haber realizado previamente y cargarse directamente a posteriori.
- **Alumnos:** Se puede invitar a nuevos alumnos a la clase, controlar cuales están ya invitados y cuáles de aquellos invitados están presentes en este momento en la lección.

²¹ <http://pencil.evolus.vn/>

- Opciones: Aquí se encuentran otras opciones principales correspondientes al funcionamiento de la aplicación en sí que son:
 - Habilitar envío de preguntas: Que los alumnos no puedan enviar sus preguntas al profesor hasta que este lo vuelva a indicar.
 - Habilitar resolución de ejercicios: No permitir la resolución de ejercicios propuestos hasta que se indique así.
 - Guardar resultados: Guardar los resultados de las preguntas en una base de datos externa.
 - Deshabilitar interactividad: Deshabilitar que los alumnos vean en directo los movimientos de transparencias del profesor o cualquier otra acción pudiera estar realizando.

En la segunda parte de la misma lo que sería el panel de clase, se subdivide en lo que sería la parte correspondiente a las transparencias que se están visualizando en la clase, y por otra el panel de interacción directa con los alumnos. Este panel cuenta con dos secciones, una para la realización de las preguntas y visualizado de las respuestas de los alumnos, y otra donde se ven las preguntas que los alumnos han realizado al docente y se habilita para poder responderlas en el mismo momento le son recibida o cuando desee. Pero la parte de preguntas sería algo más compleja de lo que ahí aparece por eso su diseño ha sido realizado separado:

The figure displays two screenshots of a software interface for a teacher's role, specifically for managing questions and answers.

Left Screenshot (Preguntas):

- At the top, there are tabs for "Preguntas" and "Respuestas".
- The "Preguntas" section is active, showing a sub-section "Problemas" with a text input field labeled "Escribe aquí la pregunta del problema". Below it are "ENVIAR" and "BORRAR" buttons.
- Below "Problemas" is the "Cuestiones" section, with a text input field labeled "Escribe aquí la pregunta del cuestionario / Encuesta".
- Under "Cuestiones", there are four pairs of radio button options, each labeled "Opción 1" through "Opción 4".
- At the bottom of the "Cuestiones" section are "ENVIAR" and "BORRAR" buttons.

Right Screenshot (Respuestas):

- At the top, there are tabs for "Preguntas" and "Respuestas".
- The "Respuestas" section is active, showing a sub-section "Problema 1" (selected) and "Cuestionario 1", "Cuestionario 2", and "Cuestionario 3".
- Below the tabs is a table with two columns: "ALUMNO" and "RESPUESTA".
- The table contains the following data:

ALUMNO	RESPUESTA
Alcar, Luis	
Beneit, Alfonso	Tan rápido como antes
Salbatorre, Regi	Nunca se puede

Figura 17.- Diseño preliminar rol docente para la aplicación – Preguntas / Respuestas

En la parte lateral izquierda se puede ver lo que sería la pestaña correspondiente a las Preguntas. En la superior se encuentra una zona habilitada la edición de preguntas escritas directamente y enviadas a los alumnos para su resolución también escrita. En la parte inferior hay otra sección preparada para la generación de preguntas de tipo test o cuestionarios.

Por su parte en el lateral derecho se ve lo que correspondería a la zona de Respuestas. Por cada uno de los problemas o cuestiones generadas, hay una tabla en la cual aparece por un lado el nombre del alumno que ha enviado su respuesta, y por otra la respuesta.

Finalmente como se ha indicado previamente, este es un diseño preliminar para la aplicación funcional en primera instancia, cualquier nuevo elemento que se integrará se añadiría a continuación de estos dos elementos del panel ya que son aquellos principales de la aplicación.

3.2.2. Rol Estudiante

El diseño de las interfaces para los estudiantes se simplifica al eliminarse los elementos de gestión de la aplicación, clase o alumnado presente, además se da más presencia al elemento de las transparencias.

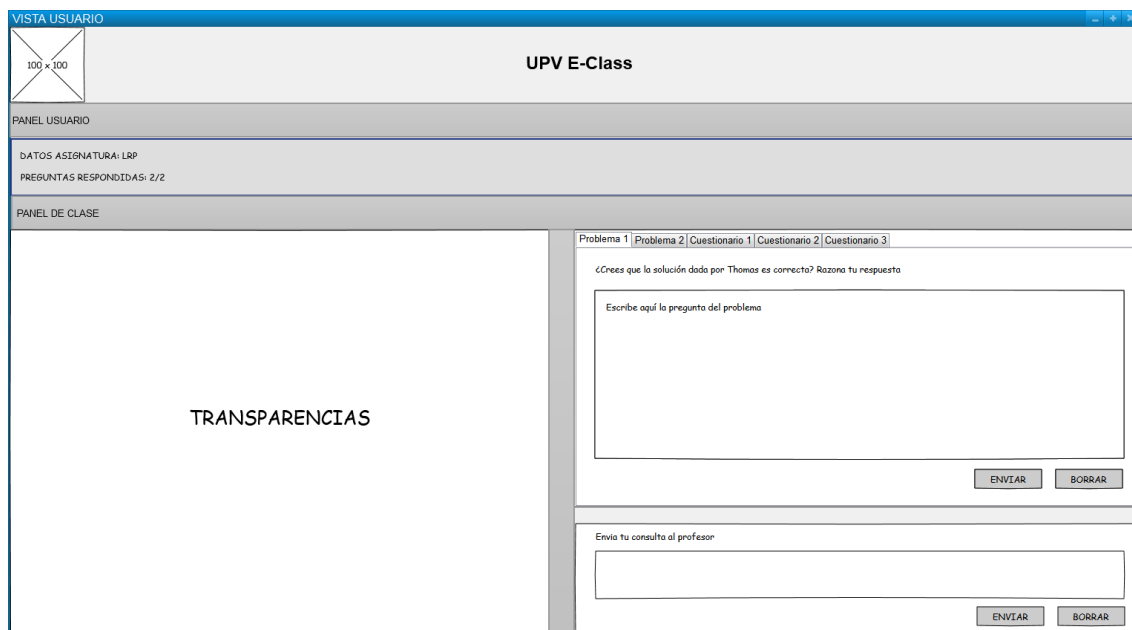


Figura 17.- Diseño preliminar rol estudiante para la aplicación

Para la parte del alumnado se han eliminado los controles del panel de administrador y se han introducido dos campos que nos indican por un lado la asignatura en la que se encuentra ahora mismo, así como un contador para visualizar aquellas preguntas realizadas por el profesor ha respondido y cuáles no.

En lo que respecta al panel de clase, se aumenta el tamaño dedicado a las transparencias (por si no quiere visualizarlas a tamaño completo que no tenga la necesidad de hacerlo). La parte correspondiente a la interacción está totalmente dedicada al envío de respuestas o generación de preguntas al profesor. El alumno podrá ver todas y cada una de las preguntas se le han enviado, aunque el poder responder a ellas o modificarlas vendrá por parte del profesor y las opciones de control que tenga habilitadas en ese momento.



4. Implementación

Una vez presentado el diseño que presentará la aplicación inicialmente es hora de mostrar los puntos más destacados de la implementación del código que hacen posible la ejecución correcta de la aplicación y de sus funciones preestablecidas. También es importante mostrar aquellos que han supuesto una mayor dificultad o que debido a su complejidad ha sido necesario variar el objetivo inicial.

Todas y cada una de las funciones establecidas y en funcionamiento dentro de la aplicación han sido testeadas y comprobadas mediante distintas pruebas, tanto desde la parte docente como la parte alumno mediante clases demo y sucesión de acciones dentro de la aplicación. Otro detalle importante en lo que a implementación se refiere es contar con las herramientas necesarias para su desarrollo. En este caso se ha trabajado siempre en la programación en un sistema Linux, distribución Mint, también se ha utilizado un control de versiones mediante github²², de esta manera además de controlar los diferentes avances de la programación de la aplicación, podemos acceder al repositorio con todo el código de la aplicación de manera online y actualizado < <https://github.com/ethelien/classroom>>.

4.1. Primeros pasos: Activando la aplicación

Lo primero que se debe hacer para comenzar a desarrollar la aplicación es activar y validarnos en las herramientas que van a ser claves a la hora de realizar la aplicación. Para ello debemos seguir los pasos correspondientes para poder desarrollar una aplicación utilizando *Google Drive Realtime API*.

En primer lugar lo que se nos solicita es la activación de nuestra aplicación como proyecto dentro de su consola de desarrollo, de esta manera se nos generaran una serie de credenciales que nos serán necesarias para utilizar dichos servicios y herramientas dentro de nuestra aplicación. Para ello se va a generar un proyecto de nombre “Classroom” que será sobre el cual posteriormente se realizará la programación para generar la aplicación.

²² <https://github.com>



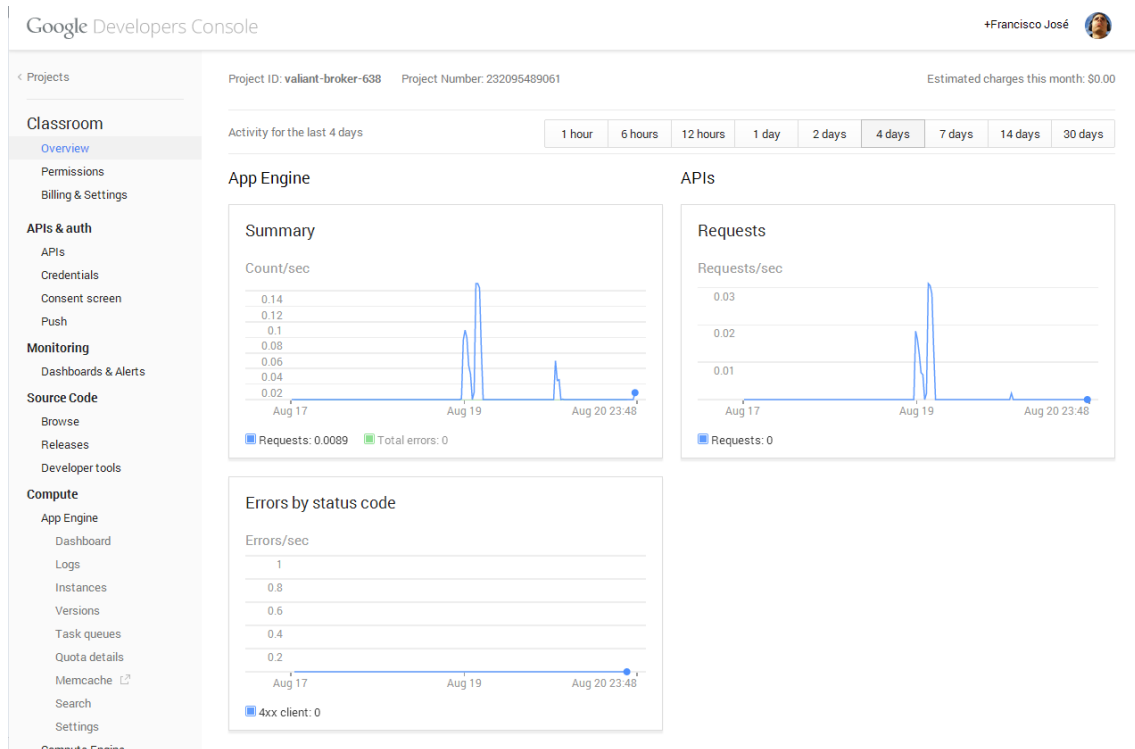


Figura 18.- Google Developer Console - Principal

Una vez creado el proyecto es necesario por una parte:

- Activar las APIs que nuestra aplicación va a necesitar acceder para su ejecución, y cuyas herramientas vamos a utilizar en la funcionalidad de la misma.

