

Gestión del proyecto, diseño e implementación de una aplicación educativa usando servicios

Marius Constantin Iacob

TRABAJO FINAL DE GRADO

GRADO EN INGENIERÍA TELEMÁTICA

ESCUELA SUPERIOR POLITÈCNICA UPF

2019

DIRECTOR DEL TRABAJO

ÀNGEL DÍAZ LEAL



Universitat
Pompeu Fabra
Barcelona

Escola Superior
Politécnica

AGRADECIMIENTOS

Quiero agradecer el apoyo recibido por parte de Laura que sin ella este sueño no habría sido posible y especialmente a mi familia que siempre ha estado a mi lado.

También quiero agradecer a mi tutor Àngel Diaz que me ha guiado a lo largo de este trabajo.

RESUMEN

Este proyecto final de grado consiste en el estudio e implementación de una aplicación multiplataforma que se enfoca al ámbito educativo del entorno rural. Para llevar a cabo este trabajo, se utilizan diferentes servicios que permiten gestionar la autenticación, disposición de contenido video y la comunicación mediante un chat.

El proyecto es muy interdisciplinar puesto que se ha estudiado y desarrollado como una solución escalable y altamente disponible que va acompañada de un sistema automatizado de entrega continua de software que permite la evolución de esta solución. Además, se aprovecha en todo momento el uso de servicios gratuitos y de pago por uso que ofrecen diferentes proveedores de servicios *Cloud*.

Gran parte del proyecto gira en torno al análisis, diseño, desarrollo e implementación de las necesidades captadas teniendo en cuenta distintas herramientas que facilitan la transmisión del conocimiento. Esto junto con el aprendizaje del *framework* Angular y la implementación de una herramienta de automatización de despliegue de software ha ocupado gran parte del tiempo empleado en este trabajo final de grado.

RESUM

Aquest projecte final de grau consisteix en l'estudi i implementació d'una aplicació multiplataforma que és enfocada a l'àmbit educatiu de l'entorn rural. Per dur a terme aquest treball s'utilitzen diferents serveis que permeten gestionar l'autenticació, disposició de contingut vídeo i la comunicació mitjançant un xat.

El projecte és molt interdisciplinari ja que s'ha estudiat i desenvolupat com una solució escalable i altament disponible que va acompanyada d'un sistema automatitzat de lliurament continuu de software que permet l'evolució d'aquesta solució. A més, s'aprofita en tot moment l'ús de serveis gratuïts i de pagament per ús que ofereixen diferents proveïdors de serveis Cloud.

Gran part del projecte gira entorn de l'anàlisi, disseny, desenvolupament i implementació de les necessitats captades tenint en compte diferents eines que faciliten la transmissió del coneixement. Això juntament amb l'aprenentatge del *framework* Angular i la implementació d'una eina d'automatització de desplegament de programari ha ocupat gran part del temps emprat en aquest treball final de grau.

ABSTRACT

This final degree project consists in the study and implementation of a multiplatform application that is focuses on the educational field of the rural environment. To carry out this work, different services are used to manage authentication, provision of video content and communication through a chat.

The project is very interdisciplinary since it has been studied and developed as a scalable and highly available solution that is accompanied by an automated system of continuous software

delivery that allows the evolution of this solution. In addition, the use of free and pay-per-use services offered by different Cloud service providers is exploited in different ways.

Much of the project revolves around the analysis, design, development and implementation of the needs captured taking into account different tools that facilitate the transmission of knowledge. This, together with the learning of the Angular framework and the implementation of a software deployment automation tool, has taken up a large part of the time spent in this final degree project.

ÍNDICE

AGRADECIMIENTOS	III
RESUMEN.....	V
LISTA DE FIGURAS	IX
LISTA DE IMAGENES	XI
LISTA DE TABLAS	XIII
1. INTRODUCCIÓN	1
1.1. MOTIVACIÓN	2
1.2. IDEA Y OBJETIVOS	3
2. ESTADO DEL ARTE	7
2.1. ESTUDIO SOBRE LAS TECNOLOGÍAS DE LA INFORMACIÓN	7
2.2. LAS APLICACIONES EDUCATIVAS.....	11
a) Moodle.....	12
b) Canvas LMS.....	13
c) Chamilo LMS.....	13
d) Sakai.....	14
e) LearnPress - WordPress LMS.....	14
f) Blackboard LMS.....	14
g) eDucativa	15
h) OpenText FirstClass.....	15
2.3. ELECCIÓN DEL FRAMEWORK PARA EL DESARROLLO	15
3. ANÁLISIS Y DISEÑO DE SOFTWARE	17
3.1. REQUISITOS DE LA APLICACIÓN	17
3.2. ARQUITECTURA.....	18
a) Componente de autenticación.....	19
b) Componente del perfil de usuario.....	19
c) Componente de búsqueda.....	19
d) Componente de contenido video.....	20
e) Componente de administrar videos.....	20
f) Componente del chat.....	20
g) Componente administrar usuarios.....	20
h) Componente de preguntas frecuentes.....	20
i) Componente formulario de contacto	20
3.3. DESCRIPCIÓN FUNCIONAL	20
a) Lista de preguntas frecuentes sobre el funcionamiento.....	21
b) Formulario de contacto	21
c) Lista de videos.....	21
d) Perfil de usuario	22
e) Administrar roles de usuarios.....	22
3.4. CASOS DE USO	22
3.5. MOCK-UP	30
a) Registro en la aplicación.....	30
b) Iniciar sesión en la aplicación.....	31
c) Preguntas frecuentes.....	31
d) Formulario de Contacto	32
e) Lista de videos.....	33
f) Perfil de usuario	34
g) Ver video	34
h) Añadir video.....	35
i) Editar video.....	35
j) Eliminar video.....	36
k) Administrar usuarios.....	36
3.6. BACK-END DE LA APLICACIÓN	37

3.7.	ESTIMACIÓN DE COSTES.....	39
3.8.	TRATAMIENTO DE LOS DATOS PERSONALES DE LOS USUARIOS LOPD.....	44
4.	IMPLEMENTACIÓN	45
4.1.	CONFIGURACIÓN INICIAL	45
a)	Entorno de desarrollo utilizado.....	45
b)	Estructura de un proyecto en Angular.....	45
4.2.	PATRONES DE ARQUITECTURA DE SOFTWARE	47
a)	El MVVM en Angular.....	49
4.3.	CONFIGURACIÓN DE LOS SERVICIOS DE BACK-END	51
a)	YouTube Data API.....	51
b)	FireBase.....	54
4.4.	DESPLIEGUE DE LA APLICACIÓN.....	55
a)	Gestión del dominio web.....	55
b)	Alojamiento web.....	56
c)	Configuración de la CDN y certificado SSL	58
4.5.	CONFIGURACIÓN DE LA HERRAMIENTA DE AUTOMATIZACIÓN.....	62
a)	Despliegue de una instancia de computo en AWS.....	62
b)	Instalación y configuración básica del Jenkins	63
c)	Proceso de automatización implementado	64
5.	ORGANIZACIÓN DEL TRABAJO	67
6.	TRABAJO FUTURO	69
7.	CONCLUSIONES.....	71
8.	GLOSARIO.....	73
9.	BIBLIOGRAFIA	77
10.	ANEXOS.....	81

LISTA DE FIGURAS

FIG. 1: ESQUEMA BÁSICO DEL ENTORNO PRODUCTIVO	4
FIG. 2: ESQUEMA BÁSICO DEL ENTORNO DE DESARROLLO	5
FIG. 3: USO DEL MÓVIL SEGÚN LA EDAD	11
FIG. 4: DIAGRAMA DE BLOQUES DE ARQUITECTURA DE LA APLICACIÓN	19
FIG. 5: DIAGRAMA CASOS DE USO - ACCESO A LA APLICACIÓN	22
FIG. 6: DIAGRAMA CASOS DE USO - LISTA DE VIDEOS.....	24
FIG. 7: DIAGRAMA CASOS DE USO - ADMINISTRACIÓN USUARIOS Y PREGUNTAS FRECUENTES.....	28
FIG. 8: <i>MOCK-UP</i> - REGISTRO	30
FIG. 9: <i>MOCK-UP</i> - INICIO SESIÓN	31
FIG. 10: <i>MOCK-UP</i> - PREGUNTAS FRECUENTES – VISTA ALUMNOS Y USUARIOS NO REGISTRADOS	31
FIG. 11: <i>MOCK-UP</i> - PREGUNTAS FRECUENTES – VISTA ADMINISTRADOR Y PROFESORES	32
FIG. 12: <i>MOCK-UP</i> - FORMULARIO DE CONTACTO	32
FIG. 13: <i>MOCK-UP</i> - LISTA DE VIDEOS – VISTA DE LOS ALUMNOS.....	33
FIG. 14: <i>MOCK-UP</i> - LISTA DE VIDEOS – VISTA DE LOS PROFESORES	33
FIG. 15: <i>MOCK-UP</i> - LISTA DE VIDEOS – VISTA DE LOS ADMINISTRADORES.....	34
FIG. 16: <i>MOCK-UP</i> - PERFIL DE USUARIO	34
FIG. 17: <i>MOCK-UP</i> - VER VIDEO Y CHAT	35
FIG. 18: <i>MOCK-UP</i> - AÑADIR VIDEO.....	35
FIG. 19: <i>MOCK-UP</i> - EDITAR VIDEO	36
FIG. 20: <i>MOCK-UP</i> - ELIMINAR VIDEO	36
FIG. 21: <i>MOCK-UP</i> - ADMINISTRAR USUARIOS.....	37
FIG. 22: ARQUITECTURA COMUNICACIONES CON FIREBASE.....	37
FIG. 23: ARQUITECTURA DE LOS SERVICIOS UTILIZADOS	39

LISTA DE IMAGENES

IMAGEN 1: USO DEL MÓVIL, INTERNET Y REDES SOCIALES [5]	7
IMAGEN 2: CRECIMIENTO ANUAL DIGITAL [5]	8
IMAGEN 3: TIEMPO DEDICADO A INTERNET [5]	8
IMAGEN 4: VELOCIDAD DE CONEXIÓN A INTERNET [5]	9
IMAGEN 5: ACTIVIDADES DE <i>STREAMING</i> DE CONTENIDO [5]	9
IMAGEN 6: LAS REDES SOCIALES MÁS ACTIVAS [5]	10
IMAGEN 7: USO DE INTERNET O MÓVIL DE LOS MENORES DE 10 A 15 AÑOS [6]	10
IMAGEN 8: LOGO MOODLE [9]	12
IMAGEN 9: LOGO CANVAS [10]	13
IMAGEN 10: LOGO CHAMILO [12]	13
IMAGEN 11: LOGO SAKAI [13]	14
IMAGEN 12: LOGO LEARNPRESS [14]	14
IMAGEN 13: LOGO BLACKBOARD [15]	14
IMAGEN 14: LOGO EDUCATIVA [16]	15
IMAGEN 15: LOGO OPENTEXT [17]	15
IMAGEN 16: SERVICIOS OFRECIDOS POR FIREBASE [18]	38
IMAGEN 17: FACTURA REGISTRO DOMINIO WEB [ANEXO 1]	41
IMAGEN 18: ESTRUCTURA DE DIRECTORIOS Y FICHEROS DEL PROYECTO	46
IMAGEN 19: ESTRUCTURA DE DIRECTORIOS Y FICHEROS DE LA CARPETA <i>SRC</i> DEL PROYECTO	47
IMAGEN 20: LISTADO DE DOMINIOS REGISTRADOS	55
IMAGEN 21: PANEL DE ADMINISTRACIÓN DE DOMINIOS PIENSA SOLUTIONS	55
IMAGEN 22: PANEL DE ADMINISTRACIÓN ROUTE 53 AWS	56
IMAGEN 23: PANEL DE ADMINISTRACIÓN DNS PIENSA SOLUTIONS	56
IMAGEN 24: PROCESO CREACIÓN <i>BUCKET</i> AWS S3	57
IMAGEN 25: PROCESO CONFIGURACIÓN ALOJAMIENTO WEB ESTÁTICO EN AWS S3	57
IMAGEN 26: PROCESO CONFIGURACIÓN DE LA POLÍTICA DE ACCESO AL <i>BUCKET</i>	58
IMAGEN 27: CERTIFICADO SSL PARA LA APLICACIÓN WEB	59
IMAGEN 28: CONFIGURACIÓN DEL ORIGEN DEL CONTENIDO PARA LA CDN	59
IMAGEN 29: CONFIGURACIÓN DEL COMPORTAMIENTO DE LA CACHE PARA LA CDN	60
IMAGEN 30: CONFIGURACIÓN DE LA DISTRIBUCIÓN CDN	61
IMAGEN 31: GRUPO DE SEGURIDAD ASOCIADO A LA INSTANCIA JENKINS	62
IMAGEN 32: ROL IAM PARA LA INSTANCIA JENKINS	63
IMAGEN 33: REPRESENTACIÓN DEL PIPELINE DE LA RAMA <i>DEV</i> DEL REPOSITORIO	64
IMAGEN 34: REPRESENTACIÓN DEL PIPELINE DE LA RAMA <i>MASTER</i> DEL REPOSITORIO	65
IMAGEN 35: DIAGRAMA DE GANTT [ANEXO 2]	67
IMAGEN 36: ORGANIZACIÓN DE TRABAJO CON TRELLO	68

LISTA DE TABLAS

TABLA 1: CASO DE USO - PREGUNTAS FRECUENTES.....	23
TABLA 2: CASO DE USO - FORMULARIO DE CONTACTO	23
TABLA 3: CASO DE USO - CREAR UNA CUENTA.....	23
TABLA 4: CASO DE USO – AUTENTICARSE.....	24
TABLA 5: CASO DE USO - CERRAR SESIÓN.....	24
TABLA 6: CASO DE USO - SINCRONIZAR DATOS	25
TABLA 7: CASO DE USO - VER LISTA VIDEOS.....	25
TABLA 8: CASO DE USO - VER VIDEO	25
TABLA 9: CASO DE USO - ENVIAR MENSAJE.....	26
TABLA 10: CASO DE USO – BUSCAR VIDEO.....	26
TABLA 11: CASO DE USO - EDITAR VIDEO	26
TABLA 12: CASO DE USO – SUBIR VIDEO	27
TABLA 13: CASO DE USO – ELIMINAR VIDEO	27
TABLA 14: CASO DE USO – ADMINISTRAR USUARIOS.....	28
TABLA 15: CASO DE USO – BUSCAR USUARIO.....	29
TABLA 16: CASO DE USO – EDITAR PREGUNTAS FRECUENTES	29
TABLA 17: CASO DE USO – AÑADIR PREGUNTAS FRECUENTES.....	29
TABLA 18: CASO DE USO – ELIMINAR PREGUNTAS FRECUENTES	30
TABLA 19: COSTES ASOCIADOS AL DESPLIEGUE DE LA APLICACIÓN [ANEXO 1]	44

1. INTRODUCCIÓN

El presente Trabajo de Final de Grado presenta y analiza los cambios que ha sufrido el sistema educativo en las últimas décadas. El número de alumnos en las escuelas del entorno rural han disminuido tanto que la administración ha desarrollado planes estratégicos para mejorar el sistema educativo del entorno rural. El principal objetivo de estos planes se enfoca en guiar a los profesores como hacer frente a esta situación. Hoy en día, existen diferentes metodologías que se aplican a la hora de impartir materias a alumnos de diferentes cursos al mismo tiempo.

La Ley 14/1970, General de Educación y Financiamento de la Reforma Educativa, define un modelo de organización escolar basado en centros escolares de 8 unidades como mínimo, una para cada uno de los cursos que consta dicho nivel. La aplicación de este modelo educativo contemplado en el año 1970 para zonas de poblamiento dispersa supuso un gran revuelo y como consecuencia el cierre de pequeñas escuelas. Las alternativas ofrecidas en aquel entonces se basaban en el transporte escolar de los alumnos. A pesar de esto muchas escuelas unitarias no pudieron adaptarse y siguieron funcionando como centros incompletos, careciendo de los recursos de profesorado o equipamientos mínimos para poder atender los objetivos propios de la Educación General Básica [1].