NAME	QUOTA	STATUS
BigQuery API	0%	ON
Drive API	0%	ON
Drive SDK		ON
Google Cloud Deployment Manager API	0%	ON
Google Compute Engine Replica Pool API	0%	ON
Google Compute Engine Resource Views API	0%	ON

Figura 19.- Google Developer Console - APIs

- Generar las credenciales necesarias para el acceso a nuestra aplicación. Por cada una de las URLs donde nuestra aplicación vaya a ser ejecutada se necesitan generar credenciales (*Client ID for web application*), para que pueda ser ejecutada. También se generarán unas credenciales públicas para el acceso a contenidos externos dentro de la aplicación (como las transparencias).

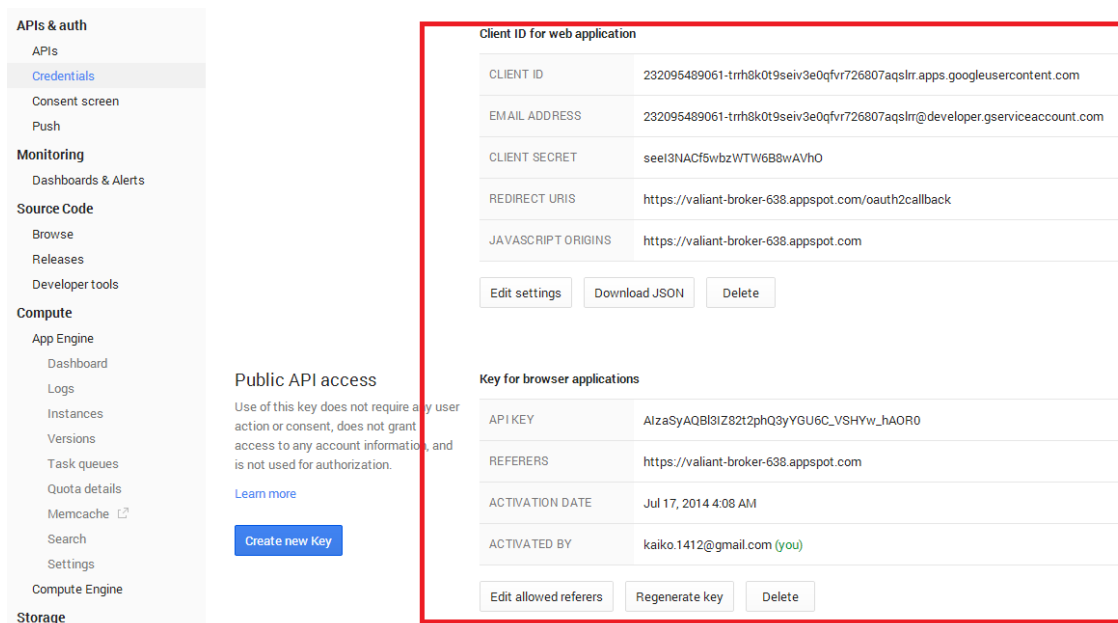


Figura 20.- Google Developer Console - Credentials

La dirección URL donde esta almacenada la aplicación es la que aparece en la imagen <<https://valiant-broker-638.appspot.com/>>, esta dirección alojada dentro de *Google App Engine*²³, que es un servicio de alojamiento web que presta Google de forma gratuita hasta determinadas cuotas, y que permite ejecutar aplicaciones sobre la infraestructura de Google. Los motivos de alojar la aplicación en dicho servidor son principalmente dos:

1. Tiene una capacidad de 500MB de almacenamiento y permite hasta 5.000.000 visitas mensuales de manera gratuita.
2. Esta implementado para soportar lenguajes de programación Python, Java, Go y PHP y dado que este primero no suele ser un lenguaje normalmente soportado en servidores (especialmente gratuitos) y que esta participe en la aplicación.

Para poder ejecutar el compilar nuestro código y posteriormente subirlo alojamiento es también necesario descargarse una SDK (Kit de desarrollo software o *software development kit*) para Python de Google App Engine <<https://developers.google.com/appengine/downloads?hl=es>>, de esta manera el código que programemos se compilará de manera local en nuestra máquina y será posteriormente subido a la dirección que tengamos indicada en la configuración de la propia app engine.

Para ello hay que enlazar nuestro proyecto generado previamente y generar una aplicación dentro de este servicio (de tipo web), será donde se almacenará el proyecto con toda su configuración y código.

²³ <https://appengine.google.com>



Application: valiant-broker-638 [High Replication] [Community Support](#) [My Applications](#)

Main

- [Dashboard](#)
- [Instances](#)
- [Logs](#)
- [Versions](#)
- [Cron Jobs](#)
- [Task Queues](#)
- [Quota Details](#)

Data

- [Datastore Indexes](#)
- [Datastore Viewer](#)
- [Datastore Statistics](#)
- [Blob Viewer](#)
- [Prospective Search](#)
- [Text Search](#)
- [Datastore Admin](#)
- [Memcache Viewer](#)

Administration

- [Application Settings](#)**
- [Permissions](#)
- [Blacklist](#)

Basics

Application Title: Classroom

Displayed if users authenticate to use your application.

Application Identifier: valiant-broker-638

Use this identifier in the application's app.yaml or appengine-web.xml.

Service Account Name: valiant-broker-638@appspot.gserviceaccount.com

Use this name when interacting with external services on behalf of your application.

Application Default Version URL: <http://valiant-broker-638.appspot.com>

Application Identifier Alias: valiant-broker-638.appspot.com

Between 6 and 30 characters. Provides an alternative URL to access your application through appspot.com. It can be used to enable Channel, XMPP, Email, and SSL access for your application.

<http://valiant-broker-638.appspot.com>

Datastore Replication Options: High Replication

Uses a highly replicated Datastore that synchronously replicates data across multiple locations simultaneously.

Google APIs Console Project Number: 232095489061

The Google APIs Console allows you to configure other Google APIs, like Translate or Cloud Storage.

Cookie Expiration: Default (1 Day)

App Engine uses a cookie to keep users logged in to your application.

Authentication Type: Google Accounts API

The [Google Accounts API](#) includes all Google Accounts. [Learn more](#)

Google Cloud Storage Bucket: [valiant-broker-638.appspot.com](#)

Blobstore operations will operate on objects in this bucket.

Figura 21.- Google App Engine

Aquí donde se obtiene la URL donde estará almacenada la aplicación, así como su ID (identificador) para llamadas. Y con esta URL es con la que, hemos creado nuestras credenciales para poder tener permiso de acceso y utilización de las distintas APIs seleccionadas en el proyecto.

Además de la configuración del proyecto y del servicio de alojamiento, hay que también configurar la API de Google Drive y que será la que controlará todo el sistema de interconectividad entre roles, la admisión de nuevos alumnos y la apertura de las transparencias desde archivos almacenados en dicha aplicación de Google.

Google apis

Classroom

- Overview
- Services
- Team
- API Access
- Billing
- Reports
- Quotas
- BigQuery
- Drive SDK**

Google Drive SDK

App ID: 232095489061

Application Info

Application Name
If you would like to customize your application's name within Drive, you can specify it here. Note that the standard i authorization dialogs.

Classroom

Short Description
A one-line description of your application.

Full Description
A full description of your application.

Figura 22.- Google Drive SDK

Es necesario incluir una serie de información para que la aplicación pueda ejecutarse correctamente como:

- Nombre
- Logos
- Drive Integration
 - Client ID: Obtenido en las credenciales sobre la URL del servidor de alojamiento.
 - Adittional Scopes: Necesarios para incluir nuevos alumnos o invitados a la aplicación y que puedan hacer uso de ella. Lo que hacen es permitir la la utilización de las herramientas encargas de proporcionar información de contacto (email y nombre) de los usuarios.
 - <https://www.googleapis.com/auth/userinfo.email>
 - <https://www.googleapis.com/auth/userinfo.email>
 - Install URL: La URL donde se ejecuta la aplicación
- Multiple File Support: Para poder abrir varios archivos dentro de una misma instancia de la aplicación si fuera necesario.
- Create With: Para poder crear nuevos documentos (clases) dentro de la aplicación y almacenarlos.

Con todo esto incluido, ya se tiene todo en lo que se refiere a la configuración online de la aplicación para poder ejecutarla, pero también que realizar ciertas acciones dentro de nuestro código para que la posterior conexión entre servicios se realice de manera correcta y con la autorización necesaria para su ejecución por parte del usuario por una parte para ejecutar la aplicación resultante, y por parte del desarrollador por otra para tener accesos y permisos de compilación.