En el año 1986 se recoge por primera vez en el Real Decreto 2731/1986 [2] la posibilidad de constituir Colegios Rurales Agrupados (C.R.A.) de Educación General Básica. Se propone mediante este decreto una alternativa organizativa en la escuela rural para los centros comarcales incompletos sin la necesidad de que el alumnado tenga que desplazarse a diario. Esta nueva forma organizativa favorece la optimización de los recursos humanos y materiales. En cuanto a los cambios propuestos, estos han supuesto de forma directa un cambio de mentalidad en la comunidad educativa a la hora de enseñar. Se requiere de una mayor coordinación entre un elevado número de profesionales y esto no siempre es fácil de conseguir.

En los C.R.A. existen diferentes tipos de profesorado: maestros ordinarios, maestros itinerantes, especialistas en Lengua extranjera, Educación física, Pedagogía terapéutica, Música, Lenguaje, Audición y además también hay otros maestros compartidos con otros centros. De aquí se deduce que en ocasiones pueden producirse problemas organizativos sobre todo cuando se deben elaborar los horarios y la utilización de los recursos.

Además de los problemas organizativos encontrados en los C.R.A. en su inicio, se encuentra la dificultad para conseguir que el profesorado reflexione y coordine mejor la actividad docente. La mayoría estaban acostumbrados a trabajar de forma independiente, pero con la constitución de estos colegios se comienzan a plantear las dificultades encontradas en torno a la organización y gestión de los recursos.

En ciertas zonas, la constitución de los C.R.A. tuvo ciertas reticencias en los centros donde hasta entonces habían funcionado de forma independiente. Al final el resultado organizativo de los C.R.A. fue asumido por una parte del profesorado, mientras que otros se mostraban críticos. La constitución de los C.R.A. apuesta por una nueva fórmula organizativa donde se pretende optimizar los distintos recursos manteniendo la estrecha relación con los principales actores involucrados: alumnos, padres y profesores [1].

En cuanto al entorno organizativo de un C.R.A., estos cuentan con unos órganos de gobiernos formados por un: Consejo escolar, Claustro de profesores, Equipo directivo; y unos órganos de coordinación formados por los Equipos de ciclo y la Comisión de coordinación pedagógica. La responsabilidad de un C.R.A. recae sobre el director como líder de la organización. Entre sus funcionalidades de director podrían destacar: la coordinación de las actividades, la gestión de los materiales, autorizar gastos, gestionar los pagos, elaborar con el quipo directivo los proyectos educativos, la programación general anual, favorecer la evaluación de todos los proyectos y actividades del centro.

Cabe destacar que el proceso de elaboración de proyectos educativos del centro, la elección del material necesario para asegurar la coherencia y continuidad durante el periodo de enseñanza obligatoria debe ser consensuada y asumida por la comunidad educativa. Se requiere de una dinámica donde el director debe ejercer como líder y hacer participe tanto a los alumnos, padres y profesores.

La idea principal de este proyecto nace a raíz de los problemas, organizativos presentados anteriormente. Por esta razón como alumno de Ingeniería Telemática nace la iniciativa de realizar como Trabajo de Final de Grado una aplicación web multiplataforma que sirva como herramienta de apoyo en la transmisión del conocimiento para ese grupo de organizaciones.

Esta aplicación pretende ayudar a solucionar problemas de movilidad y coordinación de forma puntual para toda aquella persona que tenga una dificultad de desplazamiento al centro educativo. En ningún momento se pretende cambiar, sustituir o cuestionar las actuales metodologías y normativas vigentes, pero si que podría presentarse como una posible solución alternativa que permita ayudar y mejorar el sistema educativo.

Con dicha aplicación web, se trata de llegar a las todas las plataformas, abstrayendo de esta forma la compatibilidad de los dispositivos. La aplicación se enfoca al sector educativo del ámbito rural. Esta aplicación aportara de forma fácil e intuitiva soporte para la consulta de contenido video y además un canal de comunicación entre alumnos y profesores. Finalmente se presentará un prototipo de aplicación compatible con las diferentes plataformas.

1.1. Motivación

Como se comento en el anterior apartado, el actual sistema educativo [3] contempla la posibilidad de constituir C.R.A. en la Educación General Básica para crear una escuela rural mas eficiente. Un C.R.A. básicamente es un centro educativo donde se imparten las áreas de Educación Infantil y Primaria.

Este tipo de centros educativos están presentes en las comunidades de: Aragón, Principado de Asturias, Andalucía, Galicia, Castilla-La Mancha, Castilla y León, Extremadura, Comunidad Valencia y la Comunidad de Madrid. En Catalunya las escuelas rurales se agrupan en Zonas Escolares Rurales (Z.E.R.) regulado por Decreto 195/1988. En Andalucía se encuentran bajo el nombre Colegio Publico Rural (C.P.R.) siendo un modelo de organización prácticamente idéntico a los C.R.A. mientras que en Canarias las encontramos bajo el nombre de Colegios de Escuelas Rurales (C.E.R.).

Todas estas agrupaciones tienen un único denominador común, el colegio rural o colegio unitario. El colegio unitario es un centro con un numero reducido de alumnos donde uno o dos

maestros atienden alumnos de diferentes edades y grados. Por lo general el trabajo de los maestros tiene como propósito orientar todas las asignaturas de un grupo reducido de estudiantes. Este número de estudiantes tiende a ser siempre inferior a 25 alumnos. Dependiendo de la escuela, estas pueden abarcar solo alguna de las etapas o una mezcla según el caso.

En el caso de los C.R.A. pueden haber diferentes localidades o aldeas en las que los alumnos de la educación infantil y primaria comparten una aula o varias, dependiendo de si el número de alumnos es suficiente para crear diferentes agrupaciones. En todo caso este tipo de agrupaciones generan aulas heterogéneas con diferentes necesidades a las que los docentes se deben adaptar de forma dinámica.

En el trabajo de Carolina Lacort Navarro se afirma que: *“los centros rurales deberán asumir metodologías distintas a los centros urbanos que les permitan paliar en gran parte dichas dificultades de adaptación. Esta adaptación depende de la voluntariedad del profesorado y de su implicación cuando debe trabajar en estos contextos para los que no se suele estar preparado ya que no se le facilita esta preparación al estudiar Magisterio. Los planes de estudios no hacen hincapié en el funcionamiento y la didáctica necesaria en estos centros”* [4].

Coincidiendo con lo comentado anteriormente se deduce que los profesores de estos centros están impartiendo el temario a los alumnos aplicando distintas metodologías. Esto a menudo se convierte en una tarea complicada. Teniendo en cuenta que en la mayoría de las ocasiones se comparte el aula, resulta complicado realizar un seguimiento de cada uno de los alumnos.

La aplicación web propuesta permitirá a los alumnos visualizar el contenido que los maestros proporcionan. Esta herramienta permite a los maestros utilizarla en situaciones en las que alguna de las partes no puede acudir al centro. Además, al tratarse de una plataforma totalmente flexible, esta aplicación web también podrá ser utilizada para transmitir el conocimiento. De esta forma la aplicación ayuda a los maestros a dedicar una mayor cantidad de tiempo en resolver las cuestiones que se consideren relevantes. La solución propuesta permite proporcionar contenido adicional para aquellos alumnos que desean profundizar en un área específico.

El desarrollo de este proyecto representa un gran reto para mí. De esta forma este proyecto me permite analizar, planificar, desarrollar e implantar una aplicación web partiendo de una base de conocimientos adquiridos en el grado y que parte de ellos se desean ampliar.

1.2. Idea y Objetivos

El objetivo de este trabajo es desarrollar una aplicación web con el propósito de solucionar un problema de accesibilidad en los colegios. Será proporcionada de forma gratuita y con el objetivo de ser una aplicación de software libre. Con este trabajo se pretende aplicar los conocimientos adquiridos durante el grado para analizar, planificar, diseñar, desarrollar, gestionar el proyecto y entregar a tiempo un producto mínimo viable (M.V.P.), fácil de mantener y evolucionar de manera constante.

A lo largo del trabajo se adquirirán conocimientos para el desarrollo de un *front-end* utilizando un *framework* que estará interaccionando con diferentes servicios. Para ello tras un estudio,

que se presentara más adelante, se decide que Angular¹ puede resolver este problema aplicando la tipología de aplicación de pagina única.

El M.V.P. que se desarrolla utilizando el *framework* arriba mencionado tiene como meta facilitar el proceso de transmisión del conocimiento. Para ello la aplicación utilizará la YouTube Data API (*Application Programming Interface*) para almacenar y reproducir el contenido educativo publicado en distintos canales de YouTube públicos o privados. Además, esta aplicación incorporará un chat para cada video que permitirá a los alumnos establecer de forma rápida un canal de comunicación con los maestros.

Como se explico previamente será una aplicación gratuita para diferentes plataformas aprovechando la ventaja y facilidad de transmitir el contenido a través de la web. De esta forma se facilita la implantación de nuevas metodologías a la hora de transmitir el conocimiento.

A continuación, se adjunta una figura que ilustra las principales herramientas que se aprenderán y usarán para realizar una prueba de concepto.

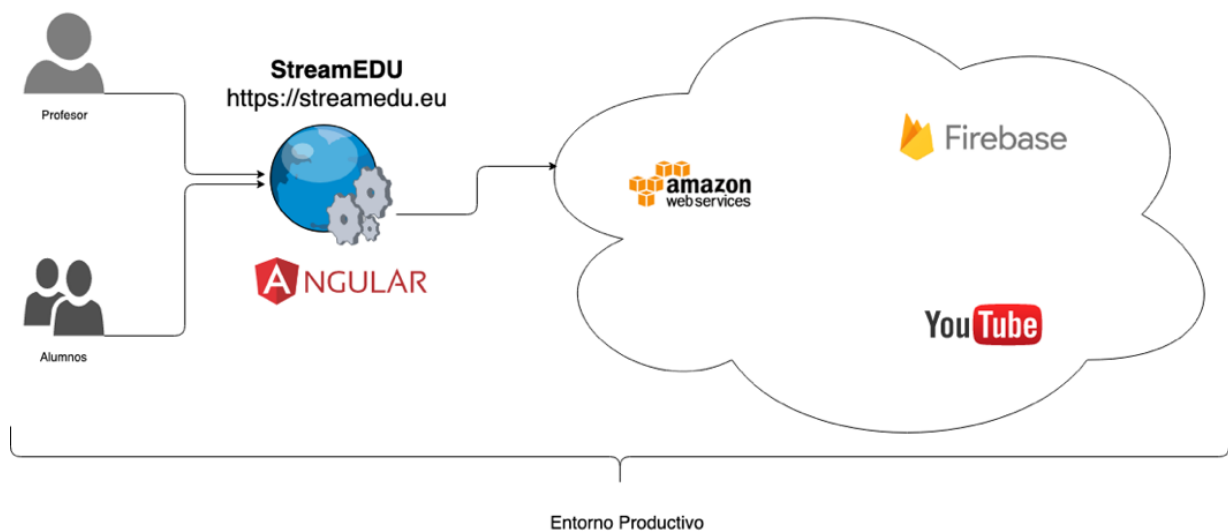


Fig. 1: Esquema básico del entorno productivo

En cuanto al entorno de desarrollo se utilizará metodología agile que permitirá analizar o replantear el progreso de la aplicación que se desarrolla. De esta forma se agiliza la entrega e integración continua de las funcionalidades que se describen en el apartado de análisis y diseño de software. En cuanto al control de versiones se utilizará un repositorio Git para almacenar y administrar las distintas versiones del código. Este repositorio será el encargado de notificar a una herramienta de automatización. Esta herramienta de automatización será la encargada de actualizar las nuevas funcionalidades implementadas en la aplicación durante todo el proceso de desarrollo.

¹ Framework Angular: <https://angular.io/>

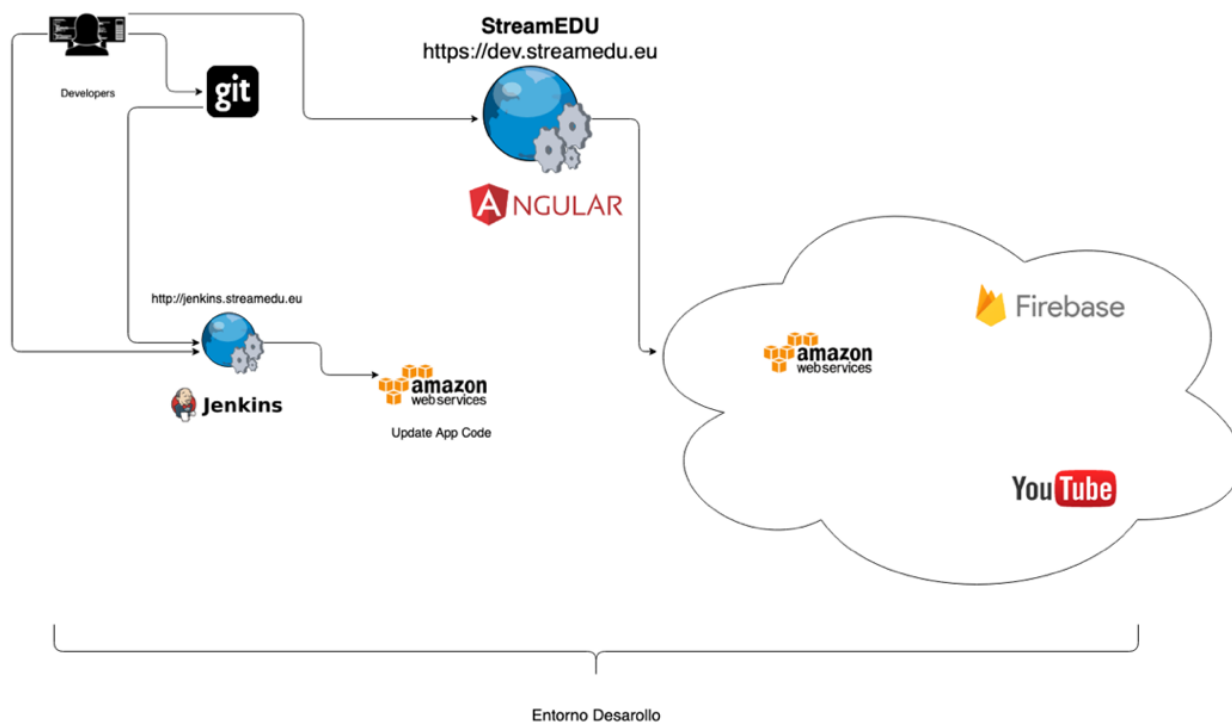


Fig. 2: Esquema básico del entorno de desarrollo

2. ESTADO DEL ARTE

2.1. Estudio sobre las tecnologías de la información

Hoy las tecnologías de la información y comunicación se han convertido en un bien imprescindible en la vida cotidiana. De forma casi instantánea lo que hace unos años se llamaba teléfono móvil se ha convertido en un Smartphone. Un Smartphone o teléfono inteligente brinda la capacidad de realizar otras funciones además de realizar llamadas o en enviar un simple mensaje. Además, estos dispositivos inteligentes han incorporado funcionalidades que hasta hora solo podrían realizarse en un ordenador. Actualmente todos los *Smartphones* incorporan un sistema operativo que hace posible ampliar el numero de sus funcionalidades. Aunque la mayoría de estos dispositivos, incluso los mas primitivos, de una forma u otra incluyen algún tipo de software que permitan almacenar contactos o gestionar los mensajes, con la aparición de los teléfonos inteligentes el limite de las funcionalidades que esos pueden incorporar lo limita la imaginación.

Utilizar estos dispositivos para navegar en internet no es algo que siempre ha sido posible. Esto hace que sea una de las características mas importantes que brindan este tipo de dispositivos. En enero de este año según el estudio que realiza cada año las compañías HootSuite² y We Are Social³ revela que en España el numero de suscripciones de móvil ha superado el de la población total. La penetración de usuarios que utilizan internet en España se sitúa en el 93% y se puede considerar como un buen indicador para el alcance de este proyecto.