Antes de empezar a programar directamente, en este caso en concreto vamos a utilizar como base la aplicación *Playground*, que utiliza Google para que los usuarios prueben la API Realtime y vean su funcionamiento. El código es abierto para cualquier usuario <<https://github.com/googledrive/realtime-playground>> así que lo descargaremos y trabajaremos sobre el. La ventaja principal es que de esta manera ya tenemos todas las librerías y elementos que se van a utilizar dentro de un mismo proyecto por lo cual lo que se deberá hacer es modificar los existentes y añadir de nuevos para obtener el resultado que queremos. Pero para ello primero hay que añadir una serie de parámetros en algunos archivos para que nuestro código se conecte directamente con la URL en la cual hemos generado nuestro proyecto de desarrollo:

- **app.yaml**: Este es el archivo de ejecución inicial sobre el cual se invoca a los demás archivos para su ejecución. Presenta una serie de manejadores para indicar donde se encuentran cada una de las carpetas dentro del servidor en las que buscar los diferentes archivos, así como una serie de reglas que debe cumplir, pero lo que más nos interesa en



este caso es que en la primera línea de código se debe indicar el ID de la aplicación (el obtenido dentro del app engine).

```
application: valiant-broker-638
version: 1
runtime: python27
api_version: 1
threadsafe: true
```

Figura 23.- app.yaml – ID aplicación

- **rtpg.js:** Esta es la librería Javascript principal de la aplicación, en ella entre otras cosas, se introducen los datos de las credenciales que hemos obtenido previamente al dar de alta y configurar nuestra aplicación dentro de la consola de desarrollo de Google y que serán los correspondientes a la URL donde nuestra aplicación se ejecutará. El APP ID es la cifra numérica se encuentra también al principio del CLIENT ID y que se corresponde al número de proyecto dentro la consola de desarrollo.

```
/** Your Application ID from the Google APIs Console. */
rtpg.APP_ID = '232095489061';

/** Your application's Client ID from the Google APIs Console. */
rtpg.CLIENT_ID = '232095489061-trrh8k0t9seiv3e0qfvr726807aqs1rr.apps.googleusercontent.com';
```

Figura 24.- rtpg.js – APP ID & CLIENT ID

- **filepicker.js:** Esta librería es la que controla el acceso a archivos almacenados dentro de Google Drive y permite su carga mediante una ventana emergente de búsqueda y su posterior visualización (Se utiliza para la apertura y carga de las transparencias). En este caso los datos a introducir son la *Public Key* que se ha generado en la consola de desarrollo y el *clientId* también generado en el mismo lugar (igual que el anterior introducido en el rtpg.js).

```
function initPicker() {
    var picker = new FilePicker({
        apiKey: 'AIzaSyAQB13IZ82t2phQ3yYGU6C_VSHYw_hAOR0',
        clientId: '232095489061-trrh8k0t9seiv3e0qfvr726807aqs1rr',
        buttonEl: document.getElementById('abrir_transparencias'),
        onSelect: function(file) {
            console.log(file);
            alert('Selected ' + file.title);
        }
    });
}
```

Figura 25.- filepicker.js – apiKey & clientId

Y con esto ya tenemos lo necesario para comenzar a programar nuestra aplicación en cuanto a lo que a código se refiere. El resultado en este momento sería la misma aplicación base que hemos tomado, pero sobre nuestra dirección URL.

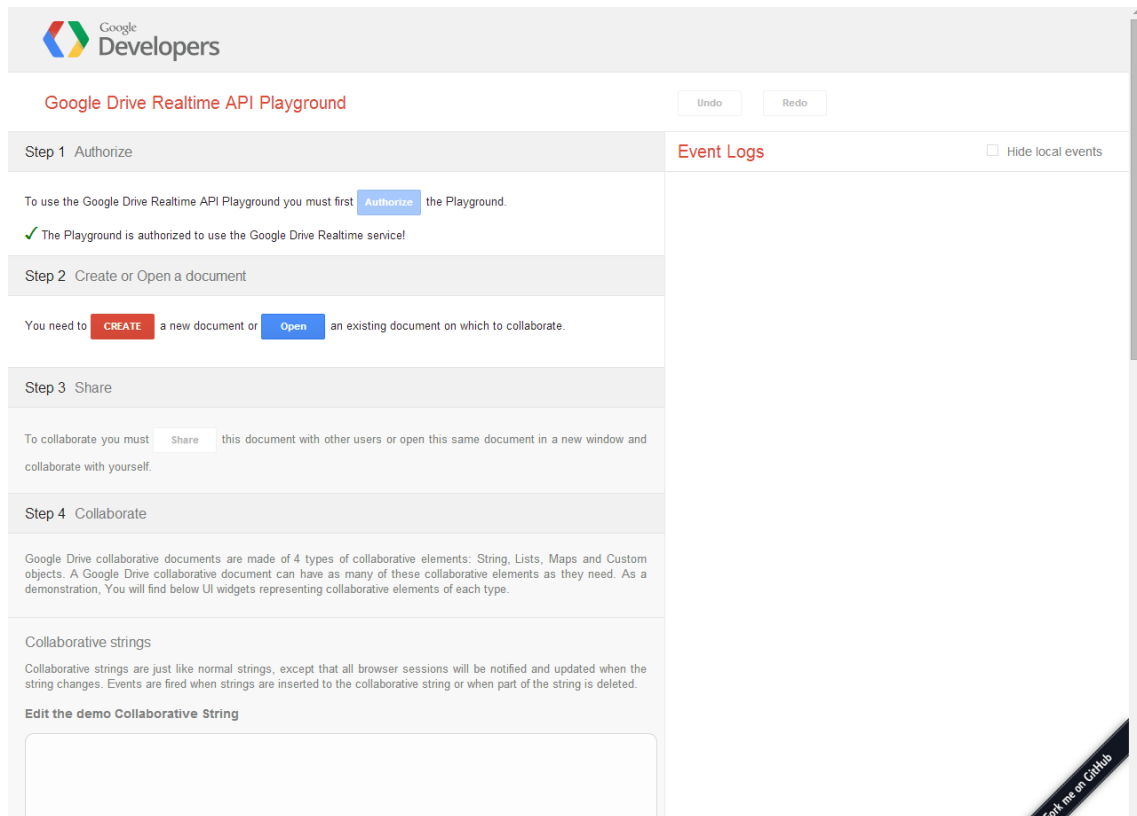


Figura 26.- Realtime API Playground

4.2. Lenguajes de programación

Durante el desarrollo de la aplicación se utilizan distintos tipos de lenguajes de programación cada uno con una funcionalidad. En este caso en concreto contamos con cuatro: JavaScript, HTML, CSS y Python.

- **JavaScript:** Es el lenguaje de programación interpretado, orientado a objetos, que es pilar principal de la aplicación. Mediante el controlamos todos los sistemas de interactividad, el acceso a las herramientas APIs proporcionadas, el control de eventos de la aplicación o los resultados de las funciones²⁴.

²⁴ <http://www.w3schools.com/js/default.asp>

- **HTML:** Es el lenguaje predominante para la elaboración de páginas web, que se utiliza para describir y transcribir la estructura e información del texto, así como complementarla con distintos elementos²⁵.
- **CSS:** Es utilizado para definir la presentación de un documento estructurado elaborado ya sea en HTML o XML²⁶.
- **Python:** Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y en menor medida, programación funcional. Se utiliza en este caso principalmente en la compilación de lo código y en su funcionamiento dentro del servidor de google app engine²⁷.

Esto en lo que sería la funcionalidad central de la aplicación pero si añadiéramos la funcionalidad del almacenamiento de información de manera externa mediante bases de datos deberíamos utilizar también otros lenguajes como PHP y MySQL.

4.3. Autorización de la aplicación

Lo primero que debemos hacer dentro de la aplicación será aquella darle permiso a la misma a estar conectada a nuestra cuenta de correo de tal manera que podamos administrarla y utilizarla pero que también pueda leer nuestros datos personales. Sin esta autorización no podríamos utilizar las funcionalidades que se nos proporcionan y están aparecerían deshabilitadas.

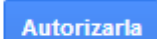
Para poder utilizar la aplicación primero debes 

Figura 27.- Autorización de aplicación

Una vez que hemos actualizado la aplicación se ejecutará el siguiente código:

```
rtpg.afterAuth = function() {  
  $(rtpg.CREATE_DOC HOLDER_SELECTOR).removeClass('disabled');  
  $(rtpg.AUTHORIZED_MESSAGE HOLDER_SELECTOR).show();  
  
  $(rtpg.CREATE_SELECTOR).click(function() {  
    $(rtpg.CREATE_SELECTOR).addClass('disabled');  
    $(rtpg.OPEN_SELECTOR).addClass('disabled');  
    rtpg.realtimeLoader.createNewFileAndRedirect();  
  });  
}
```

²⁵ http://www.w3schools.com/html/html_intro.asp

²⁶ http://www.w3schools.com/html/html_intro.asp

²⁷ <https://www.python.org/>

Si nos fijamos, al principio tanto los botones para crear una nueva clase, como para cargar una ya creada estaban desactivados, mediante dicho código se activan estas dos funcionalidades ya que nuestra aplicación tiene permiso para acceder a los datos de nuestra cuenta entre ellos a los archivos almacenados en Google Drive ya que es aquí donde se guardarán todas las clases que creemos. Si lo que deseamos es crear una nueva clase, se ejecutará como resultado una función que generará un archivo de tipo Mime dentro de nuestro Drive y que será la clase.

```

rtclient.RealtimeLoader.prototype.createNewFileAndRedirect = function() {
  // No fileId or state have been passed. We create a new Realtime file and
  // redirect to it.
  var _this = this;
  rtclient.createRealtimeFile(this.defaultTitle, this.newFileMimeType, function(file) {
    if (file.id) {
      _this.redirectTo([file.id], _this.authorizer.userId);
    }
    // File failed to be created, log why and do not attempt to redirect.
    else {
      console.error('Error creating file. ');
      console.error(file);
    }
  });
}

```

En este archivo se almacenarán tanto los usuarios que están colaborando (invitados), como toda la actividad producida en los modelos de tiempo real (*model*), además se guardará de manera automática de tal forma que cuando se produzca una interacción o se añada un nuevo usuario a la clase, esta guardará los datos referentes a dicho evento. Además el usuario que el archivo será el administrador del mismo.

Si le diéramos a cargar se ejecutaría en primer lugar una ventana emergente para que seleccionemos la clase que deseamos cargar (*picker*).