Imagen 1: Uso del móvil, internet y redes sociales [5]

Otro dato relevante para determinar el nivel de uso de las TIC es el crecimiento digital anual que experimenta un crecimiento de 9% en numero de usuarios de internet con respecto al año anterior.

² HootSuite – Plataforma de marketing digital: <https://hootsuite.com/>

³ We are Social- Compañía de marketing y comunicación online2.0 <https://wearesocial.com/>

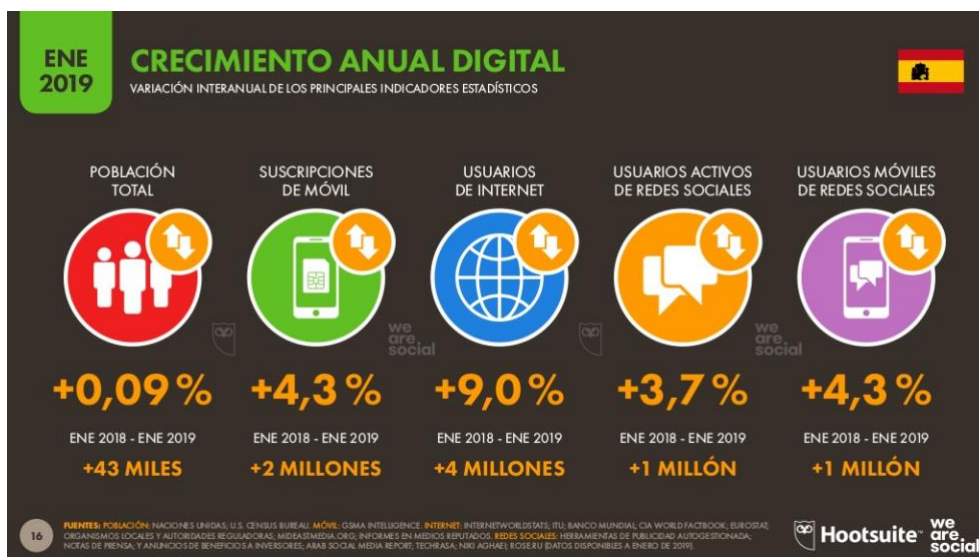


Imagen 2: Crecimiento anual digital [5]

Un alto nivel de dependencia del internet viene reflejado en la misma encuesta e indica que de media los encuestados se pasan alrededor de 5 horas diarias utilizando este servicio a través de cualquier dispositivo. De cara al proyecto, como dato interesante destacan las casi 3 horas diarias dedicadas a visualizar contenido en *Streaming* o a la carta en una televisión. Esta última cifra indica un alto nivel de aceptación para visualizar contenido a través de cualquier tipo de dispositivo.

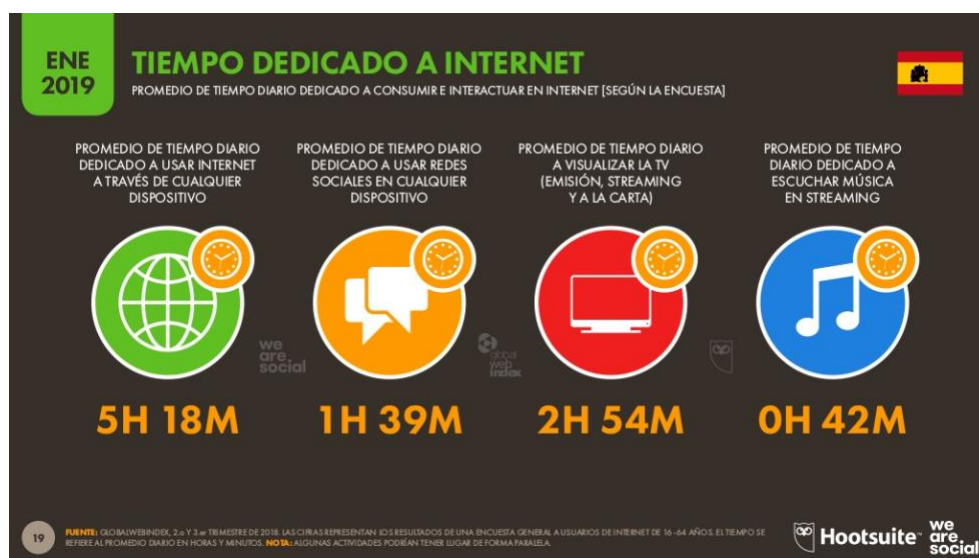


Imagen 3: Tiempo dedicado a internet [5]

Para ver mas en detalle si es posible la transmisión de contenido educativo a través de la red, la misma encuesta relévela datos interesantes sobre la velocidad de conexión. En cuanto a los dispositivos móviles, la velocidad media de conexión a internet se sitúa en 32 Mbps mientras que en las conexiones fijas esta ronda los 100Mbps de media. En la siguiente grafica se puede apreciar como en ambos casos las compañías proveedores de internet siguen mejorando ambos tipos de conexión.

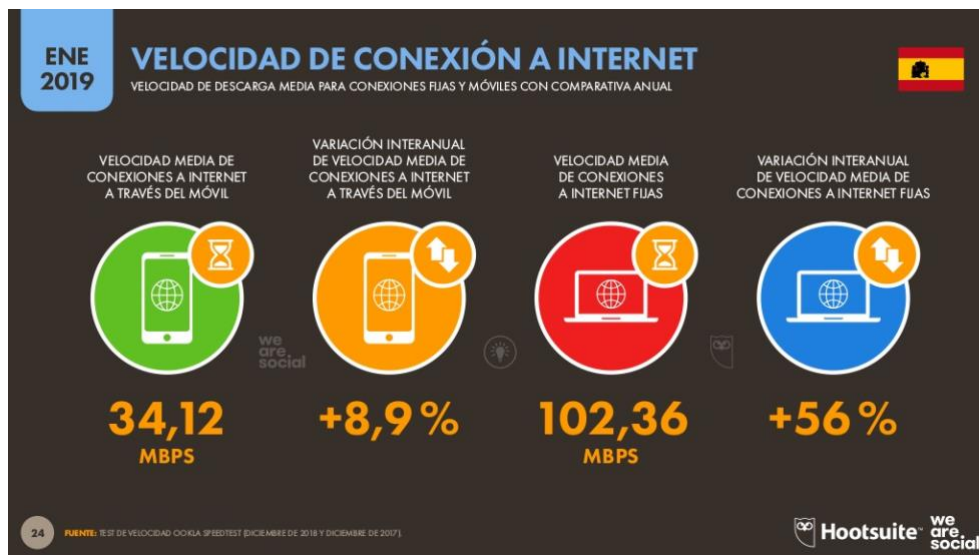


Imagen 4: Velocidad de conexión a internet [5]

Pero... ¿Que uso se dan a las conexiones a internet? De buenas a primeras volviendo al tiempo dedicado a las conexiones a internet observamos que alrededor de 3 horas se dedican a visualizar contenido en *streaming*. La misma encuesta detalla que de las actividades de *streaming* de contenido, la mayoría de los encuestados, un 93%, lo utilizan para ver videos online. A simple vista este dato puede que no ayude mucho, pero podría volverse positivo si parte de estos videos que se visualizan serian educativos.



Imagen 5: Actividades de *streaming* de contenido [5]

Otro dato interesante para desglosar de la *Imagen 3* es el tiempo que los encuestados dedican de media a las redes sociales. Los encuestados afirman que de media dedican alrededor de 100 minutos al día a las redes sociales. Entre las redes sociales mas activas de los usuarios destaca en las primeras dos posiciones la presencia de YouTube y Whatsapp. De la siguiente imagen se puede deducir que la mayoría de las aplicaciones utilizadas por los encuestados implementan de alguna forma u otra la transmisión del contenido video y la mensajería entre sus funcionalidades básicas.



Imagen 6: Las redes sociales más activas [5]

Se pueden encontrar infinidad de estadísticas y datos que confirmen el uso de la tecnología, pero en el anuario publicado por el Instituto Nacional de Estadística [6] destaca el uso de dispositivos móviles dentro del público más pequeño. En el año 2017 el número de menores de 10 a 15 años que son usuarios de internet o móvil ha aumentado de forma considerable con respecto a la pasada década. Un 95,1% de ellos han usado internet en los últimos tres meses. Otro dato que se puede apreciar de la siguiente imagen es el incremento del número de menores que ya disponen de un teléfono móvil.

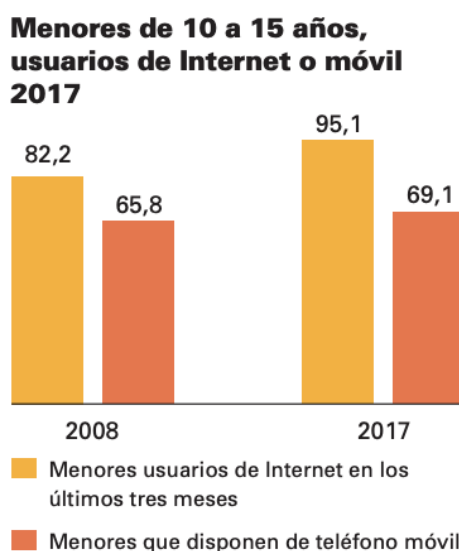


Imagen 7: Uso de internet o móvil de los menores de 10 a 15 años [6]

Teniendo en cuenta esta franja de edad solo un 26,2% de los niños de 10 años tienen un teléfono móvil y este valor aumenta hasta el 75,1% a los 12 años y llega hasta un 94,8% para los menores de 15 años.

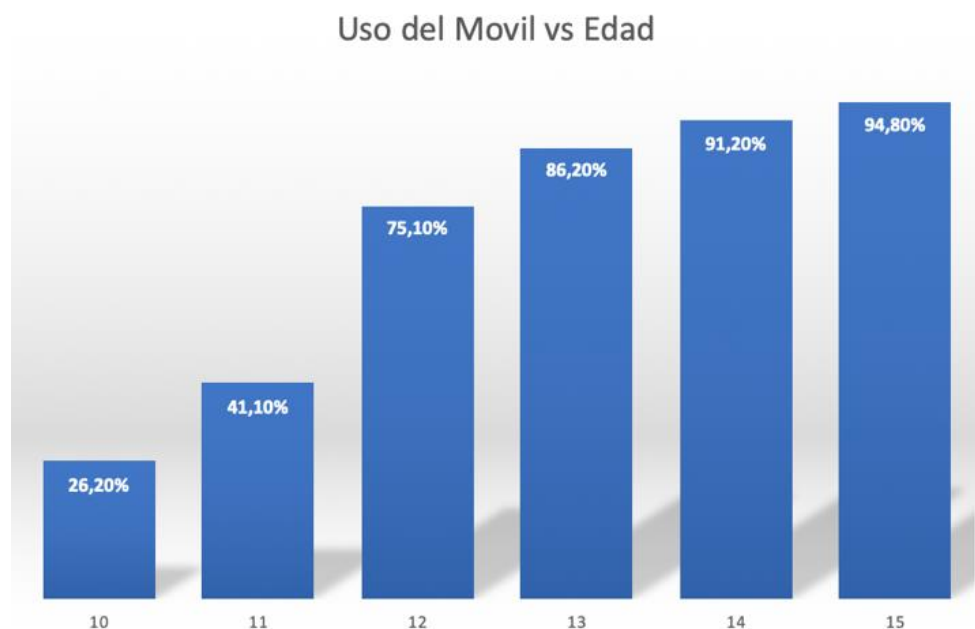


Fig. 3: Uso del móvil según la edad

Según el presidente autonómico de educación del sindicato CSIF en la comunidad Valenciana, el aprendizaje y el trabajo cooperativo es la manera de organizar las actividades dentro de las aulas para así poder convertirlas en una experiencia social. El intercambio de información entre los estudiantes aumenta su motivación tanto para el propio aprendizaje como para ayudar a mejorar los resultados de los demás. En este sentido las comunicaciones representan una base para adquirir conocimientos en el ámbito escolar [7].

2.2. Las aplicaciones educativas

El sector educativo al igual que otros sectores se han visto obligados a adaptarse a las nuevas tecnologías y cada vez es mas común encontrarse con plataformas educativas o e-learning con el propósito de acercar la educación a cualquier persona que dispone de acceso a internet.

En términos generales los Sistemas de Gestión de Aprendizaje o Learning Management System (LMS) es un tipo de software que se encuentra instalado en un servidor y ofrece la posibilidad de crear un campus virtual para la impartir formación en línea y son conocidas bajo el nombre de plataformas e-learning [8]. Este tipo de plataformas están siendo utilizadas comúnmente en las universidades, en centros de formación o en las empresas con el fin de gestionar a los usuarios, facilitándoles a través de ellas materiales didácticos, seguimiento personalizado sobre los avances realizados o la elaboración de informes entre otros. Las posibilidades a la hora de gestionar este tipo de plataformas son muy amplias y se pueden catalogar en dos principales grupos, las de código abierto y las comerciales.

- Plataformas LMS de código abierto

Cuando se habla de este tipo de plataformas de código abierto se entiende que se trata de un tipo de software que se ha diseñado con el propósito de ser distribuido o desarrollado de forma libre bajo una licencia de software libre.

Un factor importante para tener en cuenta a la hora de escoger este tipo de plataformas es que, aunque su distribución y uso es libre, de forma indirecta presentarían costes adicionales como podrían ser la administración y/o el mantenimiento que requiere en este caso de personal cualificado según la plataforma elegida.

- Plataformas LMS comerciales

En cuanto a las plataformas comerciales encontramos diferentes alternativas que se podrían catalogar según la forma de pago. En este aspecto tenemos:

- Plataformas LMS de pago por uso:

Este tipo de plataformas de pago por uso suelen encontrarse en la nube y vienen siendo una de las alternativas más utilizadas en los últimos años. En este caso se trata de una suscripción que va relacionada con el uso que se le dará a la plataforma. Entre los puntos para tener en cuenta a la hora de escoger este tipo de solución serían: la cantidad de usuarios activos de forma concurrente en la plataforma, la cantidad de cursos, módulos o la cantidad de información que distribuirá a través de esta solución.

- Plataformas LMS de pago por licencia

En este caso, este tipo de plataformas no tienen el código libre y por lo tanto hay que pagar una licencia para el uso y suelen ir acompañadas de un periodo de tiempo determinado. Entre los aspectos positivos que podríamos encontrar a esta modalidad sería que, mientras se disponga de una licencia activa, se podrán impartir tantos cursos y módulos como sean necesarios para un número ilimitado de alumnos.

- Plataformas LMS de código abierto

- a) Moodle⁴



Imagen 8: Logo Moodle [9]

Moodle (Module Object-Oriented Dynamic Learning Environment) que traduce sus siglas a Entorno Modular de Aprendizaje Dinámico Orientado a Objetivos, es la plataforma LMS de código abierto más utilizada, conocida y con más prestigio a nivel mundial, tanto en universidades, centros de formación o empresas. Se trata de una plataforma de aprendizaje diseñada para ofrecer un sistema integrado único, robusto y seguro para crear entornos de aprendizajes personalizados. La solución se puede descargar desde la página oficial para instalarla en un servidor web y además ofrece un listado de socios que pueden asistir la puesta en marcha.

Además, se trata de una solución mundialmente probada y de confianza, respaldada por una red mundial de más de 80 compañías de servicios que financian el desarrollo y la evolución continua de la plataforma. Entre las principales características destaca que esta diseñada para soportar tanto la enseñanza como el aprendizaje, la facilidad de uso, disponible en múltiples idiomas y con actualizaciones constantes.

⁴ Moodle – Plataforma de aprendizaje: <https://moodle.org/?lang=es>

Dispone de un entorno personalizable, escalable a entornos de uso de cualquier tamaño y para ser utilizado en cualquier lugar y desde cualquier tipo de dispositivo. También cuenta con una amplia comunidad que comparte numerosas extensiones.

b) Canvas LMS⁵



Imagen 9: Logo Canvas [10]

A diferencia de otros LMS, Canvas es de código abierto y funciona completamente en la nube. Dispone de una interfaz gráfica muy visual e intuitiva. Al igual que en el caso anterior, esta herramienta está diseñada para ser utilizada en navegadores web, en dispositivos móviles y *tablets*.

Dispone de dos versiones, una enfocada al sector empresarial y otra opción gratuita para los profesores. Este LMS según las necesidades permite la creación de aulas virtuales interactivas. Con la ayuda de estas aulas virtuales se permite el desarrollo de actividades que incluye funcionalidades como, por ejemplo: una bandeja de correos del Aula Virtual, un tablón de anuncios, un sistema de seguimiento de actividades, la evaluación de estas o incluso un sistema de alertas de las actividades a evaluar [11].

c) Chamilo LMS⁶



Imagen 10: Logo Chamilo [12]

Según la propia definición que proporcionan en su web, el término Chamilo se usa para denominar el software Chamillo, un LMS (y LCMS) de software libre. El proyecto nace en el año 2010 de la mano de la asociación sin ánimo de lucro que lleva el mismo nombre que el proyecto. Este representa una agrupación de personas y organizaciones que lo usan y tienen el objetivo de mejorar la disponibilidad de una educación de calidad en todo el mundo. Teniendo en cuenta su corta trayectoria, está presente en más del 80% de los países del mundo y su propósito es acercar la educación a países en procesos de desarrollo.

Entre las principales ventajas en términos específicos es que se trata de un software libre que avanza todo el tiempo, su sencillez de uso, su atractivo visual y su eficiencia en el lado del servidor. En términos funcionales, comparando con otros LMS tiene funcionalidades que en un 90% se cruzan con las de otros y que dispone de funcionalidades que otros LMS no tienen.

⁵ Canvas – Plataforma de aprendizaje: <https://es.guides.instructure.com/m/70283>

⁶ Simplificar el e-learning con Chamilo LMS – Plataforma de aprendizaje: <https://chamilo.org/es/>



d) Sakai⁷

Imagen 11: Logo Sakai [13]

Al igual que las anteriores soluciones Sakai es un LMS de código abierto creado entre varias universidades americanas y que ofrece una amplia gama de características para satisfacer distintas necesidades. Incluyen todas las herramientas estándar de aprendizaje, enseñanza y colaboración en línea al igual que las demás plataformas de aprendizaje.

Dispone de un amplio abanico de funcionalidades aportadas por comunidad y cuenta con numerosas integraciones de herramientas externas. Estas integraciones con herramientas externas se encuentran bajo el nombre de componentes adicionales y permiten configurar el sistema según las necesidades demandadas. La solución ha sido implementada en mas de 100 universidades y algunas de ellas son españolas como, por ejemplo: la Universidad Complutense de Madrid o la Universidad politécnica de Valencia.



e) LearnPress - WordPress LMS⁸

Imagen 12: Logo LearnPress [14]

WordPress es un sistema de gestión de contenidos (CMS) que permite crear y mantener sitios web. La propia definición que encontramos en su web, WordPress es un software libre diseñado para todos, donde sus principales características son la accesibilidad, el rendimiento, la seguridad y la facilidad de uso. En un principio requiere de una configuración mínima ofreciendo de esta forma un amplio abanico de características para el crecimiento y el éxito.

LearnPress es un complemento completo gratuito de LMS para WordPress que se asemeja al Moodle. Este complemento permite crear cursos de forma personalizada adaptándose a las necesidades del plan de estudios. Permite crear lecciones, añadir contenido visual, vender cursos o elaborar cuestionarios a través de su fácil interfaz de usuario sin la necesidad de los conocimientos de programación.

- Plataformas LMS comerciales



f) Blackboard LMS⁹

Imagen 13: Logo Blackboard [15]

La solución de Blackboard ha sido diseñada por la compañía Blackboard Inc fundada en el año 1997. Después de varias fusiones y versiones de software desarrolladas para plataformas de aprendizaje, en el año 2005 esta licencia sus aplicaciones. Al llevar una trayectoria tan larga, hace que con los años su experiencia sea valorada y la convierte en la plataforma mas prestigiosa dentro de la categoría comercial.

⁷ Sakai – Plataforma de aprendizaje: <https://www.sakaiproject.org/>

⁸ LearnPress – Plataforma de aprendizaje: <https://es.wordpress.org/plugins/learnpress/>

⁹ Blackboard – Plataforma de aprendizaje <https://es.blackboard.com/index.html>

Entre las ventajas que brinda esta plataforma encontramos la facilidad de comunicación e interacción con los alumnos, permite agregar materiales, crear encuestas, exámenes o llevar un seguimiento del aprendizaje.

Cabe destacar que Blackboard Learn es una plataforma e-learning utilizada por una multitud de universidades de renombre, pero que también ofrece soluciones para el sector público, educación primaria o funciones complementarias para empresas. Dispone de diferentes tipos de plataformas según el uso entre las cuales encontramos a Blackboard Learn, Blackboard Connect o Blackboard Collaborate.

g) eDucativa¹⁰



Imagen 14: Logo eDucativa [16]

Se trata de una plataforma educativa sencilla para poder gestionar cursos a través de internet siendo utilizada por todo tipo de empresas e instituciones. Educativa es una alternativa de LMS bastante económica en función del alcance de la solución.

De igual forma que las demás soluciones de pago expuestas, eDucativa ofrece diferentes soluciones que se adaptan a diferentes necesidades. Entre las principales características destaca: la facilidad a la hora de implementar las diferentes funcionalidades y secciones que esta ofrece. Se ofrece una serie de funcionalidades como, por ejemplo: el depósito de archivos, foros de debate, mensajería, chat, una wiki, calendario o la posibilidad de realizar encuestas entre otros.

h) OpenText FirstClass¹¹



Imagen 15: Logo OpenText [17]

En esta ocasión FirstClass es una solución de pago por uso y es ofrecida por la empresa OpenText. Debido a su forma de suscripción esta solución no requiere de instalación en el servidor y además puede utilizarse en la nube, adaptándose a cualquier tipo de dispositivo.

La solución está diseñada para que la información sea fácilmente accesible y permite la colaboración y la comunicación de una forma fácil y segura. Permite la personalización, la integración con otras aplicaciones o el archivado de contenido de forma segura y confiable.

2.3. Elección del *framework* para el desarrollo

¿Que es un *framework*? Un *framework* en el mundo del desarrollo se define como un conjunto de: herramientas, librerías y buenas prácticas, que permiten resolver una serie de problemas. Entre las principales ventajas que ofrece un *framework* se encuentra la no necesidad de plantearse la estructura global de la aplicación. La mayoría de ellos proporcionan un esqueleto

¹⁰ eDucativa - Plataforma de aprendizaje <http://www.educativa.com/campus/>

¹¹ FirstClass - Plataforma de aprendizaje <https://www.opentext.com/products-and-solutions/products/specialty-technologies/firstclass>

básico con la lógica que necesita la aplicación. Además, hace que el desarrollador no este continuamente “reinventado la rueda” y se centre en el problema que se desea resolver. Al mismo tiempo este facilita la implementación de funcionalidades, que son de uso común y que en la mayoría de las ocasiones ya se encuentran resueltas.

En los últimos años han aumentado considerablemente la cantidad de *frameworks* para diseñar una aplicación *front-end*. Cada uno de estos *frameworks* es capaz de construir grandes aplicaciones web y aquí llega la gran duda: ¿Cual de ellos se debería escoger para el proyecto? La respuesta no es fácil y la mayoría implican un coste inicial de aprendizaje.

La elección del *framework* se basa en el cumplimiento de las necesidades de la aplicación que se desarrolla. Para ello se decide usar un *framework* que permite crear una aplicación web progresiva o PWA (Progressive Web Apps). Con respecto a otras tecnologías las PWA permiten llevar la aplicación a cualquier sistema que dispone de un navegador web. Las aplicaciones web del tipo PWA permiten mejorar la experiencia del usuario, implementando una serie de características. Estas características se refieren al manifiesto de la aplicación, la reducción de los tiempos de carga de los servicios y el Shell de aplicación.

Por un lado, el manifiesto de la aplicación es un archivo en formato *json* que describe la aplicación. Además, este archivo permite controlar como se muestra la aplicación al usuario e indica lo que puede iniciar y que apariencia tendrá la aplicación al iniciarse.

Otro punto para tener en cuenta es el uso de los servicios que permite mejorar el tiempo de carga, al guardar en cache los *assets* de la aplicación e incluso las llamadas a API.

En el caso de la Shell de aplicación esta se refiere a los archivos HTML, CSS y JavaScript que se requieren para activar la interfaz de usuario. De esta forma estos archivos pueden ser almacenados en cache y se mejora de esta forma el rendimiento.

En efecto, los candidatos que mejor se adaptan a estas necesidades son Angular, React y Ionic. Dicho de otra manera, React y Ionic llevan menos trayectoria comparándose con Angular. Si se evalúa basándose en cuanto a comunidad de usuarios y cantidad de documentación disponible, Angular ofrece mas confianza.

Con respecto a la curva de aprendizaje React no es un *framework* completo y las características avanzadas requieren del uso de librerías que dependen de terceros. Por otra parte, el *framework* Ionic en sus principios fue construido utilizando como base Angular. Aunque las bases de Ionic han cambiado desde sus principios, hoy este ofrece cierta compatibilidad con las aplicaciones desarrolladas en Angular.

Por tal razón se decide en términos de: documentación, comunidad y aprendizaje que la elección mas adecuada es el uso del *framework* Angular. En tal sentido este permite un diseño modular usando componentes que separan la interfaz de usuario del comportamiento en forma de código TypeScript. En conjunto Angular es un *framework* moderno construido de forma integra utilizando el lenguaje de programación TypeScript. El lenguaje TypeScript es un superconjunto de ECMAScript 6 incluyendo de esta forma todas las funcionalidades de EC6. Además, incorpora una capa de abstracción por encima con funcionalidades extra como por ejemplo el tipado estático y objetos basados en clases.

3. ANÁLISIS Y DISEÑO DE SOFTWARE

El objetivo principal de este apartado es el análisis y diseño del software de la aplicación. Por lo que se comienza a explicar los requisitos y la arquitectura de la aplicación. Después se explica la descripción funcional de la aplicación con los principales casos de uso. Seguidamente se mostrarán y describirán los *mock-ups* con las pantallas de la aplicación. Finalmente se explica el *back-end* de la aplicación y se realiza una estimación de los costes asociados a la aplicación.

3.1. Requisitos de la aplicación

En este punto, a partir de la información extraída de las diferentes soluciones analizadas anteriormente, se definen los principales requisitos de la aplicación. En primer lugar, se detecta la necesidad de diferenciar entre 3 perfiles de usuarios: alumnos, profesores y administradores. Al mismo tiempo estos perfiles compartirán una serie de vistas y además dependiendo del tipo de usuario tendrá una serie de vistas y funcionalidades que los diferencia.

Requisitos funcionales de los alumnos:

- El alumno debe poder registrarse, iniciar sesión y finalizar sesión.
- El alumno debe poder modificar su nombre y foto de perfil.
- El alumno debe poder realizar comentarios e iniciar un chat.
- El alumno debe poder consultar las preguntas frecuentes.
- El alumno debe poder visualizar el contenido video publicado.
- El usuario debe poder realizar una búsqueda del contenido video publicado.

Requisitos funcionales del profesorado:

- Los profesores deben poder registrarse, iniciar sesión y finalizar sesión
- Los profesores deben poder modificar su nombre y foto de perfil
- Los profesores deben poder añadir, modificar y eliminar las preguntas frecuentes
- Los profesores deben poder realizar una búsqueda del contenido video publicado.
- Los profesores deben poder visualizar, subir, editar y suprimir el contenido video.
- Los profesores deben poder mantener mas de una conversación con diferentes alumnos.

Requisitos funcionales del administrador:

- Los administradores deben poder asignar y revocar permisos de los usuarios.
- Los administradores deben poder realizar una búsqueda de los usuarios registrados.
- Los administradores deben poder visualizar, subir, editar y suprimir el contenido video.
- Los administradores deben poder visualizar, añadir, modificar o eliminar las preguntas frecuentes.

Requisitos no funcionales del sistema:

- El sistema debe usar conexiones cifradas utilizando SSL/TLS
- El sistema debe actualizar los datos cada vez que un usuario lo solicita.
- El sistema debe notificar a los usuarios cuando se producen cambios.
- El sistema debe controlar el acceso de las personas a la plataforma.
- El sistema debe comprobar de forma obligatoria el cumplimiento de los formularios para hacer uso de la aplicación.
- El sistema debe notificar al usuario la falta de conexión.
- El sistema debe estar disponible y accesible desde cualquier dispositivo móvil con acceso a internet a través del navegador o de aplicaciones Android o iOS.
- El sistema debe asegurar una alta disponibilidad y rápida entrega del contenido.

Requisitos no funcionales para el sistema productivo y desarrollo continuo:

- La solución debe disponer de al menos dos entornos, uno entorno productivo y otro entorno para el desarrollo evolutivo.
- La solución de automatización debe desplegar las actualizaciones de código de forma automática para ambos entornos.
- Se debe comprobar que el código se compila correctamente antes de actualizar las nuevas versiones.

3.2. Arquitectura

En cuanto a la arquitectura, se utiliza un modelo cliente - servidor formado por el dispositivo y los diferentes servidores que proporcionan los distintos servicios que se detallaran a continuación. Ahora bien, del lado del cliente la arquitectura física como por ejemplo el sistema operativo o la gestión de la memoria esta totalmente controlada por el dispositivo. Dicho de otra manera, solo se podrá acceder a los recursos y los datos de usuario que la aplicación permite.

Ahora bien, la arquitectura lógica de la aplicación se puede dividir en tres grandes bloques. De modo que este diagrama de bloques de la aplicación queda formado por: los recursos del sistema operativo, la aplicación y las librerías externas utilizadas.