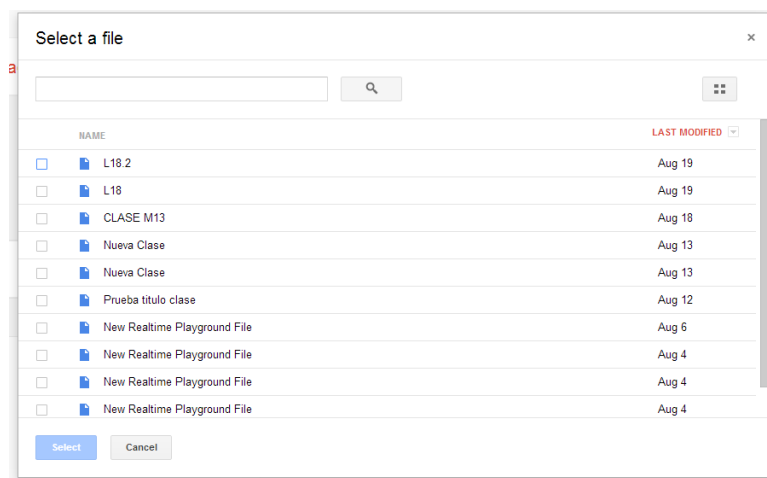


Figura 28.- Cargar clase

```

rtpg.popupOpen = function() {
    var token = gapi.auth.getToken().access_token;
    var view = new google.picker.View(google.picker.ViewId.DOCS);
    view.setMimeTypes("application/vnd.google-apps.drive-sdk."+
rtpg.realTimeOptions.appId);
    var picker = new google.picker.PickerBuilder()
        .enableFeature(google.picker.Feature.NAV_HIDDEN)
        .setAppId(rtpg.realTimeOptions.appId)
        .setOAuthToken(token)
        .addView(view)
        .addView(new google.picker.DocsUploadView())
        .setCallback(rtpg.openCallback)
        .build();
    picker.setVisible(true);
}

```

Este código lo que hace es abrir un Pop Up (ventana emergente) como el que se ve en la imagen anterior. Si seleccionamos una de las clases de la lista, recuperara sus valores, lo convertirá en la vista a visualizar y finalmente realizara .setCallBack(rtpg.openCallback) que lo que hará es recargar todos los valores de la página con la información que se encontraba almacenada en la clase recuperando todas las posibles preguntas, respuestas, consultas o usuarios estuvieran ya en ella

```

rtpg.openCallback = function(data) {
    if (data.action == google.picker.Action.PICKED) {
        var fileId = data.docs[0].id;
        rtpg.realtimeLoader.redirectTo([fileId], rtpg.realtimeLoader.authorizer.userId);
    }
}

```

4.4. Control de usuarios

Para poder empezar una clase tenemos que tener una serie de usuarios conectados a la aplicación de tal forma pueda producirse interactividad entre ellos y tenga sentido. Por una parte uno de ellos que tenga el rol de “Docente” y el otro de ellos que tenga el rol de “Alumno. Pero para ello hemos también primero de haber invitado a las personas que queremos que estén con nosotros en la clase.

Si accedemos directamente la URL la aplicación nos da la posibilidad de (una vez ya autorizada), crear una nueva clase o cargar una previa. Cuando realizamos la primera opción lo que se produce es una inicialización de los modelos existentes dentro de la aplicación. Estos modelos son lo que se entiende como elementos en tiempo real. Todos los modelos tienen a su vez declaradas todas las funciones incluidas dentro y que se declaran en sus correspondientes archivos JavaScript. Para ello la aplicación realiza una función de inicialización para cada uno de ellos:

```

rtpg.initializeModel = function(model) {
  var l = rtpg.allDemos.length;
  for (var i = 0; i < l; i++) {
    var demo = rtpg.allDemos[i];
    demo.initializeModel(model);
  }
};

```

Dentro de la aplicación en este caso contamos con modelos por cada una de las librerías .js existentes (excepto el filepicker.js que es un script externo, y funciones.js que son las funciones auxiliares generadas para un correcto funcionamiento). Pero cada vez que accedemos a la aplicación esta debe comprobar cuál es nuestro rol dentro de la misma.

```

rtpg.usercontrol = function(fileID_aux){
  validar_usuario(fileID_aux);
}

```

Hay que tener en cuenta que el rol que se ejecuta por defecto en el sistema es el de administrador (Docente) ya que se da la opción de crear o cargar clases según se desee, con el rol de alumno esto no es posible. Por lo tanto además de una manera inicial en la aplicación, podemos tener un rol de administrador si:

- Hemos entrado a una aplicación y creado una clase nueva.
- Hemos cargado una clase creada previamente por el usuario en cuestión.

En el caso de que así sea la aplicación realizará una serie de funciones (además de las que realiza por defecto con la carga de datos refrescando a su vez la página) para un correcto funcionamiento del sistema en función del usuario que somos. De igual manera ocurrirá si somos un “Alumno”:

- Hemos cargado una clase a la cual hemos sido invitados.

En el caso carguemos un archivo (clase) el sistema ejecutará una función de validación que nos devolverá cual es nuestra autorización y en función de ello refrescaremos de nuevo la página, visualizando la información de manera distinta si somos docente (administrador), o si somos alumno.

```

function validar_usuario(fileId) {
  var request = gapi.client.drive.files.get({
    'fileId': fileId
  });

  request.execute(function(resp) {

    try{
      var Admin = resp.ownerNames;
      var title = resp.title;
    }
  });
}

```



Aplicación para clase interactiva para tablets ANDROID

```
        catch(e){
    }

    var collaborator = rtpg.getMe();

    if(collaborator.displayName == Admin){
        control_admin();
    }
    else{
        control_usuario(Admin,title);
    }
    });
}
```

UNIVERSITAT POLITÈCNICA DE VALÈNCIA UPV E-Class

Panel de Administrador Para poder utilizar la aplicación primero debes [Autorizarla](#)

Clase
NUEVA Cargar
✓ Clase cargada correctamente
L18.2

Alumnos
Invitar: Agregar

Asistentes
Francisco José Guerra
Pepe Guerra

Opciones
 Habilitar envío de preguntas
 Habilitar resolución ejercicios
 Guardar Resultados
 Deshabilitar interactividad
✓ La aplicación esta lista!

Panel de Clase

Transparencias
Abrir Transparencias

Pizarra Virtual
Preguntas Respuestas

Problemas
Escribe aquí la pregunta deseas enviar
Clear the map Enviar Limpiar

Cuestiones
Escribe aquí la cuestión deseas enviar
A:
B:
C:
D:
Enviar Limpiar

Consultas
Pepe Guerra Pepe Guerra

UNA SOLA
Escribe aquí tu respuesta
Limpiar Enviar

Figura 29.- Interfaz docente

Panel de Administrador

 Para poder utilizar la aplicación primero debes [Autorizarla](#)

 Datos Clase: **L18.2**

Preguntas respondidas:

Panel de Clase

Transparencias

Abrir Transparencias

Pizarra Virtual

Pregunta001

Pregunta002

Y esto?

Escribe aqui tu respuesta

Enviar

Limpiar

Consultar

Respuestas

Escribir consulta

Escribe aqui la consulta deseas enviar

Clear the map

Enviar

Limpiar

Figura 30.- Interfaz alumno

Esta diferenciación de componentes de la interfaz, se produce tras la validación del usuario. Como se ve en el código, en el caso que nuestro rol sea de administrador, se ejecutará la función *control_admin()* mientras que si somos alumno será la función *control_usuario()* pasando también el parámetro *title* que se corresponde con el nombre de la clase y que aparece en el panel de clase del Alumno. Estas dos funciones lo que hacen es que en cada caso, ejecutan las funciones necesarias para en primer lugar cargar correctamente todos los elementos de los que está compuesta la página, como por ejemplo que los botones estén correctamente asignados a sus identificados, y que estos a su vez tengan la función que se ejecutará al pulsar sobre ellos de manera correcta con los parámetros correctos. Estos valores en mucho casos son dinámicos y

dependen de los valores que ya estén inicializados debido a una posible clase anterior (si estamos cargando una clase) o que son distintos si somos un docente a si somos un alumno.

```
//CONTROL DE USUARIO
function control_usuario(admin,title) {

    $("#panel").load( "html/user.html" );
    $("#contenedorcentral").load( "html/clase_user.html" );
    $(document).ready(function(){
        setTimeout(function(){cargar_panel_clase(title);},600);
        setTimeout(function(){cargar_preguntas_alumno();},600);
            setTimeout(function(){rtpg.map.updateUiAlumno();},600);
            setTimeout(function(){rtpg.map.updateDonAlumno();},600);
            setTimeout(function(){rtpg.map.connectUiAlumno();},600);

            setTimeout(function(){rtpg.map_consultas.updateDonAlumno();},600);
            setTimeout(function(){rtpg.map_consultas.updateUiAlumno();},600);
            setTimeout(function(){rtpg.map_consultas.connectUi();},600);

    });
}
```

```
//CONTROL DE ADMINISTRADOR
function control_admin() {

    $(document).ready(function(){
        setTimeout(function(){rtpg.map_consultas.updateUi();},600);
        setTimeout(function(){rtpg.map_consultas.updateDonDocente();},600);
        setTimeout(function(){rtpg.map_consultas.connectUiDocente();},600);
    });
}
```

Las funciones `$("#panel").load("html/user.html");` y `$("#contenedorcentral").load("html/clase_user.html");` son las que tienen como objetivo modificar los contenidos HTML de la página, eliminando aquellos que no se utilizan en la parte de alumno y sustituyéndolos. Estos contenidos son cargados directamente desde dos archivos, `clase_user.html` es donde se encuentra el código perteneciente a la sección del panel de clase y que se coloca en sustitución del panel de administrador. Por otra parte `user.html` es donde se encuentra la parte central de la aplicación con las modificaciones HTML para la parte de transparencias y la parte de interactividad (Pizarra virtual). Por su parte también está la función `cargar_panel_clase(title);` que con la variable `title` indicada previamente, genera una línea en el panel del alumno que le indica en que clase se encuentra actualmente.

```
//Cargar panel de clase
function cargar_panel_clase(title){
    $("#titulo_asignatura").text(title);
}
```

Cada una de las otras funciones tiene también un propósito específico, pero dado que pertenecen a la sección de interactividad entre usuarios, las explicaremos en un punto posterior a este, de tal forma que aquí únicamente indicamos que se ejecutan siempre y cuando realizamos la comprobación del tipo de usuario.

4.5. Invitar alumno

Si como administrador, deseamos invitar a una nueva persona a unirse a la clase, entonces pulsaremos el botón de “Invitar”. Al hacerlo se ejecutará una breve función que es la que abre la ventana emergente y en la que podemos escoger a las personas que queremos invitar a participar, añadiendo su dirección de correo electrónico. Es importante tener en cuenta que para ejecutar la aplicación, la persona receptora de la invitación debe tener cuenta también en Google ya que de esta manera podrá dotar de permisos a la misma y ejecutarla.

```
rtpg.popupShare = function() {  
  var shareClient = new gapi.drive.share.ShareClient(rtpg.realTimeOptions.appId);  
  shareClient.setItemIds(rtclient.params['fileIds'].split(','));  
  shareClient.showSettingsDialog();  
  
}
```



Figura 31.- Configuración uso compartido

Un detalle a tener en cuenta es que la aplicación funciona mejor si se entra directamente a la dirección URL y desde ahí se accede a la clase. La invitación de correo todavía da ciertos problemas la re-dirección URI²⁸ puede dar problemas a la hora de acceder a la clase en algunas

²⁸ http://es.wikipedia.org/wiki/Identificador_de_recursos_uniforme



ocasiones. No hay que olvidar que esta tecnología todavía se encuentra en fase beta y por tanto es normal todavía encontrar algunos problemas puntuales.