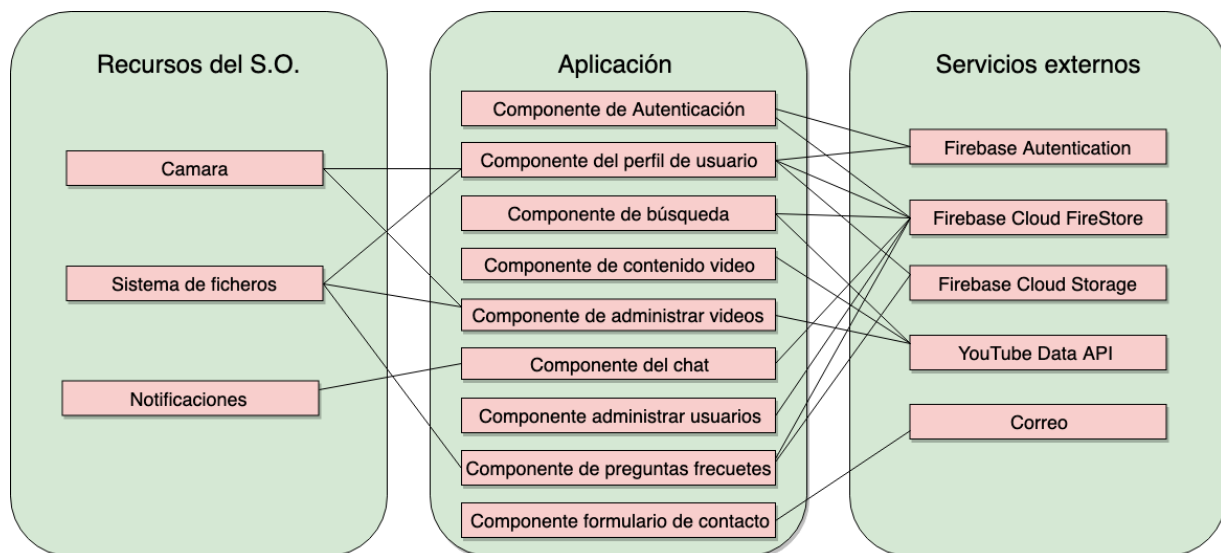


Fig. 4: Diagrama de bloques de arquitectura de la aplicación

Por lo que se refiere el diagrama de bloques se explicara brevemente a continuación las dependencias de los componentes. Mas adelante se detallará con mas profundidad los casos de uso de estos componentes.

a) Componente de autenticación

Es el encargado de gestionar el registro y la autenticación de los usuarios. En particular debe asegurar el envío de los datos al servicio de autenticación de Firebase para que estas sean validadas.

b) Componente del perfil de usuario

La idea principal de este componente se basa en mostrar los datos personales de usuario que se reciben del servicio de autenticación. Gracias a este componente el usuario puede cambiar el nombre de usuario o la foto de perfil. Para esto se necesita de los servicios de FireStore y Storage de Firebase. Debido a que el usuario puede cambiar la foto de perfil este componente necesita acceder a los recursos del sistema de ficheros o de la cámara del dispositivo.

c) Componente de búsqueda

Con el motivo de facilitar la localización del contenido video o de los usuarios se diseña este componente. Sera el encargado de proporcionar las herramientas necesarias para realizar las búsquedas. Teniendo en cuenta estas necesidades este componente debe comunicarse con los servicios de Youtube Data API para localizar el contenido video y Firebase FireStore para localizar a los usuarios registrados en la aplicación.

d) Componente de contenido video

Este componente es el encargado de mostrar todo el contenido video publicado en la aplicación. Para ello este componente necesita comunicarse con el servicio de YouTube. Además de mostrar una lista de videos, también se recompila y muestra una serie de información relevante como, por ejemplo: el titulo, una breve descripción o la categoría del contenido video publicado.

e) Componente de administrar videos

El componente de administrar videos extiende las funcionalidades del componente de contenido video, facilitando de esta manera la gestión del contenido. Para ello necesita comunicarse con el servicio de YouTube con el fin de editar, modificar o eliminar los videos publicados.

f) Componente del chat

Dado que la comunicación es un aspecto importante en cualquier entorno, este componente es el responsable de gestionar las dudas entre los principales actores de aplicación. Para ello, este componente necesita comunicarse con el servicio de Firebase Firestore que es el encargado de almacenar los mensajes. Además, este componente necesita acceder al recurso de notificaciones que provee el sistema operativo.

g) Componente administrar usuarios

Con el objetivo de tener un control en la aplicación, se diseña este componente para administrar los usuarios. Puesto que la aplicación almacena un registro de los usuarios registrados, este componente necesita comunicarse con el servicio de Firebase Firestore para poder administrarlos.

h) Componente de preguntas frecuentes

Teniendo en cuenta las posibles dudas que pueden llegar a surgir en cuanto al uso de la aplicación, se necesita de un componente que administre las preguntas frecuentes. En consecuencia, estas preguntas deben ser almacenadas y administradas utilizando los servicios de Firebase Firestore y Firebase Storage.

i) Componente formulario de contacto

Por último, este componente de formulario de contacto necesita utilizar un servicio de correo para poder enviar las consultas realizadas.

3.3. Descripción funcional

A continuación, se explicarán las funcionalidades del sistema. Como se comento en los primeros apartados, cuando se realizo una primera introducción de la aplicación, esta distinguirá entre tres tipos de usuarios: alumnos, profesores y administrador.

a) Lista de preguntas frecuentes sobre el funcionamiento

Cuando se accede a la aplicación, esta dispondrá en el menú de una vista con una lista de preguntas frecuentes, que podrán ser visualizadas por cualquier visitante de la aplicación. El contenido de esta lista podrá ser editado a través del panel de administración, que solo será accesible para los perfiles de los profesores y administradores. Este apartado será útil para incluir manuales de uso de la aplicación con el propósito de conseguir una mayor usabilidad del aplicativo.

b) Formulario de contacto

De forma similar a lista de preguntas frecuentes, cuando un nuevo visitante accede a la aplicación este dispondrá de un formulario de contacto para poder comunicarse con la persona que administra la aplicación.

c) Lista de videos

En cuanto a la lista de videos esta ofrecerá unas funcionalidades u otras dependiendo de tipo de usuario. A continuación, se especifican estas funcionalidades para cada uno de los perfiles:

i. Vista de los alumnos

En primer lugar, se encuentran los alumnos que deberán estar autenticados para poder acceder al contenido que proporciona la aplicación. La cantidad de videos que se le mostraran en la aplicación dependerá de las listas de contenido que el alumno selecciona. Esta cantidad de contenido dependerá de cada alumno y sus preferencias de aprendizaje.

Al realizar clic sobre un video este será redirigido a otra vista dentro de la aplicación. En este punto el alumno podrá visualizar el contenido de forma similar al que se muestra en YouTube y junto a el, un chat donde podrá realizar las consultas o dudas que le puedan surgir sobre el video que esta visualizando.

Cuanto el usuario esta redirigido a la pagina del video a visualizar, el contenido del chat será cargado por defecto y de esta forma esta incluyendo las consultas previas realizadas sobre ese video. Por lo tanto, los alumnos también podrán consultar las preguntas realizadas otros alumnos y en caso de que su duda sobre el contenido sea idéntica a la de otro alumno esta pueda ser resuelta con la mayor brevedad.

ii. Vista de los profesores

A diferencia de los alumnos los profesores dispondrán de mas funcionalidades en esta vista. Entre ellas encontraran la posibilidad de editar los metadatos de los videos que han publicado en su cuenta de YouTube, subir un nuevo video a su canal de YouTube o eliminar un video subido a su canal de YouTube. En ningún caso se le permitirán a este tipo de usuarios modificar el contenido de los demás.

iii. Vista de los administradores

El perfil de administrador, a diferencia de los definidos anteriormente tendrá acceso total para eliminar o modificar el contenido publicado por los profesores en la plataforma.

d) Perfil de usuario

Cuando un nuevo usuario accede por primera vez a la aplicación este dispondrá de una vista donde se visualiza su foto de perfil, el nombre y la dirección de correo con la que se ha registrado. En una primera instancia los nuevos usuarios registrados tendrán asignado por defecto el rol de alumno. Esto solo les permita acceso a las funcionalidades de listar videos, consultar preguntas frecuentes, modificar su perfil o cerrar sesión.

e) Administrar roles de usuarios

Cuando un usuario del tipo administrador accede a la aplicación este dispondrá en el menú de una vista que le lista todos los usuarios registrados en la aplicación. Podrá realizar una búsqueda sobre los nombres de los usuarios y también podrá asignar o revocar permisos de tipo editor.

3.4. Casos de uso

Con el propósito de facilitar la comprensión de todos los diagramas que se explicaran a continuación se han dividido según la estructura de un proyecto en Angular.

Al acceder a la aplicación cualquier usuario dispone un menú de navegación que permite al usuario registrarse, autenticarse, cerrar sesión, consultar las preguntas o realizar una consulta mediante el formulario de contacto.

Diagrama casos de uso Acceso a la aplicación:

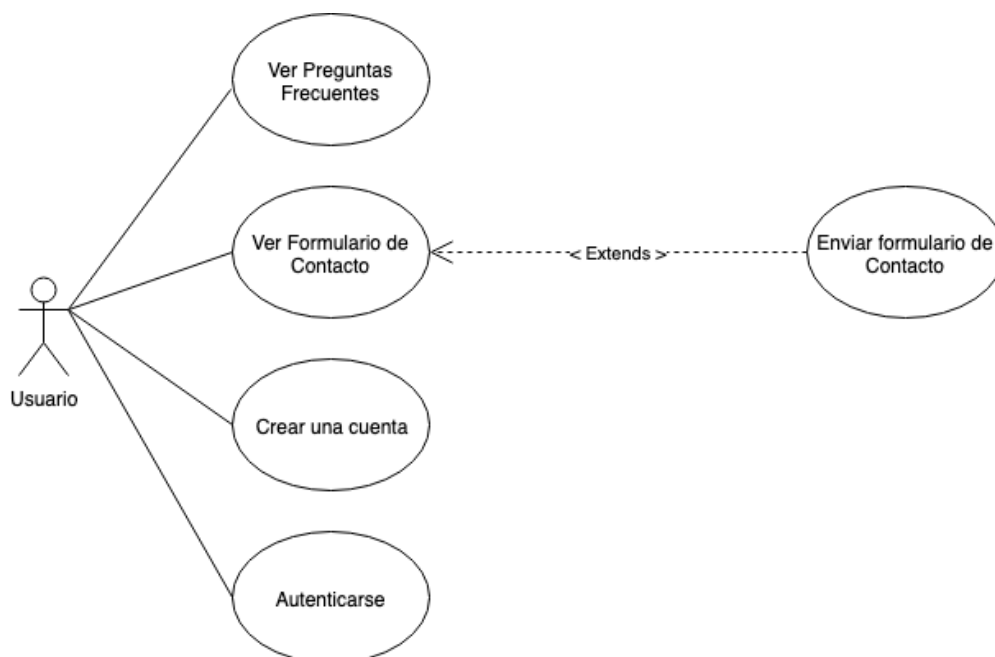


Fig. 5: Diagrama Casos de uso - Acceso a la aplicación

Descripción de los casos de uso mas relevantes:

Caso de uso	Ver las preguntas frecuentes
Descripción	El usuario accede a la aplicación y consulta las preguntas frecuentes
Actores	Cualquier visitante de la aplicación
Pre-Condición	1. Acceder a la aplicación 2. Debe haber como mínimo una pregunta publicada
Flujo Normal	1. El usuario accede a https://www.streamedu.eu 2. Selecciona en el menú Preguntas Frecuentes 3. Se muestra un listado de preguntas frecuentes
Éxito	El usuario ha podido consultar la lista de preguntas frecuentes

Tabla 1: Caso de uso - Preguntas Frecuentes

Caso de uso	Ver el formulario de Contacto
Descripción	El usuario accede a la aplicación y realiza una consulta
Actores	Cualquier visitante de la aplicación
Pre-Condición	1. Acceder a la aplicación 2. Cumplimentar todos los campos
Flujo Normal	1. El usuario accede a https://www.streamedu.eu 2. El usuario selecciona en el menú de navegación Contacto 3. En el cuerpo de la pagina aparece un formulario de contacto 4. El usuario debe cumplimentar todos los campos 5. El usuario envía el formulario de contacto 6. Un mensaje de confirmación será mostrado
Éxito	El usuario envía una consulta a través del formulario de contacto

Tabla 2: Caso de uso - Formulario de Contacto

Caso de uso	Crear una cuenta
Descripción	El usuario accede a la aplicación y se registra
Actores	Cualquier visitante de la aplicación
Pre- Condición	1. Acceder a la aplicación 2. Cumplimentar todos los campos
Flujo Normal	1. El usuario accede a https://www.streamedu.eu 2. El usuario selecciona en el menú de navegación Registrar 3. Se muestra un formulario para realizar el registro 4. El usuario debe cumplimentar todos los campos 5. El usuario envía el registro 6. El usuario será redirigido a su perfil de usuario
Éxito	El usuario ha podido registrarse y esta redirigido a su perfil de usuario

Tabla 3: Caso de uso - Crear una Cuenta

Caso de uso	Autenticarse
Descripción	El usuario accede a la aplicación y se autentica
Actores	Cualquier usuario registrado en la aplicación
Pre-Condicion	1. Acceder a la aplicación 2. Ingresar el email y la contraseña o disponer de una cuenta de Gmail
Flujo Normal	1. El usuario accede a https://www.streamedu.eu 2. Selecciona en el menú la opción Login 3. Se muestra un formulario para iniciar sesión 4. El usuario debe cumplimentar los campos email y contraseña 5. El usuario envía la solicitud de autenticación 7. El usuario será redirigido a su perfil de usuario
Éxito	El usuario se autentica y esta redirigido a su perfil de usuario

Tabla 4: Caso de uso – Autenticarse

Caso de uso	Cerrar sesión
Descripción	El usuario cierra la sesión iniciada en la aplicación
Actores	Cualquier usuario autenticado en la aplicación
Pre-Condicion	1. Estar autenticado
Flujo Normal	1. El usuario accede a https://www.streamedu.eu 2. El usuario se autentica 3. El usuario cierra la sesión
Éxito	El usuario cierra sesión y esta redirigido a la pagina de inicio

Tabla 5: Caso de uso - Cerrar Sesión

Diagrama casos de uso en Lista Videos:

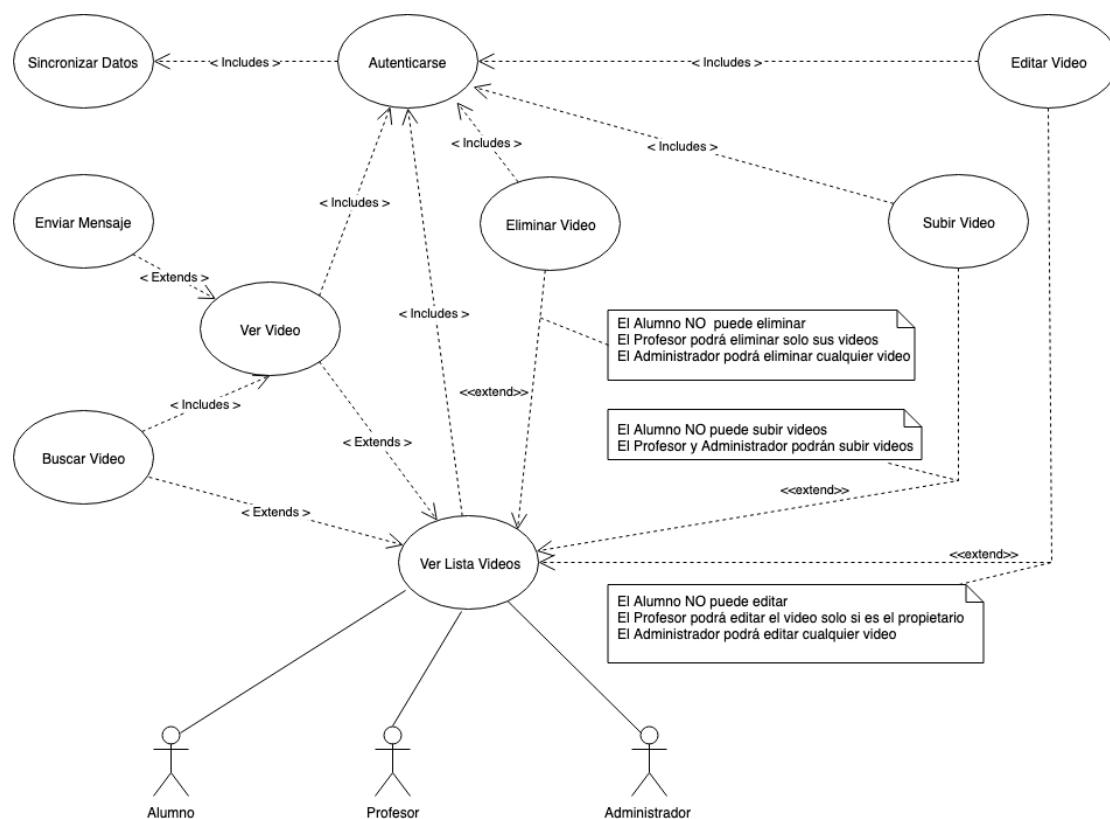


Fig. 6: Diagrama Casos de uso - Lista de Videos

Descripción de los casos de uso mas relevantes:

Caso de uso	Sincronizar Datos
Descripción	El usuario autentica en la aplicación y los datos sincronizan
Actores	Cualquier usuario autenticado en la aplicación
Pre-Condición	<ol style="list-style-type: none"> 1. Estar autenticado 2. Al menos un video publicado en la aplicación 3. Disponer de una cuenta de Gmail
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario se autentica 2. Se sincroniza el contenido necesario de este usuario
Éxito	El usuario dispone del contenido autorizado