Cuando un usuario está conectado a la clase al mismo tiempo que el administrador podemos obtener sus datos, ya que estos aparecerán dentro del *model* `rtpg.collaborators`. De esta forma es la que obtenemos los Asistentes a la clase, que podemos ver en el panel de administrador.

Para ello lo que hacemos es ejecutar una función (que siempre se ejecutará cada vez que se produzca un cambio en alguno de los modelos de tiempo real), y que devuelve los colaboradores Online que hay en este momento dentro del archivo. En nuestro caso lo que interesa principalmente es el nombre de dicho colaborador que se puede obtener a través del parámetro `.displayName`. También se podrían controlar todas las acciones que realiza el usuario en particular a través de la librería `rtpg.log.js` pero no parece necesario llegar a ello por lo cual solo nos interesa en este caso que se nos muestre cuando está el usuario, y cuando no por tanto dicha funcionalidad queda incluida dentro de la aplicación por si se deseará utilizar en posteriores modificaciones, pero no está introducida en este primer modelo.

```
rtpg.collaborators.updateUi = function() {
  var collaboratorsList = rtpg.realtimeDoc.getCollaborators();
  $(rtpg.collaborators.COLLABORATORS_SELECTOR).empty();
  for (var i = 0; i < collaboratorsList.length; i = i + 1) {
    var collaborator = collaboratorsList[i];
    //var imgSrc = collaborator.photoUrl == null ? 'images/anon.jpeg' :
collaborator.photoUrl;
    //var img = $('<img>').attr('src', imgSrc).attr('alt', collaborator.displayName).attr('title',
collaborator.displayName + (collaborator.isMe ? " (Me)" : ""));
    //img.css('background-color', collaborator.color);
    //$(rtpg.collaborators.COLLABORATORS_SELECTOR).append(img);

    var nombre = collaborator.displayName;

$(rtpg.collaborators.COLLABORATORS_SELECTOR).append("<span>" + nombre + "</span>
<br>");
  }
};
```

4.6. Transparencias

La sección de las transparencias se sitúa en lo que es el *leftcontainer* de la página. Esta sección no funciona de la misma manera que el resto de la aplicación ya que no utiliza los mismos parámetros y variables que se emplean para el resto de acciones.

```
<section id="AbrirDocumento" class="disabled">
  <div class="notesBox rp-greyRuledTop rp-greyRuledBottom">
    <h3>
      <span>Transparencias</span>
```



```

        </h3>
    </div>
    <div class="rp-section-content" style="line-height: 30px;">

        <div>
            <iframe height="240" width="400" id="result"
seamless='true' src=""></iframe>
        </div>

        <button id="abrir_transparencias" type="button" >Abrir
Transparencias</button>
    </div>
</section>

```

Este código lo que indica es que nos encontramos con una sección que cuenta con un componente `iframe`²⁹ que es donde se cargarán las transparencias y un botón de “Abrir transparencias”, el cual nos abrirá un dialogo de carga, similar al de carga de clase, pero que en este caso nos mostrará aquellos archivos de tipo presentación que tenemos almacenados en Google Drive. Puede darse el caso que estos archivos no los tengamos almacenados propiamente, si no que estemos invitados a su visualización. Esto plantea un método de trabajo por el cual, en primer lugar el docente carga las transparencias en su espacio de trabajo dentro de dicho servicio de Google Drive y posteriormente invita a los alumnos de su clase a tener acceso a dicha presentación (con opción de visualización sería suficiente) y luego los alumnos pueden cargar las transparencias dentro de la aplicación para así tener a la vez dicha visualización mientras pueden responder las preguntas que el profesor les lanza o plantear las suyas propias, todo desde una misma pantalla web.

Para que esto se realice, una vez pulsamos el botón de apertura de transparencias, se ejecutan las funciones de la librería `filepicker.js` que es la que se encarga del control y carga de archivos. En particular interesa el siguiente código:

```

    _pickerCallback: function(data) {
        var url = 'nothing';
        if (data[google.picker.Response.ACTION] ==
google.picker.Action.PICKED) {
            var doc = data[google.picker.Response.DOCUMENTS][0];
            url = doc[google.picker.Document.EMBEDDABLE_URL] //
doc[google.picker.Document.URL];
            //document.getElementById('result').src = url;
            document.getElementById('result_location').innerHTML = url;
        }
    },

```

Este código se ejecuta una vez el alumno ha seleccionado que presentación quiere visualizar en la aplicación y lo que hace en primer lugar es obtener el documento seleccionado dentro de la

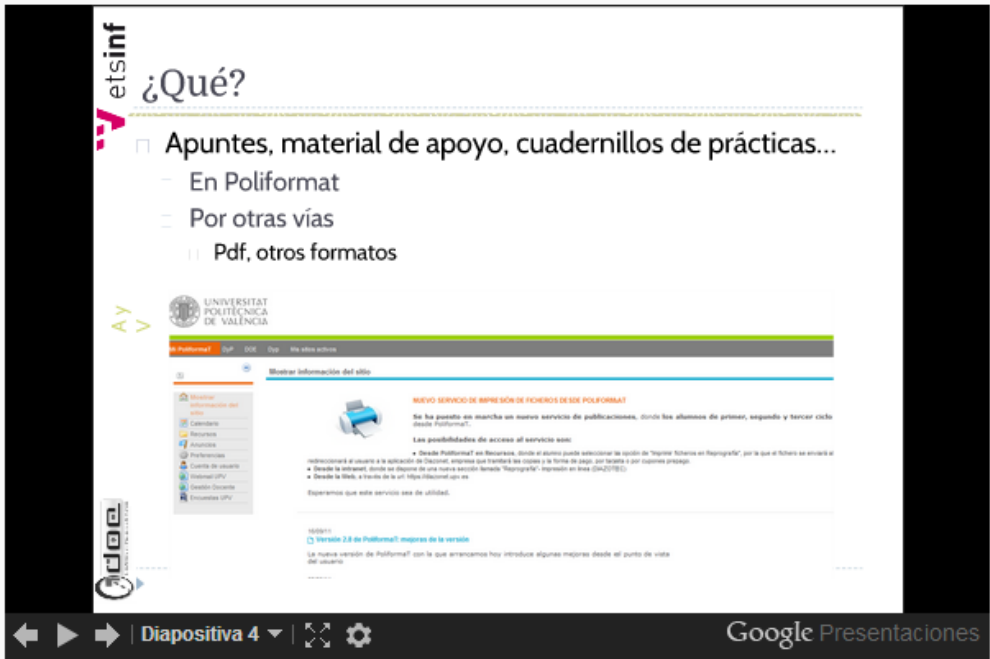
²⁹ http://www.w3schools.com/tags/tag_iframe.asp



variable *doc* y una vez almacenado obtiene la URL del archivo en cuestión que viene como parámetro del documento. Al tener la URL de la presentación y dado que en un iframe podemos cargar de manera embebida código html (incluso externo de la página), lo que hacemos es abrir dicha dirección dentro del iframe para visualizar el archivo, el cual no está alojado dentro de la aplicación (pues no hay necesidad de guardarlo) sino que realizamos una carga directa desde el lugar donde está almacenado y lo visualizamos.

Panel de Clase

Transparencias



etsinf

¿Qué?

- Apuntes, material de apoyo, cuadernillos de prácticas...
 - En Poliformat
 - Por otras vías
 - Pdf, otros formatos

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Mostrar información del sitio

NUOVO SERVIZIO DI IMPRESIONE DI FILE DA POLIFORMAT

Se ha puesto en marcha un nuevo servicio de publicaciones, desde los alumnos de primer, segundo y tercer ciclo desde Poliformat.

Las posibilidades de acceso al servicio son:

- Desde Poliformat en Recursos, desde el alumno puede seleccionar la opción de "Impedir ficheros en Poliformat", por lo que el fichero se envía al ordenador y el usuario de la aplicación de destino, descarga los ficheros de forma de cada vez cuando se quiere imprimir.
- Desde la página de inicio de una nueva sección llamada "Impresión" integrada en base (JAN2015).
- Desde la lista de ficheros de un sitio Poliformat que es.

Esperamos que este servicio sea de utilidad.

NOTAS

- Desde 2.0 de Poliformat mejoras de la versión

La nueva versión de Poliformat con la que arrancamos hoy introduce algunas mejoras desde el punto de vista del usuario.

Diapositiva 4

Google Presentaciones

Abrir Transparencias

Figura 32.- Transparencias

4.7. Pizarra virtual

Esta es probablemente la sección más importante de la aplicación, y la que más trabajo ha llevado en su desarrollo. En primer lugar vamos a explicar lo que sería el envío de preguntas y cuestiones desde el rol de docente al de alumno, y la posterior posible respuesta que este envíe de vuelta. También como se almacenan estas últimas y como son mostradas al docente. Básicamente este es el funcionamiento estándar que se seguirá también con las cuestiones, y posteriormente con las consultas, aunque con detalles que los diferencian.

```

<div id="problemas" class="rp-section-content">
  <h4>Problemas</h4>
  <textarea rows="8" cols="50" id="ProblemasValue" type="text"
placeholder="Escribe aquí la pregunta deseas enviar"></textarea>
  <button id="ProblemasClean" class="rp-button" value="">Limpiar</button>
  <button id="ProblemasPut" class="rp-button" value="">Enviar</button>
  <button id="demoMapClear" class="rp-button" style="margin-bottom:
10px;">Eliminar preguntas</button><br />
</div>

```

La parte del envío de preguntas (y de respuestas por parte del alumno que es muy similar con diferencias en los id) Se integra por un cuadro de texto (*textarea*) y una serie de botones, para limpiar el cuadro, para enviar la pregunta, y para eliminar todas las preguntas enviadas hasta el momento (este solo hábil en la función docente). El docente introduce la pregunta quiere realizar al alumno y pulsa el botón de “enviar” para que la pregunta sea enviada al alumno y le aparezca en tiempo real en su pantalla para poder resolverla. El alumno realizará un proceso similar, en vez del texto “problemas” encontrará la pregunta que le ha enviado el docente, y dentro del cuadro de texto deberá escribir su solución para el posterior envío.

```

<div class="content">
  <div id="respuestas_alumno" class="tabs"></div>
</div>

```

Por su parte habrá otra sección (pestaña) correspondiente a las respuestas enviadas por los alumnos donde se ira generando el código que nos presenten cada una de ellas mostradas dentro de una tabla para cada una de las preguntas con dos columnas, una para el nombre del alumno y otra para su respuesta.

Figura 33.- Envío preguntas – Docente

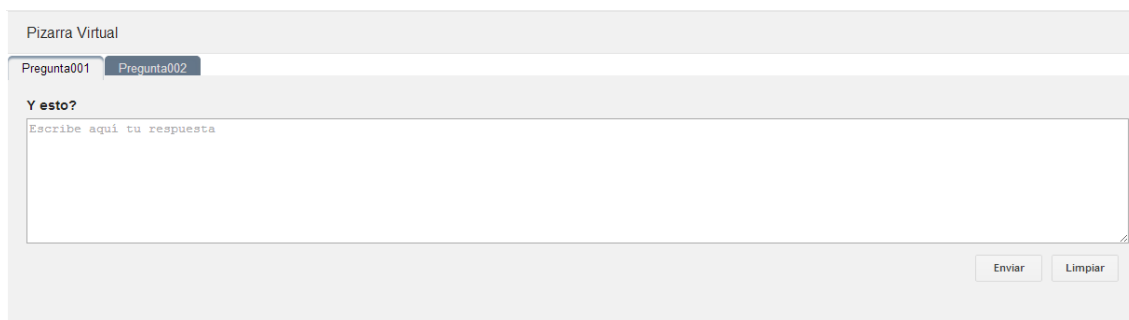


Figura 34.- Envío respuestas – Alumno

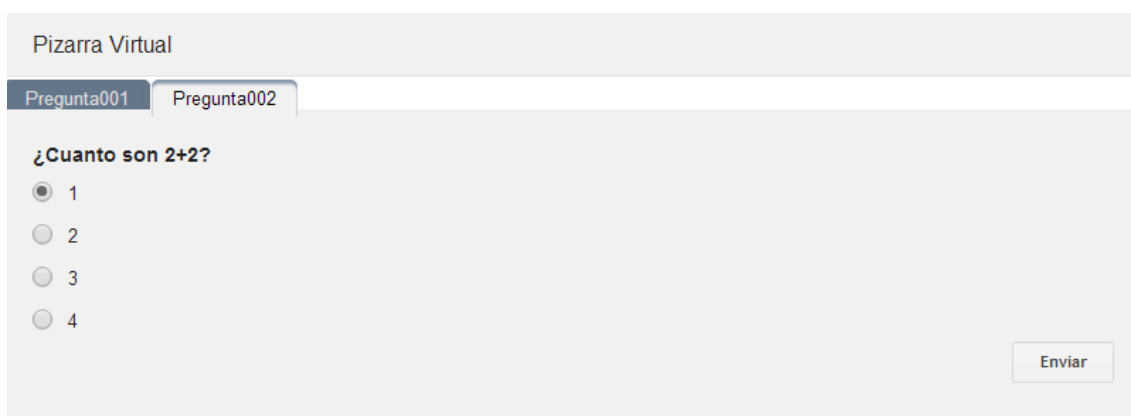


Figura 35.- Envío respuestas cuestiones – Alumno

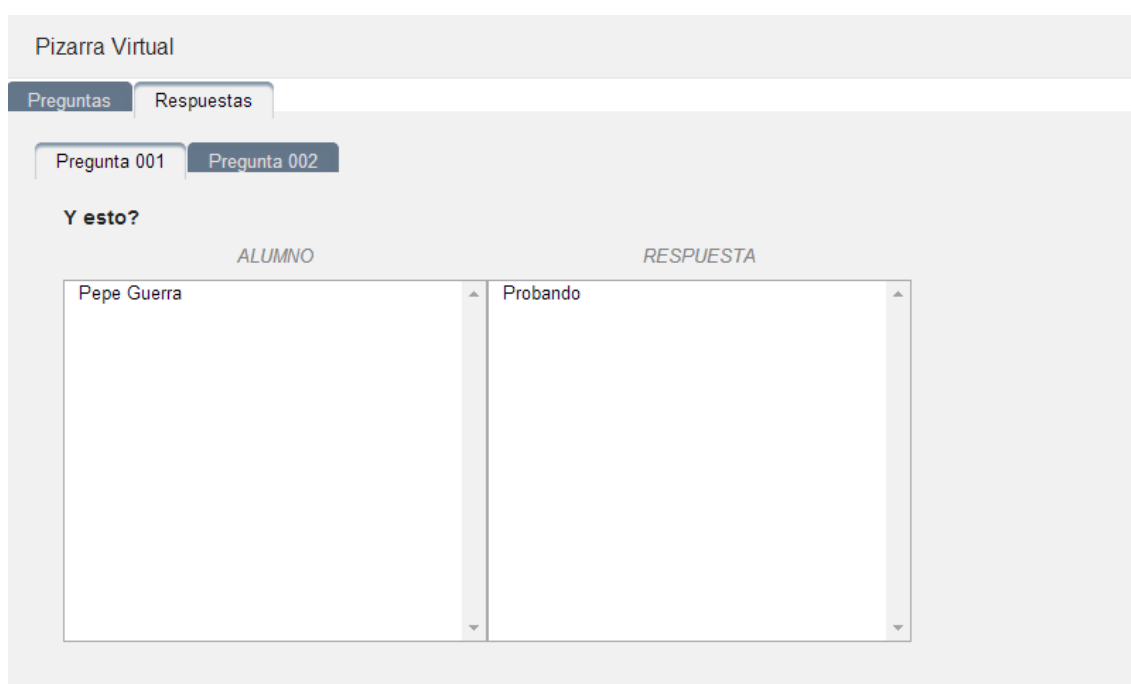


Figura 36.- Visualización respuestas – Docente

Pero antes de esto, hay que inicializar una serie de parámetros y valores que son los que permitirán la ejecución del código, el almacenamiento de datos, y la interactividad en tiempo real. Para esto seguimos una serie de pasos, que se repetirán para cada una de las secciones aquí encontradas, con algunas diferencias entre cada una de ellas. En este caso explicaremos paso por paso la secuencia de acciones desde que lanzamos la aplicación, hasta que llega la pregunta al alumno. También la carga de respuestas de las mismas para el docente, de existir.

1. Inicializar valores:

En primer lugar inicializamos la variable que almacenará los valores de las preguntas (y posteriores respuestas), como un modelo de tiempo real vacío con el nombre **map**. También lanzamos una función para que al inicio de la aplicación este modelo sea cargado. A su vez, le asignaremos un valor al campo del modelo, que es donde almacenaremos los datos y posteriormente recuperaremos.

```
rtpg.map = rtpg.map || {};  
rtpg.allDemos.push(rtpg.map);  
  
/**  
 * Realtime model's field name.  
 */  
rtpg.map.FIELD_NAME = 'demo_map';  
  
/**  
 * Realtime model's field.  
 */  
rtpg.map.field = null;  
rtpg.map.START_KEYS = ({});  
  
rtpg.map_consultas.initializeModel = function(model) {  
  
    var field = model.createMap(rtpg.map_consultas.START_KEYS);  
    for (var key in rtpg.map_consultas.START_KEYS) {  
        field.set(key, rtpg.map_consultas.START_KEYS[key]);  
    }  
    model.getRoot().set(rtpg.map_consultas.FIELD_NAME, field);  
}
```

Una vez esto procedemos a asignar variables a cada una de las secciones (por id) del código HTML que nos hará falta tener en cuenta en dicha sección ya sea para tomar los datos (como el *textarea*), o por su funcionalidad (botones). En este caso también están incluidos los que serían correspondientes a la sección de respuestas y también al sistema de pestañas dentro del mismo para diferenciar las respuestas en contenedores distintos y poder ver claramente lo que ha respondido cada alumno a cada respuesta por separado.



```

/**
 * DOM selector for the elements for Problemas.
 */
rtpg.map.MAP_KEYS_SELECTOR = '#ProblemasKeys';
rtpg.map.MAP_VALUES_SELECTOR = '#ProblemasValues';
rtpg.map.REMOVE_SELECTOR = '#demoMapRemove';
rtpg.map.PUT_SELECTOR = '#ProblemasPut';
rtpg.map.CLEAN_SELECTOR = '#ProblemasClean';
rtpg.map.PUT_VALUE_SELECTOR = '#ProblemasValue';

rtpg.map.MAP_KEYS_SELECTOR_NOMBRE = '#ProblemasNombre';
rtpg.map.MAP_VALUES_SELECTOR_RESPUESTA = '#ProblemasRespuesta';
rtpg.map.MAP_PES_ALUMNO = '#preguntas_alumno';
rtpg.map.MAP_RES_ALUMNO = '#respuestas_alumno';

```

En algunos casos de la aplicación, puede darse el caso que el código se genere dinámicamente mientras estamos trabajando dentro de la aplicación, es por ello que no siempre se puede realizar una asignación estática de las variables y asignarlas a elementos HTML ya que estos aún no han sido generados en el código y por tanto tendríamos problemas de ejecución con estos elementos. Es por ello que en algunos caso donde este código se genera dinámicamente, también hacemos lo mismo con la asignación de variables a los elementos por su id, de tal forma que cuando un nueva respuesta, o pregunta sea lanzada, se ejecuten las funciones necesarias para controlar todos los elementos de la aplicación. A su vez también, lo mismo ocurre con los procesos que ocurrirán cuando pulsemos los botones que se han generad dinámicamente, si no les indicamos la acción a realizar y los parámetros que deben enviar o recibir, la aplicación no funcionaria correctamente.

```

rtpg.map.updateDonDocente = function(){

    var keys = rtpg.map.field.keys();
    keys.sort();
    var l = keys.length;

    for (var i=0; i < l; i++) {
        var key = keys[i];
        if(key.length=="3"){
            eval("rtpg.map.PROBLEMAS_NOMBRE_"+key+"=
'#ProblemasNombre_"+key+"'");
            eval("rtpg.map.PROBLEMAS_RESPUESTA_"+key+"=
'#ProblemasRespuesta_"+key+"'");
        }
    }
}

rtpg.map.connectUiAlumno = function() {