Tabla 6: Caso de uso - Sincronizar Datos

Caso de uso	Ver Lista Videos
Descripción	El usuario autenticado accede a la lista de videos
Actores	Cualquier usuario autenticado en la aplicación
Pre-Condicion	<ol style="list-style-type: none"> 1. Estar autenticado
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario se autentica 2. El usuario hace clic en Lista Videos 3. Se listan videos en el cuerpo de la aplicación
Éxito	El usuario visualiza una lista de videos

Tabla 7: Caso de uso - Ver Lista Videos

Caso de uso	Ver Video
Descripción	El usuario visualiza un video
Actores	Cualquier usuario autenticado en la aplicación
Pre-Condición	<ol style="list-style-type: none"> 1. Estar autenticado 2. Se encuentra en la lista de videos 3. Selecciona el video
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario entra a la lista de videos 2. El usuario decide que video visualizar 3. El usuario hace clic en Ver Video 4. Se redirige a otra pagina para visualizar el video 5. El usuario tiene que hacer clic en el botón Play
Éxito	El usuario visualiza el contenido del video

Tabla 8: Caso de uso - Ver Video

Caso de uso	Enviar Mensaje
Descripción	El usuario envía un mensaje al chat del video
Actores	Cualquier usuario autenticado en la aplicación
Pre-Condición	<ol style="list-style-type: none"> 1. Estar autenticado 2. Estar visualizando un video
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario esta autenticado 2. El usuario selecciona un video de la lista 3. El se encuentra en la pagina del video 4. Escribe un mensaje 5. Envía el mensaje 6. Se almacena el mensaje 7. Se muestra el mensaje en el chat
Éxito	El usuario envía un mensaje

Tabla 9: Caso de uso - Enviar Mensaje

Caso de uso	Buscar Video
Descripción	El usuario busca un video por su nombre
Actores	Cualquier usuario autenticado en la aplicación
Pre-Condición	<ol style="list-style-type: none"> 1. Estar autenticado 2. Encontrarse en la lista de videos
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario esta autenticado 2. El usuario selecciona la pagina Lista Videos 3. El usuario escribe al menos un carácter 4. Envía la solicitud de búsqueda con el botón buscar 5. Se muestra los resultados o un mensaje de advertencia
Éxito	Visualiza una lista de videos o un mensaje de no coincidencias

Tabla 10: Caso de uso – Buscar Video

Caso de uso	Editar Video
Descripción	El usuario edita un video de la lista
Actores	Cualquier usuario autenticado con rol de administrador o editor.
Pre-Condición	<ol style="list-style-type: none"> 1. Estar autenticado 2. Encontrarse en la lista de videos 3. El rol editor solo puede editar sus videos 4. El rol administrador puede editar cualquier video
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario esta autenticado 2. Selecciona la pagina Lista Videos 3. Escoge el video a editar haciendo Clic en Editar Video 4. Es redirigido a la pagina del video a editar 5. El usuario actualiza los metadatos 6. Envía la petición de actualizar el video 7. El video con los metadatos actualizados será mostrado
Éxito	El usuario actualiza los metadatos de un video

Tabla 11: Caso de uso - Editar Video

Caso de uso	Subir Video
Descripción	El usuario añade un nuevo video
Actores	Cualquier usuario autenticado con rol de administrador o editor.
Pre-Condición	<ol style="list-style-type: none"> 1. Estar autenticado 2. Encontrarse en la lista de videos 3. Disponer de permisos suficientes 4. Disponer de un video
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario esta autenticado 2. El usuario selecciona la pagina Lista Videos 3. Clic en Añadir Video 4. Se redirige a la pagina para subir un video 5. Cumplimenta los campos que considera oportunos 6. Selecciona el video 7. Clic en subir video 8. El proceso finaliza y un mensaje notificara el estado.
Éxito	El video es subido a YouTube

Tabla 12: Caso de uso – Subir Video

Caso de uso	Eliminar Video
Descripción	El usuario elimina un video
Actores	Usuario autenticado con un rol de administrador o editor.
Pre-Condicion	<ol style="list-style-type: none"> 1. Estar autenticado 2. Encontrarse en la lista de videos 3. Disponer de permisos suficientes 4. El rol editor solo puede borrar sus videos. 5. El rol administrador puede borrar cualquier video
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario esta autenticado 2. El usuario selecciona la pagina Lista Videos. 3. Clic en Borrar Video. 4. Se muestra un mensaje de confirmación. 5. El usuario confirma o deniega la operación.
Éxito	El video es eliminado de YouTube

Tabla 13: Caso de uso – Eliminar Video

Diagrama casos de uso Administrar Usuarios y Preguntas Frecuentes:

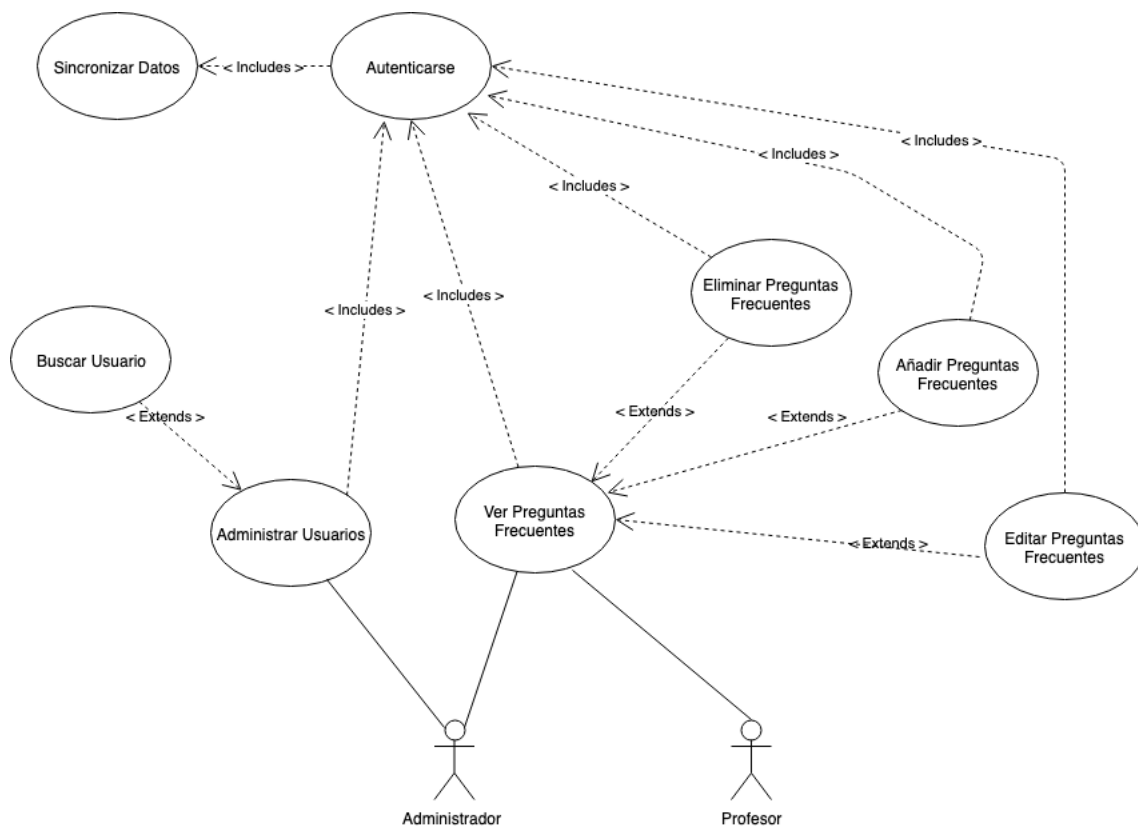


Fig. 7: Diagrama Casos de uso - Administración Usuarios y Preguntas Frecuentes

Caso de uso	Administrar Usuarios
Descripción	El usuario administra los roles de los usuarios
Actores	Cualquier usuario autenticado con un rol de administrador
Pre-Condición	1. Estar autenticado 2. Encontrarse en la pagina Administrar Usuarios
Flujo Normal	1. El usuario esta autenticado 2. El usuario selecciona en el menú la pagina Administrar Usuarios 3. Selecciona el usuario que desea haciendo clic en “ Editar ” 4. Asigna o revoca permisos 5. Guarda los cambios
Éxito	El rol del usuario se actualiza

Tabla 14: Caso de uso – Administrar Usuarios

Caso de uso	Buscar Usuario
Descripción	El usuario administrador busca en la lista de usuarios registrados
Actores	Cualquier usuario autenticado con un rol de administrador
Pre-Condicion	<ol style="list-style-type: none"> 1. Estar autenticado 2. Encontrarse en la pagina Administrar Usuarios
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario esta autenticado 2. Selecciona en el menú la pagina Administrar Usuarios 3. Se sitúa en el buscador 4. Escribe al menos un carácter 5. Envía la solicitud de búsqueda haciendo clic en Buscar 6. Se muestra lista de usuarios o mensaje de no coincidencias
Éxito	Obtiene una lista de usuarios o mensaje de no coincidencias

Tabla 15: Caso de uso – Buscar Usuario

Caso de uso	Editar Preguntas Frecuentes
Descripción	El usuario edita las preguntas frecuentes
Actores	Cualquier usuario autenticado con un rol de administrador o profesor
Pre-Condicion	<ol style="list-style-type: none"> 1. Estar autenticado 2. Encontrarse en la pagina Preguntas Frecuentes
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario esta autenticado 2. Selecciona en el menú la pagina Preguntas Frecuentes 3. Selecciona una pregunta frecuente y hace clic en Editar 4. Realiza los cambios y hace clic en Guardar 5. Se muestra la pregunta frecuente actualizada
Éxito	Se muestra la pregunta frecuente actualizada

Tabla 16: Caso de uso – Editar Preguntas Frecuentes

Caso de uso	Añadir Preguntas Frecuentes
Descripción	El usuario añade preguntas frecuentes
Actores	Cualquier usuario autenticado con un rol de administrador o profesor
Pre-Condicion	<ol style="list-style-type: none"> 1. Estar autenticado 2. Encontrarse en la pagina Preguntas Frecuentes
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario esta autenticado 2. Selecciona en el menú la pagina Preguntas Frecuentes 3. Hace clic en Añadir Pregunta Frecuente 4. Se redirige a la pagina para añadir una pregunta frecuente 5. Cumplimenta los campos que considera oportunos 6. Clic en Publicar pregunta frecuente
Éxito	La pregunta frecuente se publicada en la lista de preguntas frecuentes

Tabla 17: Caso de uso – Añadir Preguntas Frecuentes

Caso de uso	Eliminar Preguntas Frecuentes
Descripción	El usuario elimina preguntas frecuentes
Actores	Cualquier usuario autenticado con un rol de administrador o profesor
Pre-Condicion	1. Estar autenticado 2. Encontrarse en la pagina Preguntas Frecuentes
Flujo Normal	1. El usuario esta autenticado 2. El usuario selecciona la pagina Preguntas Frecuentes 3. Hace Clic en Borrar Pregunta Frecuente 4. Se muestra un mensaje de confirmación 5. El usuario confirma o deniega la operación
Éxito	La pregunta es eliminada de la lista de Preguntas Frecuentes

Tabla 18: Caso de uso – Eliminar Preguntas Frecuentes

3.5. Mock-up

En este apartado se muestra el *mock-up* a partir del cual se creará la interfaz de usuario de la aplicación. Al igual que en el apartado anterior se explicarán los diferentes componentes que tiene la aplicación. En función del rol que tiene cada uno de los usuarios en la aplicación estas se diseñan siguiendo el mismo padrón para poder facilitar su comprensión.

a) Registro en la aplicación

En este primer *mock-up* podemos apreciar un menú que grafica la pagina de ayuda con las preguntas frecuentes y la pagina de contacto. Además, en la parte superior izquierda del menú el usuario tiene a su disposición dos botones de navegación que le llevan a la vista de registro o inicio de sesión.

≡	Ayuda (FAQ)	Contacto		Registro	Login
---	---------------	----------	--	----------	-------

Logo APP

Registro

Nombre

Email

Password

Repite Password

Registrar

Tienes una Cuenta?

TFG	Copyright	Dirección
-----	-----------	-----------

Fig. 8: *Mock-up* - Registro

b) Iniciar sesión en la aplicación

Para poder acceder a la aplicación al usuario se le ofrece dos alternativas:

- Mediante email y contraseña previamente registradas en la aplicación.
- “Login con Google” que le permite acceder a la aplicación a través del sistema de autenticación de con un correo de Gmail.

☰	Ayuda (FAQ)	Contacto		Registro	Login
<div>Logo APP</div> <div>Login</div> <div>Email</div> <div>Password</div> <div>Login</div> <div>Eres nuevo ?</div> <div>Login con Google</div>					
TFG	Copyright			Dirección	

Fig. 9: *Mock-up* - Inicio Sesión

c) Preguntas frecuentes

i. Vista de los alumnos y usuarios no registrados

En esta pantalla se ilustra una lista de preguntas frecuentes que podrá ser consultada por cualquier usuario.

☰	Ayuda (FAQ)	Contacto		Registro	Login
<div>¿Como puedo conseguir el video que habla de Pitagoras?</div> <div></div> <div>¿ Puedo visualizar los videos desde una tablet ?</div> <div></div>					
TFG	Copyright			Dirección	

Fig. 10: *Mock-up* - Preguntas Frecuentes – Vista alumnos y usuarios no registrados

ii. Vista para administrador y profesores

Los usuarios del tipo profesor o administrador podrán añadir, modificar o borrar las preguntas frecuentes de esta vista.

☰	Ayuda (FAQ)	Contacto		Registro	Login
---	---------------	----------	--	----------	-------

Añadir Pregunta Freceuyente

¿Como puedo conseguir el video que habla del Teorema Pitagoras?

Editar

Eliminar

¿ Puedo visualizar los videos desde una tablet ?

Editar

Eliminar

TFG	Copyright	Dirección
-----	-----------	-----------

Fig. 11: *Mock-up* - Preguntas Frecuentes – Vista administrador y profesores

d) Formulario de Contacto

La pantalla de contacto es sencilla. Simplemente requiere que todos los campos sean rellenados para realizar una consulta.

☰	Ayuda (FAQ)	Contacto		Registro	Login
---	---------------	----------	--	----------	-------

Nombre

Telefono

Email

Mensaje

Enviar

TFG	Copyright	Dirección
-----	-----------	-----------

Fig. 12: *Mock-up* - Formulario de Contacto

e) Lista de videos

i. Vista de los alumnos

En este mock-up se muestra la lista de videos de los alumnos y las funcionalidades implementadas para este tipo de perfil de usuario. El alumno podrá “Cargar mas videos” y dispone un enlace “Ver Video”.

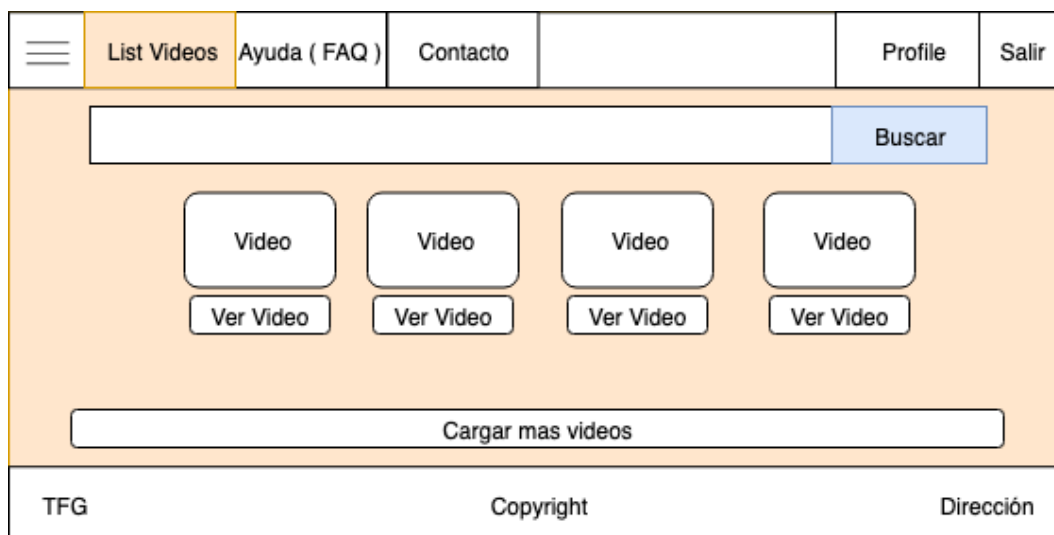


Fig. 13: *Mock-up* - Lista de Videos – Vista de los alumnos

ii. Vista de los profesores

Además de las funcionalidades que muestra la lista de videos de los alumnos, la de los profesores implementan las funcionalidades de añadir, editar video y borrar un video. En caso de que el profesor no sea el propietario de un video los botones de editar y borrar video estarán desactivados y se presentaran de otro color.