```

```

var keys = rtpg.map.field.keys();
keys.sort();
var l = keys.length;

for (var i=0; i < l; i++) {
  var key = keys[i];
  if(key.length=="3"){

      var val = obtener_pregunta(rtpg.map.field.get(key));

      if(val[1]!=null){

          eval("${rtpg.map.PUT_RESPUESTA_CUESTIONARIO"+key+").click({key:""+key+""},
rtpg.map.onPutMapRespuestaCuestionario)");
              }
          else{

              eval("${rtpg.map.PUT_RESPUESTA_"+key+").click({key:""+key+""},rtpg.map.onPut
MapRespuesta)");

              eval("${rtpg.map.CLEAN_RESPUESTA_"+key+").click({key:""+key+""},rtpg.map.Clea
nRespuesta)");
          }
      }
  }
}

```

2. Carga y modelo de datos:

Una vez inicializado el modelo, cuando se carga el programa (o se produce alguna actualización sea cual sea el modelo) hay que comprobar los datos existentes el mismo y proceder a cargarlos. Lo primero será recuperar esos valores almacenados en campo previamente establecido y guardarlos dentro de la variable local del modelo, **field**

```

rtpg.map.loadField = function() {
  rtpg.map.field = rtpg.getField(rtpg.map.FIELD_NAME);
}

```

Si ya tenemos la variable con los datos, entonces podemos proceder a lanzar la función *updateUi*. Esta función se encarga de visualizar almacenados en la aplicación. Esta función esta replicada en varias más ya que la visualización de datos no se realiza de una misma forma para mostrar las preguntas lanzadas que para las respuestas por poner un ejemplo, es por ello ha sido mejor separarlas en varias replicadas, cada una correspondiente a la sección deseamos cargar. Este contenido en este caso lo que hacer es que visualiza las respuestas en los docentes.

```

rtpg.map.updateUi = function() {

  var key_aux = null;

```

```

var check = true;

$(rtpg.map.MAP_RES_ALUMNO).empty();

var keys = rtpg.map.field.keys();
keys.sort();
var l = keys.length;

for (var i=0; i < l; i++) {

    var key = keys[i];
    var val = obtener_pregunta(rtpg.map.field.get(key));

    if(key.length=="3"){

        eval("rtpg.map.PROBLEMAS_NOMBRE_"+key+"=
'#ProblemasNombre_'+key+'");
        eval("rtpg.map.PROBLEMAS_RESPUESTA_"+key+"=
'#ProblemasRespuesta_'+key+'");

        eval("$ (rtpg.map.PROBLEMAS_NOMBRE_"+key+").empty()");
        eval("$ (rtpg.map.PROBLEMAS_RESPUESTA_"+key+").empty()");

        if(i==0)check=true;
        else check=false;

        var html = generar_resultados(key,val[0],check);
        $(rtpg.map.MAP_RES_ALUMNO).append(html);

        key_aux = key;
    }

    if(key.length>"3"){
        var elem = key.split('-');

        var newOptionKey = $('<option>').val(elem[1]).text('\xa0\xa0' + elem[1]);
        var newOptionValue = $('<option>').val(val[0]).text('\xa0\xa0' + val[0]);

        eval("$ (rtpg.map.PROBLEMAS_NOMBRE_"+key_aux+").append(newOptionKey)");

        eval("$ (rtpg.map.PROBLEMAS_RESPUESTA_"+key_aux+").append(newOptionValue
)");
    }

}

l = l <= 1 ? 2 : l;

```



```

$(rtpg.map.MAP_RES_ALUMNO).attr('size', l);
}

```

La primera parte del código corresponde a la carga preliminar de datos y recuperación de la información, a partir de ahí entramos en el procesamiento del conjunto de valores *keys*.

Para entender el proceso lo primero a realizar explicar cómo están formados los elementos. Cada valor almacenado se almacena con un campo clave (*key*) y otro campo que sería el valor en sí (*value*). Cada pregunta está formada por un campo clave de tres dígitos (no se pueden repetir para distintas preguntas) y en el campo valor se encuentra la pregunta en sí. Si un alumno responde esta pregunta entonces se generará un nuevo campo dentro de la tabla de valores donde la *key* serán los mismos dígitos componían el campo clave de la pregunta, añadiendo en este caso el nombre del alumno que plantea la respuesta. De esta forma teniendo en cuenta que el campo clave se mantiene constante, podemos obtener las respuestas de todos los alumnos sacando para cada uno de ellos la segunda parte del campo clave que corresponde a su nombre.

Por ejemplo:

CLAVE	VALOR
001	¿Cuánto son 2+2?
001 – Pedro Juan	4
001 - Alfonso Ruz	4
002	¿Cuánto son 3+3?

De esta forma, aquellos elementos con *key* de longitud 3 es lo que corresponde a las preguntas, y aquellos con longitud mayor a 3, son respuestas. Si son las preguntas lo que queremos visualizar se ejecutará la función, para generar el HTML correspondiente:

```

var html = generar_resultados(key,val[0],check);

```

Esta función genera una nueva pestaña de resultados con su correspondiente tabla de los mismos, para la pregunta en cuestión. En cambio sí es una respuesta lo que haremos será procesarla y añadiremos valores a la tabla de dato correspondiente a la pregunta según su *key* numérica, donde el primero valor será el nombre del alumno y el segundo su respuesta:

```

var newOptionKey = $('<option>').val(elem[1]).text('\xa0\xa0' + elem[1]);
var newOptionValue = $('<option>').val(val[0]).text('\xa0\xa0' + val[0]);

```

La misma forma de trabajo (aunque con distintas funciones) se utiliza para las cuestiones, con la diferencia que para diferenciar entre una pregunta normal y una cuestión con opciones, estas se añaden también en el campo del valor en sí, unidas a la pregunta aunque separadas por una secuencia de signos para poder recuperarlas también y poder cargarlas al alumno. Por su parte en alumno al responderlas, elegiría una de las opciones cuyo valor sería el que sería la respuesta.



Por ejemplo:

CLAVE	VALOR
001	¿Cuánto son 2+2? <i>/**/**/**/1/**/2/**/3/**/4/**/</i>
001 – Pedro Juan	4
001 - Alfonso Ruz	4
002	¿Cuánto son 3+3? <i>/**/**/**/1/**/3/**/5/**/6/**/</i>

Por su parte las consultas tienen un funcionamiento muy parecido a las preguntas aunque se almacenan en un modelo distinto para que no se confundan ambas secciones. En este caso es el alumno el que lanza las preguntas al docente, por lo cual aquí es el docente el que genera respuestas. Además cada docente puede ver las preguntas de todos sus alumnos, pero estos solo pueden ver sus propias preguntas (y respuestas del docente). Aquí por tanto además de la formula realizada previamente del campo numérico y el nombre del alumno se añade el carácter **R** a la *key* cuando la pregunta generada por el alumno ha sido respondida, para marcar que es la respuesta de la misma. De esta forma el modelo sería:

CLAVE	VALOR
001 – Pedro Juan	¿Cómo está planteado el modelo?
001R – Pedro Juan	De manera paralela
002 – Alfonso Ruz	¿La pregunta 1 es modelo de examen?

En este caso siempre está presente el nombre del alumno en la *key* para que podamos enviarle la respuesta a él directamente, sabiendo así quien es el destinatario de la misma y no le aparezca a otro alumno distinto.

```

if(elem[0].length=="4" && elem[1]==obtener_alumno()){

    var val = rtpg.map_consultas.field.get(key);
    var aux = elem[0].substring(0, elem[0].length-1)+"-"+elem[1];

    var pregunta = rtpg.map_consultas.field.get(aux);

    var html = generar_respuesta_consultas(elem[0],pregunta,val,check);
    check=false;

    $(rtpg.map_consultas.RESPUESTAS_PROFESOR).append(html);
}
    
```

3. Introducir datos:

Si bien ya se ha comentado como se cargan los datos, ahora hay que ver las funciones que se ejecutan para almacenar los nuevos datos dentro del sistema. Para ello se hará uso de la función *onPutMap* y que al igual que antes tendrá varias réplicas en función de tipo de elemento (pregunta, respuesta, cuestión o consulta) se esté guardando.

```

rtpg.map.onPutMap = function(evt) {

    var keys = rtpg.map.field.keys();
    if(keys.length==0) var aux="1";
    else{
        keys.sort();
        keys.reverse();
        var aux = ((parseInt(keys[0]))+1).toString();
    }

    var key = add_zero(aux);
    var val = $(rtpg.map.PUT_VALUE_SELECTOR).val();

    rtpg.map.field.set(key, val);

    alert("Su pregunta ha sido enviada correctamente");
    $(rtpg.map.PUT_VALUE_SELECTOR).val("");

};

```

Esta función lo que hace es recuperar los valores almacenados en el campo de texto del código HTML donde se ha escrito, una vez recuperado genera la clave del valor, en función del tipo elemento se trata y también de las claves ya existentes en el modelo. Una vez generada se introduce en el modelo y se almacena. La principal diferencia entre los elementos y su generación es lo que se ha comentado previamente y que se resume de la siguiente forma:

ELEMENTO	VALORES
Pregunta	Key numérica + valor pregunta
Cuestión	Key numérica + valor pregunta y opciones
Respuesta pregunta / cuestión	Key numérica y nombre alumno – valor respuesta
Consulta	Key numérica y nombre alumno – valor consulta
Respuesta consulta	Key numérica(+R) y nombre alumno – valor respues consu.

4. Actualización datos:

Al inicio de la explicación de la implementación, se ha comentado sobre la diferenciación entre los distintos roles, lanzándose como docente el *control_admin* una vez la aplicación nos validaba como administrador y por su parte la función *control_usuario* si somos alumno. Cuando estas funciones se ejecutan, ordenan la ejecución de la inicialización y carga de datos correspondientes a cada rol.

Por una parte como administrador:

```

setTimeout(function(){rtpg.map_consultas.updateUi();},600);
setTimeout(function(){rtpg.map_consultas.updateDonDocente();},600);
setTimeout(function(){rtpg.map_consultas.connectUiDocente();},600);

```



Estas funciones recargan la sección de consultas dentro de la aplicación docente, ya que es la única cuyos valores no se carga de manera inicial cuando abrimos la página web de manera acorde al funcionamiento como docente y existen valores se generan dinámicamente y cuyo comportamiento debe ser controlado de la misma forma. Por otra parte, como alumnos:

```
setTimeout(function(){rtpg.map.updateUiAlumno();},600);
setTimeout(function(){rtpg.map.updateDonAlumno();},600);
setTimeout(function(){rtpg.map.connectUiAlumno();},600);

setTimeout(function(){rtpg.map_consultas.updateDonAlumno();},600);
setTimeout(function(){rtpg.map_consultas.updateUiAlumno();},600);
setTimeout(function(){rtpg.map_consultas.connectUi();},600);
```

Hemos de controlar los elementos pertenecientes a la sección de consultas igualmente, pero también a las preguntas que se nos cargan del sistema y que estaban almacenadas previamente en la clase. Esta es una carga inicial que solo se ejecutará en este momento de comprobación de roles. Finalmente indicar que al englobarla dentro de la función *setTimeout*, lo que hacemos es dotar de un tiempo mínimo de ejecución a cada una de las funciones para que no se ejecute sin haberse ejecutado correctamente la anterior y darle un tiempo suficiente a cada una para completarse de manera correcta.

Una vez estemos ya funcionando con la aplicación y se produzca alguna actualización o modificación en un modelo, entonces se ejecutarán las funciones *onRealtime* correspondientes a cada uno. Estas funciones son ejecutadas siempre que se produzca una recarga de datos, y es por ello que aquí a su vez introduciremos las llamadas a las funciones que hemos generado para el control de datos dinámico, y que también deben actualizarse.

```
rtpg.map.onRealtime = function(evt) {
  rtpg.map.updateUi();
  rtpg.map.updateUiAlumno();
  rtpg.map.updateDonAlumno();
  rtpg.map.connectUiAlumno();
  cargar_preguntas_alumno();
};

rtpg.map_consultas.onRealtime = function(evt) {
  rtpg.map_consultas.updateUi();
  rtpg.map_consultas.updateUiAlumno();
  rtpg.map_consultas.updateDonDocente();
  rtpg.map_consultas.connectUiDocente();
};
```

De esta manera se cierra el bucle en lo que respecta al ciclo de la aplicación desde el inicio de la misma, la carga de valores, el almacenamiento de nuevos elementos y la actualización en tiempo real de todos los elementos existentes en la misma para una visualización en tiempo real del estado del modelo.

5. Conclusiones

Finalmente, podemos decir que la parte principal e imprescindible de la aplicación está terminada y en funcionamiento. Se ha conseguido por una parte tener un control de los usuarios que entran a la aplicación además de conseguir diferenciarlos y visualizar el contenido según su rol. Por otra parte se ha conseguido poner en marcha mecanismos interactivos para la comunicación online, en tiempo real y con una estructura definida, así como también otros mecanismos de soporte para la realización de la clase con mayor funcionalidad, como la visualización de transparencias y presentaciones a través de un repositorio online sin necesidad de almacenamiento interno extra dentro de la propia web.

Para poder llevar a cabo dicho proyecto y finalizarlo con éxito, el trabajo se ha estructurado de tal manera que primero se ha hablado de lo que es la educación, principalmente en los últimos años a partir de la aparición de la TIC. A continuación hemos estudiado una herramienta de docencia online de hace ya algunos años y que ha servido como base para la toma de información y funcionalidad de nuestro proyecto llamada Classroom Presenter 3. Posteriormente hemos visto las distintas soluciones que existen hoy en día para la docencia online, y los dos modelos de trabajo que presentan las empresas o proyectos actualmente, código abierto o empresas de software propio. Para finalizar hemos conocido un poco más lo que significa el término E-learning y que representa ese medio de aprendizaje actualmente, cuales son las ventajas y los inconvenientes que presenta y a que debe su auge este tipo de aprendizaje.

Es importante señalar que pese a que su funcionalidad estructural básica pueda estar finalizada, hay muchos puntos de la aplicación que han quedado bien encaminados para ser desarrollados e implementados en un futuro no muy lejano, tal y como se ha podido comprobar el apartado de especificaciones. Muchos de los puntos no ha sido posible su puesta en funcionamiento debido a la falta de tiempo y de recursos para su desarrollo ya que una puesta en marcha completa de la aplicación con todos los rasgos y funcionalidades diseñadas hubiera conllevado un trabajo mucho mayor al posible en estas condiciones, así como también un equipo técnico y humano mayor. Por otra parte también hay otros aspectos que no ha sido posible desarrollar por falta de funcionalidades o recursos al respecto, en algunos casos estos recursos estaban fuera del alcance por motivos económicos en cuanto a su uso, y en otros porque este tipo de tecnología todavía está en desarrollo.

Un aspecto destacable de la aplicación resultado es que su diseño puede ser editado con facilidad al estar generado en lenguaje HTML y CSS, muy simple y conocido lo cual permite que otras personas con conocimientos en el puedan trabajarlo y dotarle de nuevos recursos o un nuevo diseño más acorde a futuras necesidades como podría ser interfaces más pequeñas o elementos de usabilidad mobile.

Durante todo el trabajo de la aplicación ha sido fundamental la información y documentación obtenida a través de las páginas de ayuda y para desarrolladores de Google, donde ha sido posible extraer gran parte de los recursos necesarios para el desarrollo de la aplicación y donde he podido encontrar las soluciones a los diversos problemas que han ido surgiendo durante su desarrollo. En concreto ha sido imprescindible un conocimiento y un



estudio previo de la herramienta Google Drive Realtime API, así como la puesta en funcionamiento de varios ejemplos y pruebas previas al desarrollo del proyecto para ser conocedor de sus posibilidades y de lo que podía extraer de dicha herramienta.

En un primer momento la programación resulto algo complicada debido a la multitud de elementos y funciones que se ejecutaban, agravado a su vez porque en muchos casos se producían cada vez que ciertas acciones se realizaban en la aplicación lo que cual generaba multitud de funciones en ejecución y elementos a controlar, incluso con la página ya cargada. Pero una vez entendido el patrón de funcionamiento, y aislando los elementos que se debían utilizar de aquellos que no era necesario, y de conocer cómo se trabajan los datos a nivel interno para poder utilizarlos posteriormente a nivel externo, el trabajo ha podido ser más constante, siempre con unas metas y unos objetivos precisos en cada etapa.

Puedo decir que mi desarrollo personal y conocimientos adquiridos han sido muy productivos y que al final de la misma he podido valorar gratamente lo que he adquirido y obtener la sensación de saber los pasos necesarios para poder continuar con esta aplicación con un mayor y completo desarrollo teniendo las ideas claras así como también la estructura y método de trabajo que sería necesario llevar a cabo para sacarlo adelante. He superado problemas iniciales que parecían de gran complejidad gracias a un conocimiento profundo de los recursos de que disponía y a constantes pruebas y errores en su puesta en marcha. Creo que ha sido una experiencia muy satisfactoria que me ha enseñado muchas posibilidades y conocimientos que podrían dar lugar no solo a este proyecto sino a otros nuevos con funcionalidades, aplicaciones y conocimientos que he adquirido desarrollando esta aplicación y los cuales estoy gratamente satisfecho haber podido llevar a cabo.

Sin otra cosa más que comentar, espero haber sido capaz de cumplir los objetivos planteados inicialmente, de una manera actual y conforme a los recursos más actuales existentes hoy en día que posee mucho potencial y que deseo sea de agrado de aquellos que tengan la oportunidad de utilizar, así como también agradecer a las personas han colaborado en el su trabajo y su dedicación, sin la cual no habría sido posible dicho proyecto.

6. Referencias

1. Google. *Google Drive Realtime API Quickstart*.
< <https://developers.google.com/drive/realtime/realtime-quickstart>> [Consulta: 07 de Julio de 2014]
2. University of Washington. *UW Classroom Presenter*.
< <http://classroompresenter.cs.washington.edu/>> [Consulta: 07 de Julio de 2014]
3. *Miriada*^x
< <https://www.miriadax.net/> [Consulta: 08 de Julio de 2014]
4. *Los primeros rastros de educación en Mesopotamia*.
<http://www.ehowenespanol.com/primeros-rastros-educacion-mesopotamia-info_288999/> [Consulta: 09 de Julio de 2014]
5. Im conciencia blog. *La educación en el antiguo egipto*
<<http://www.iccc.es/2008/01/la-educacion-en-el-antiguo-egipto/>> [Consulta: 09 de Julio de 2014]
6. Wikipedia. *Historia de la función docente*.
<http://es.wikipedia.org/wiki/Historia_de_la_funcion_docente> [Consulta: 09 de Julio de 2014]
7. *Educación romana*
<<http://www.profesorenlinea.cl/universalhistoria/Roma/RomaEducacion.htm>> [Consulta: 10 de Julio de 2014]
8. *Nuevas tecnologías de la información: Herramientas para la educación*
<http://campus.usal.es/~teoriaeducacion/rev_numero_06/n6_art_bustillo.htm >[Consulta: 10 de Julio de 2014]
9. Skype. < <http://www.skype.com/es/>>
10. “Google Drive SDK: Writing your first Realtime API app” Youtube
< <https://www.youtube.com/watch?v=Mlaov3J4hDA>> [Consulta: 10 de Julio de 2014]
11. Dropbox. < <https://www.dropbox.com/>>
12. Sakai < <http://sakaiproject.org/>>
13. Portaltic. *El 90% de profesores y estudiantes universitarios utilizan plataformas de docencia virtual*. < <http://www.europapress.es/portaltic/movilidad/software/noticia-90-profesores-estudiante-universitarios-utilizan-plataformas-docencia-virtual-20111006161958.html>> [Consulta: 16 de Julio de 2014]
14. Moodle < <https://moodle.org/?lang=en>>
15. 20 minutos. *Catorce de las 20 universidades más prestigiosas del mundo utilizan Sakai, una herramienta de E-Learning*.
<<http://www.20minutos.es/noticia/1104579/0/#xtor=AD-15&xts=467263>> [Consulta: 21 de Julio de 2014]
16. Formación en red. < <http://formacionprofesorado.educacion.es/index.php/es/materiales/270-moodle-plataforma-de-aprendizaje?showall=1>>
17. Massaber < <http://www.massaber.es/contenido/>>
18. UNED
< http://portal.uned.es/portal/page?_pageid=93,1&_dad=portal&_schema=PORTAL>
19. How the MoSCoW Method Helps You Prioritize Goals < <http://www.bizengine.com/moscow-method/>> [Consulta: 26 de Julio de 2014]

20. Poliformat < <https://poliformat.upv.es/>>
21. Pencil < <http://pencil.evolus.vn/>>
22. Github < <https://github.com>>
23. Google App Engine < <https://appengine.google.com>>
24. JavaScript < <http://www.w3schools.com/js/default.asp> >
25. HTML < http://www.w3schools.com/html/html_intro.asp>
26. CSS < http://www.w3schools.com/html/html_intro.asp>
27. Python < <https://www.python.org/> >
28. URI < http://es.wikipedia.org/wiki/Identificador_de_recurso_uniforme>
29. Tag iframe < http://www.w3schools.com/tags/tag_iframe.asp>