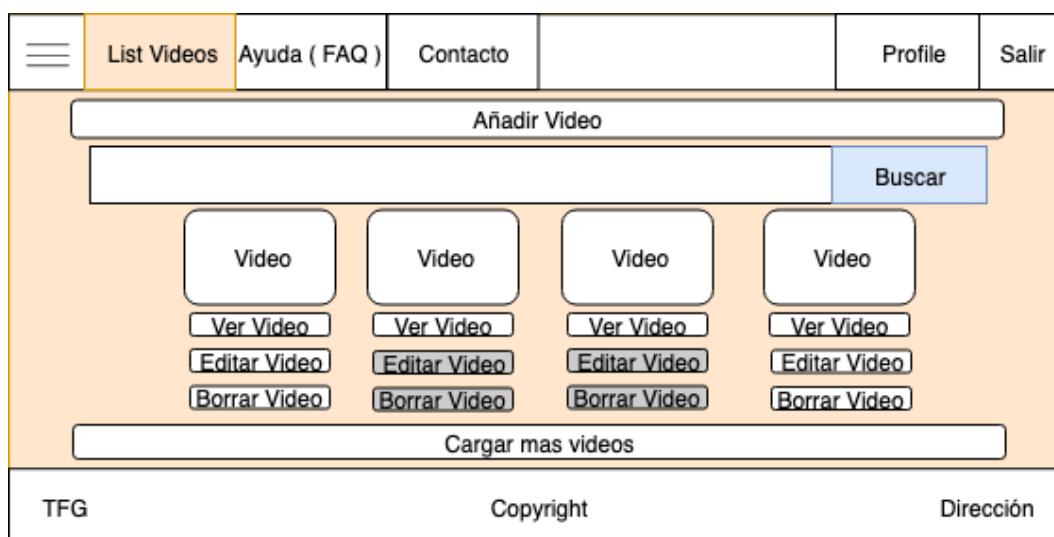


Fig. 14: *Mock-up* - Lista de Videos – Vista de los profesores

iii. Vista de los administradores

Al tratarse de un perfil administrador este usuario dispone de permisos suficiente para editar o borrar en caso oportunos cualquiera de los videos publicados en la plataforma.

The mock-up shows a web interface for administrators. At the top is a navigation bar with a hamburger menu icon, followed by links: 'List Videos' (highlighted in orange), 'Ayuda (FAQ)', 'Contacto', 'Administrar Usuarios', an empty space, 'Profile', and 'Salir'. Below the navigation bar is a large orange rectangular area. Inside this area, at the top, is a white input field labeled 'Añadir Video' with a blue 'Buscar' button to its right. Below this is a grid of four video entries. Each entry consists of a white box labeled 'Video', followed by three buttons: 'Ver Video', 'Editar Video', and 'Borrar Video'. At the bottom of the orange area is another white input field labeled 'Cargar mas videos'. Below the orange area is a white footer bar containing the text 'TFG', 'Copyright', and 'Dirección'.

Fig. 15: *Mock-up* - Lista de Videos – Vista de los administradores

f) Perfil de usuario

En esta pantalla se muestran la información que los usuarios visualizan de su perfil y la información que pueden modificar.

The mock-up shows a user profile page. The navigation bar at the top has a hamburger menu icon, followed by links: 'Videos', 'Ayuda (FAQ)', 'Contacto', an empty space, 'Profile' (highlighted in orange), and 'Salir'. The main content area has a light orange background. On the left side of this area is a circular placeholder labeled 'Photo' with an 'Editar' button below it. To the right of the photo are four white input fields stacked vertically, labeled 'Nombre', 'Email', 'Ultimo acceso', and 'Guardar Cambios'. To the right of the 'Nombre' field is an 'Editar' button. At the bottom of the orange area is a white footer bar containing the text 'TFG', 'Copyright', and 'Dirección'.

Fig. 16: *Mock-up* - Perfil de Usuario

g) Ver video

Volviendo a la imagen en el cual se representa el listado de videos, todos los usuarios que han iniciado sesión pueden utilizar la funcionalidad de “Ver Video”

que los llevara a esta pantalla en la cual podrán visualizar el contenido y realizar consultas sobre el mismo.

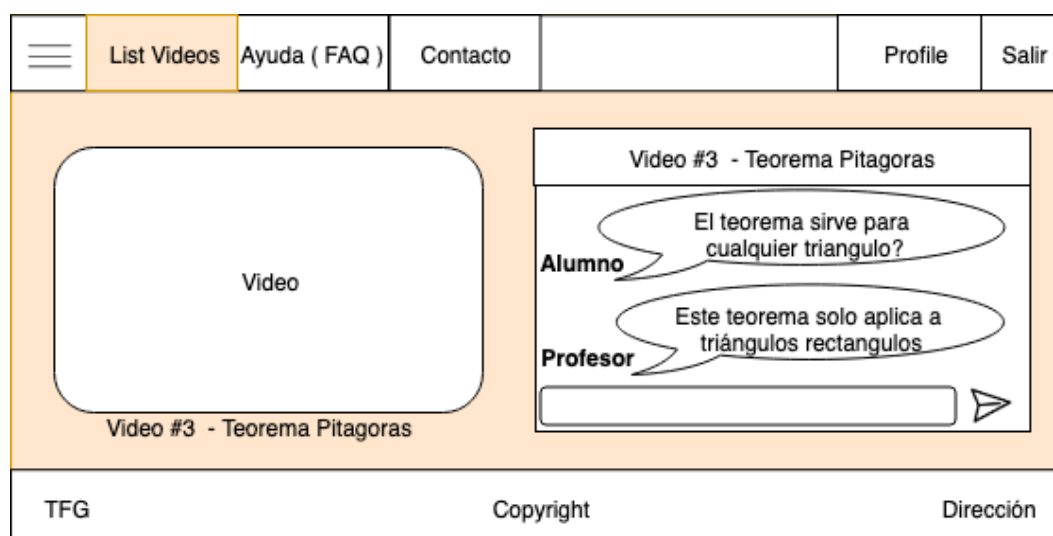


Fig. 17: *Mock-up* - Ver video y chat

h) Añadir video

En esta imagen se muestra la pantalla de un usuario del tipo profesor o administrador cuando añade un nuevo video.

Fig. 18: *Mock-up* - Añadir video

i) Editar video

En esta imagen se representa las opciones que tendrán los usuarios del tipo administrador o profesores cuando desean editar los metadatos de un video ya publicado.

	List Videos	Ayuda (FAQ)	Contacto		Profile	Salir
--	-------------	---------------	----------	--	---------	-------

Esta editando el video : **Teorema Pitagoras**

Cambiar Nombre

Cambiar Categoria

Cambiar Tags

Cambiar Descripción

TFG	Copyright	Dirección
-----	-----------	-----------

Fig. 19: *Mock-up* - Editar video

j) Eliminar video

Esta pantalla es sencilla y solo estará disponible para los profesores y el administrador. Cuando un usuario de este tipo quiere eliminar un video una ventana emergente espera confirmación del proceso.

	List Videos	Ayuda (FAQ)	Contacto		Profile	Salir
--	-------------	---------------	----------	--	---------	-------

Video

Ver Video

Editar Video

Borrar Video

Esta seguro que desea borrar el video :
Teorema Pitágoras?

Video

Ver Video

Editar Video

Borrar Video

TFG	Copyright	Dirección
-----	-----------	-----------

Fig. 20: *Mock-up* - Eliminar video

k) Administrar usuarios

Esta vista solo estará disponible para los usuarios administradores. Podrán realizar una búsqueda, asignar o revocar permisos a los usuarios o cambiar de pagina para que mas resultados sean mostrados.






	List Videos	Ayuda (FAQ)	Contacto	Administrar Usuarios		Profile	Salir
<input type="text"/>							Buscar
Nombre	UID	Email	Editor	Admin			
Pepe L.	9ZGdM7jT323	aaa@bbb.es			Edit / Save		
Lola P.	90tjZuFQGM7jT	bbb@gmail.com			Edit / Save		
Marc G.	87dsG09S73nsT	ccc@yahoo.es			Edit / Save		
Jose F.	tCEdQGHlsM7jT	ddd@personal.es			Edit / Save		
<div> 1 2 3 4 19 20 </div>							
TFG		Copyright				Dirección	

Fig. 21: *Mock-up* - Administrar usuarios

3.6. Back-end de la aplicación

Como ya se menciona en uno de los apartados anteriores la aplicación esta enfocada a utilizar diferentes servicios con tal de mantener de forma sincronizada los datos que almacena la aplicación. Para llevar a cabo esta tarea se ha decidido utilizar la plataforma Firebase que ofrece un back-end como servicio y que puede satisfacer las diferentes necesidades previamente expuestas. Entre las necesidades que se deben satisfacer se encuentra gestión de la autenticación, sincronización de datos en tiempo real y el almacenamiento de ficheros. Un ejemplo grafico de arquitectura de la aplicación a implementar utilizando este servicio se representa a continuación:

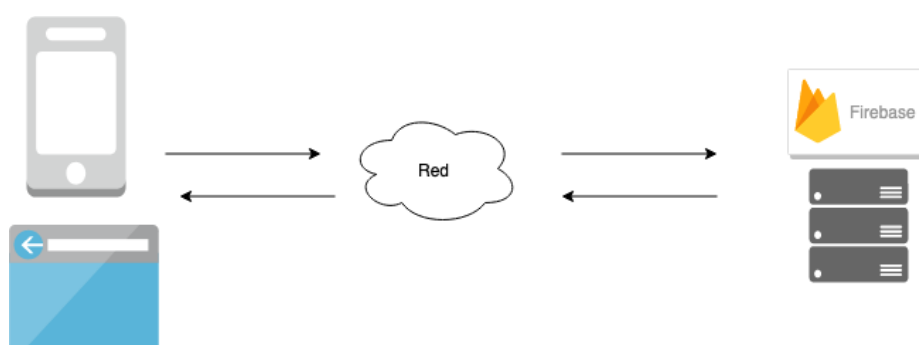


Fig. 22: Arquitectura comunicaciones con Firebase

Hoy en día palabras como aplicación móvil o la nube están creando un gran revuelo y cada día hay una nueva aplicación que sale al mercado para ofrecer un mejor rendimiento a sus usuarios. Las aplicaciones móviles necesitan tener un back-end que este conectado a Internet. Firebase es uno de ellos y cumple las funciones de BaaS (Back-end-as-a-Servide) siendo uno de los mas conocidos y popular en el mundo de los desarrolladores.

Según su propia definición, Firebase representa un conjunto de herramientas que facilitan la creación de aplicaciones de alta calidad. Esta plataforma fue adquirida por Google en el año 2016 y tiene como objetivo ayudar a los desarrolladores crear aplicaciones mejores. Por lo

tanto, Firebase facilita a crear la lógica de negocio y el mantenimiento del back-end de las aplicaciones.

Entre los principales servicios ofrecidos por Firebase encontramos una base de datos en tiempo real para guardar los datos en formato *json*. Otro servicio que pone a disposición esta plataforma es la autenticación. De esta forma se simplifica la forma de iniciar y gestionar la sesión de los usuarios. Firebase facilita la autenticación con proveedores como por ejemplo Google, Facebook, Twitter, mediante un correo y una contraseña, con un numero de teléfono móvil o incluso de forma anónima. A continuación, se muestra una lista de los servicios que ofrece esta plataforma.

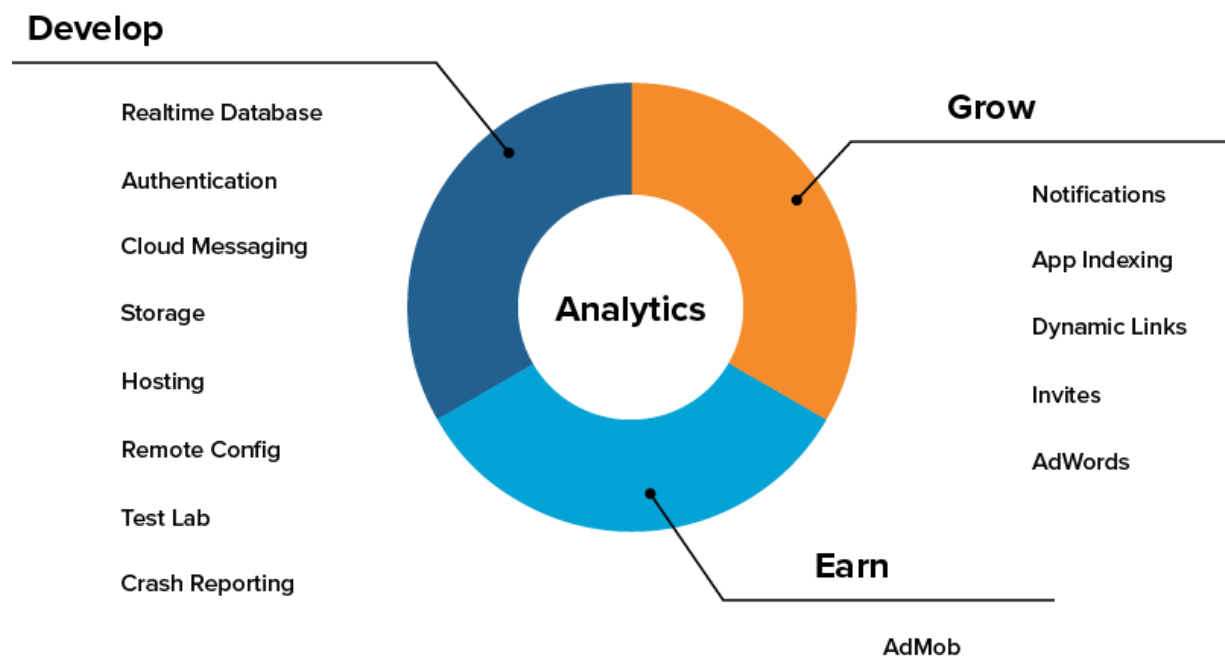


Imagen 16: Servicios ofrecidos por Firebasse [18]

Otros de los servicios utilizados para mostrar y almacenar el contenido video es la plataforma YouTube. Con la ayuda de YouTube se consigue que la aplicación se despreocupe del almacenamiento, codificación y distribución del contenido video. Para ello se ha utilizado la versión 3 de YouTube Data API, que permite incorporar todas las funciones que normalmente se ejecutan en el sitio web de YouTube¹².

De modo que una arquitectura completa del *back-end* de la aplicación utilizando los servicio comentados anteriormente quedaría plasmada en la siguiente figura:

¹² YouTube Data API – Documentación de la API <https://developers.google.com/youtube/v3/docs/>

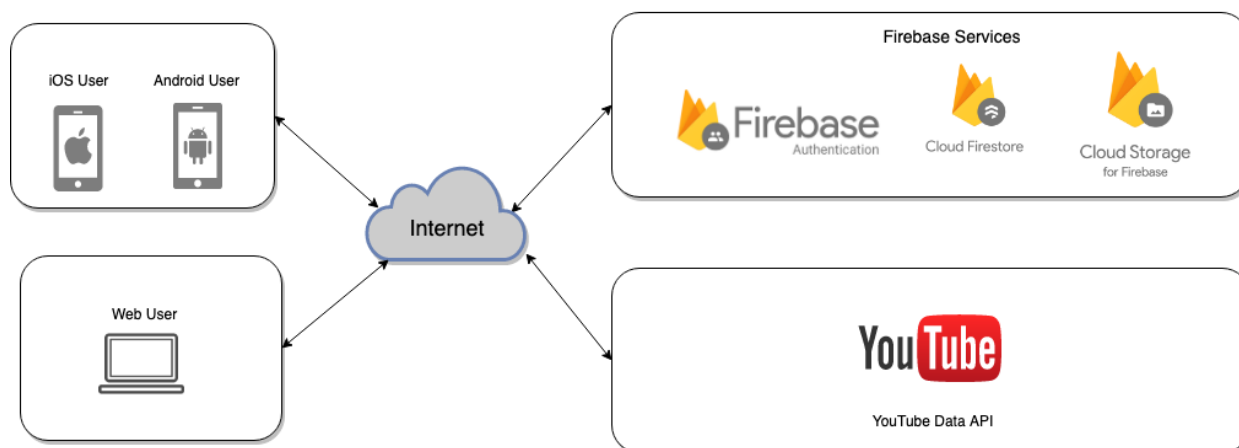


Fig. 23: Arquitectura de los servicios utilizados

3.7. Estimación de costes

Por lo que se refiere el calculo de los costes que implican el desarrollo y despliegue de esta aplicación se deberían tener en cuenta los siguientes servicios:

YouTube Data API.

En cuanto a este servicio no se aplica un coste implícito, según las condiciones de uso del servicio de YouTube [19]. En estas condiciones de uso se especifica que YouTube puede usar cuotas y restricciones de uso para asegurar que estos servicios de API se usan adecuadamente. En otras palabras, a cada nueva aplicación registrada se le asigna una cuota diaria con el fin de garantizar que este servicio es utilizado según lo previsto. En el caso de que se exceda el limite de la cuota para un servicio, los usuarios deben esperar hasta el día siguiente o el administrador de la aplicación puede solicitar, mediante un formulario, la extensión de esta cuota.

En cuanto a la cantidad de consultas API que se pueden realizar durante 24 horas, YouTube fija una cuota de 10.000 unidades por día. Con respecto al coste de las solicitudes que se realizan a la API este varía entre 3 unidades para listar un recurso y van hasta 1616 unidades para insertar un nuevo video. A simple vista si se realiza un promedio, la cuota asignada serviría para subir unos 5 videos diarios. Hay que tener en cuenta que dependiendo del uso que se le daría a la aplicación, esta cuota podría agotarse y ciertos usuarios podrían quedarse en estado de *starvation*.

Para concluir, la cuota ofrecida por YouTube es mas que suficiente para visualizar el contenido video, pero supone ciertas limitaciones a la hora de desarrollar y usar la subida de videos desde la aplicación. Como bien se comento anteriormente esta cuota puede ser ampliada si se cumplen todos los requerimientos de YouTube.

Servicios *Firebase*.

En cuanto los servicios de *Firebase* este ofrece tres modalidades de pago [20]. La primera modalidad es el “*Plan Spark*” que ofrece unos limites generosos para aficionados sin ningún cargo. La segunda opción ofrecida es el “*Plan Flame*” que ofrece por 25\$ mensuales una cantidad de recursos superior al anterior plan. Y por lo ultimo el tercer plan ofrecido es el “*Plan*

Blaze”. Este ultimo plan ofrece de una forma dinámica un pago por uso incluyendo de forma gratuita todos los beneficios del “*Plan Spark*” que se detallan a continuación.

Para evitar un aumento de los costes para el desarrollo, en este proyecto se elige el “*Plan Spark*”. Este plan incluye de forma gratuita los servicios de *Firebase* aplicando unos limites en cada uno de ellos.

En primer lugar, el servicio *Firebase Authentication*, se ofrece de forma gratuita e ilimitada incluyendo los servicios de autenticación que se proponen para implementar en los apartados anteriores con una excepción. En caso de que en un futuro se decida utilizar el servicio de autenticación a través de un numero de teléfono móvil, este plan contempla de forma gratuita 10.000 verificaciones de autenticaciones al mes. En caso de superar este limite, la aplicación debería pasar al “Plan Blaze” donde se aplicaría un coste de 0,06\$ por cada verificación.

Por otra parte, se encuentra el servicio de almacenamiento de base de datos también llamado *Cloud Firestore*. Este servicio ofrece una base de datos *NoSQL* flexible y escalable en la nube, permitiendo de esta forma almacenar y sincronizar los datos necesarios de la aplicación. Con el plan propuesto se obtiene de forma gratuita 1 Gb de almacenamiento de documentos con un ancho de banda de 10GB mensuales. Por otro lado, con este plan se establece un limite de las operaciones de escritura, lectura o eliminación de documentos diarios. Estos limites de operaciones se establecen en 20.000 operaciones diarias para la escritura o la eliminación y en 50.000 operaciones diarias para lectura. Vistos estos valores se decide que son mas que suficientes para realizar una primera aproximación del producto mínimo viable.

Por último, se utiliza el servicio de *Firebase Storage* y hay que tener en cuenta que a diferencia del anterior este permite almacenar ficheros. Con el plan anteriormente propuesto se consigue de forma gratuita 5 GB de almacenamiento. Esta cantidad de almacenamiento es mas que suficientes para almacenar las imágenes de los perfiles de usuario o de imágenes de las preguntas frecuentes. En cuanto a operaciones de archivos este servicio establece un limite de 20.000 subidas y 50.000 descargas diarias. En cuanto al volumen de datos descargados diarios se establece un limite de 1GB.

En resumen, como primera aproximación del proyecto, el “*Plan Spark*” de *Firebase* es mas que suficiente y permite implementar de forma gratuita todas las funcionalidades explicadas en los apartados anteriores.

Dominio web.

Por lo que se refiere al dominio web existe un amplio abanico de empresas que ofrecen registro de dominós web económicos. Para empezar, se realiza una comparativa entre diferentes proveedores y se encuentran unos precios atractivos ofrecidos por la compañía Piensa Solutions¹³. Para realizar el despliegue de la aplicación web se decide adquirir el dominio necesario a esta empresa por el precio de 1,21€.

¹³ Piensa Solutions – Empresa que ofrece registro de dominios y hosting
<https://www.piensasolutions.com/>

Datos de facturación:	
Marius Iacob	SAbadell
NIF: x8070284k	08206 Sabadell
ID: 171438	BARCELONA - ESPAÑA

Nº de factura:	Fecha de emisión:	Vencimiento:	Forma de pago:
2019B048850	13/04/2019	13/04/2019	TARJETA

Concepto	Importe	% Dto.	Importe total
Registro streamedu.es (13/04/2019-12/04/2020)	4.95	100% ⁽¹⁾	0.00 €
Protección de Dominio Registro .es streamedu.es (13/04/2019-12/04/2020)	3.95	100% ⁽²⁾	0.00 €
Registro .eu streamedu.eu (13/04/2019-12/04/2020)	4.95	79.8% ⁽³⁾	1.00 €
		Base Imponible	1.00 €
		IVA 21,00 %	0.21 €
		Total	1.21 €

Imagen 17: Factura registro dominio web [Anexo 1]

Administración del dominio web.

Con respeto a la gestión y administración del dominio web, la solución ofrecida por la empresa a la cual se contrato el dominio web es bastante limitada. Con este motivo se decide migrar la gestión del dominio web a la plataforma AWS. El servicio AWS utilizado en esta ocasión es el Amazon Route 53¹⁴ y que en pocas palabras es un administrador de DNS (Domain Name System).

Se debe tener en cuenta que, al realizar esta migración, el servicio de gestión de dominios de AWS implica un coste de 0,5€ mensuales. No obstante, este cambio puede parecer innecesario ya que implica un incremento a nivel de costes, pero tiene un retorno inmediato teniendo en cuenta los beneficios que ofrece esta plataforma y que algunos de ellos se explicaran a continuación.

Alojamiento de la aplicación web.

En cuanto al alojamiento de una aplicación web, hay que comenzar realizando un análisis entre las distintas opciones que se ofrecen hoy en el mercado. De modo que previamente se ha elegido el *framework Angular*, este ofrece la posibilidad de crear una aplicación web del tipo S.P.A. La idea central de una aplicación del tipo S.P.A. es que el *framework* genera los archivos estáticos necesarios para desplegar una pagina web estática. En pocas palabras la aplicación estará formado por un único archivo HTML, un archivo CSS, los *assets* y los archivos JavaScript necesarios.

Por tal motivo las soluciones del tipo: Servidor Compartido (Hosting), Servidor Virtual Privado (VPS) o el uso de un Servidor Dedicado quedarían descartadas debido a que no se

¹⁴ Amazon Route 53 – Servicio administración de DNS: <https://aws.amazon.com/es/route53/>

realizaría un uso óptimo de los recursos necesarios. Hay que destacar que estos tipos de soluciones no llegarían a cumplir ciertos requisitos de la aplicación como, por ejemplo: la alta disponibilidad, la escalabilidad o elasticidad en términos de pago por uso.

Desde hace varios años nuevos servicios como, por ejemplo: GitHub Pages, GitLab Pages, Google Firebase, Heroku, Google Drive, Dropbox o AWS S3 entre otros, ofrecen la posibilidad de alojar una página web estática.

Gracias a la experiencia previa utilizando los servicios de AWS se decide optar por el servicio *Amazon Simple Storage Service* (AWS S3). S3 es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento. Este servicio está diseñado para ofrecer una durabilidad del 99,999999999% y como consecuencia almacena datos de millones de aplicaciones en todo el mundo [21].

En cuanto a los costes asociados a este servicio para una aplicación de estas características no supera los 0,1 € mensuales. Para este cálculo se toma en cuenta un almacenamiento de un 1Gb de datos y 100.000 solicitudes mensuales del tipo GET que resultan más que suficientes para el propósito de la aplicación.

Certificado SSL.

Con respecto al certificado SSL (Secure Sockets Layer) existen distintas autoridades certificadoras que ofrecen certificados a precios que van por ejemplo desde los 6€/anuales ofrecidos por empresas como Comodo hasta los 349€/años por empresas como DigiCert.

Uno de los beneficios de migrar la gestión de DNS a AWS es que esta plataforma ofrece certificados SSL de tipo wildcard de forma gratuita para usarlos en sus servicios. Para ello se decide usar el servicio AWS Certificate Manager¹⁵ ya que dispone de una fácil integración con la CDN (Content Delivery Network) de AWS que se explicará a continuación.

CDN.

Con respecto a la red de distribución de contenido se pueden encontrar distintas soluciones que ofrecen este tipo de servicio. Este servicio es una red de servidores que contiene copias de datos. Estos servidores están colocados de forma estratégica en varios puntos de la red con el propósito de maximizar el ancho de banda y de esta forma facilitar el acceso a los datos de clientes por la red [22].

También se deben enumerar otros servicios similares como por ejemplo CloudFlare, Akamai o MaxCDN entre otros. Todas estas soluciones tienen un mismo propósito, pero se decide utilizar la solución de AWS en gran parte gracias a la amplia presencia que tiene a nivel mundial.

Se decide utilizar el servicio de AWS CloudFront para crear esta red de distribución de contenido porque es una solución rápida y global. Además, ofrece una alta seguridad en el borde de la red y de aplicación. Por otra parte, ofrece una alta capacidad de propagación de los cambios y una fácil integración con los servicios de AWS.

¹⁵ AWS Certificate Manager – Servicio administrador de certificados SSL
<https://aws.amazon.com/es/certificate-manager/>

En cuanto al tema costes, este servicio incluye en la capa gratuita [23], durante los primeros 12 meses, incluye 50GB de transferencia de datos mensuales. Además, ofrece 2 millones de solicitudes HTTP/S mensuales de forma gratuita y que son mas que suficientes para el propósito de este proyecto. Pasados los primeros 12 meses esta solución pasara a ser de pago por uso, aplicando unos costes de 0,085 € por cada GB de datos transferidos y 0,009 € por cada 10.000 solicitudes HTTP/S realizadas.

Coste que involucran la herramienta de automatización de despliegue de software.

En cuanto a la herramienta de automatización se ha escogido una solución Jenkins. También se debe tener en cuenta que existe una amplia cantidad de alternativas a este servicio. Entre algunas de estas alternativas se encuentra Codeship¹⁶ o CircleCI¹⁷, pero teniendo en cuenta la larga trayectoria y comunidad de software libre que tiene Jenkins, se decide escoger esta solución.

Para desplegar la herramienta Jenkins se necesita un servidor que disponga de 2 Gb de memoria RAM. Para ello se utilizará el servicio AWS EC2¹⁸ que permite aumentar o disminuir la capacidad de computo necesaria en cuestión de minutos. En este caso se decide utilizar una instancia de computo del tipo t3.small¹⁹ con 2 CPU virtuales y 2 GiB de memoria. El coste mensual de este servicio es de 15,23 \$ mensuales²⁰. Considerando la flexibilidad que ofrece AWS a la hora de contratar sus servicios, este servidor podría detenerse si no se utiliza y de esta forma se puede reducir de forma considerable el coste mensual asociado a este servidor.

Despliegue aplicación iOS

Otro coste para tener en cuenta, si se decide en un futuro publicar la aplicación en la tienda de App Store, es la suscripción que se ha de realizar en el programa de desarrolladores de Apple. El coste relacionado a este servicio es de 99€ anuales y además de permitir la publicación de aplicación en el AppStore, brinda acceso a: características de *testing* para dispositivos iOS, a versiones beta del sistema operativo iOS y a los *SDKs* que ofrece la compañía.

Despliegue aplicación Android

Por ultimo, si en futuro se decide publicar la aplicación en la tienda de Google Play, se debe realizar una suscripción que, a diferencia de la tienda de Apple, se debe abonar un pago único de 25\$.

Coste total

Vale la pena decir que, en el caso de varios de los servicios implementados, los precios publicados por estas compañías son en dólares. Para evitar la conversión de dólares a euros se ha hecho una estimación al alza y todos estos costes se aproximan a euros.

¹⁶ Codeship – Herramienta de automatización despliegue software: <https://codeship.com/>

¹⁷ CircleCI – Herramienta de automatización despliegue software: <https://circleci.com/>

¹⁸ AWS EC2 – Servicios de computo ofrecido por AWS <https://aws.amazon.com/es/ec2/>

¹⁹ t3.small – Tipo de instancia de computo ofrecida por AWS EC2 :

<https://aws.amazon.com/es/ec2/instance-types/>

²⁰ [Coste servidor Jenkins](#)

Servicio	Coste
YouTube Data API	Gratis
Servicio Firebase – Plan Spark	Gratis
Dominio web	1,21 € /anuales* - pago unico
Administración del dominio web	6 € /anuales
Alojamiento web	1,2 € / anuales
Certificado SSL	Gratis
CDN	Gratis**
Herramienta Jenkins	91,38 € / anuales ***
Despliegue aplicación iOS	99 € /anuales
Despliegue aplicación Android	25 € / pago único
Coste anual	223,79 €
Coste asociado a 6 meses de TFG	~ 112 €

Tabla 19: Costes asociados al despliegue de la aplicación [Anexo 1]

* Después del primer año este coste aumenta a 9,99 €

** Gratis el primer año según se explico mas arriba

*** Se estima un funcionamiento de servidor durante 12 h/ día

3.8. Tratamiento de los datos personales de los usuarios LOPD

En relación con la Ley Orgánica de Protección de Datos (LOPD), esta legislación también aplica al desarrollo de las aplicaciones web y móviles. Dada la gran complejidad e importancia, a continuación, se explican los aspectos mas relevantes que se deben cumplir en relación con los datos y contenidos publicados en la aplicación. Con este motivo es importante detectar que datos sensibles del usuario almacena la aplicación y con que propósito. Con este motivo la LOPD requiere cumplir las obligaciones de informar con que finalidad se guardan y utilizan estos datos. Además, se debe obtener el consentimiento del usuario para poder utilizar su identidad para publicar mensajes y el contenido video en la plataforma YouTube en su nombre.

Entre los datos que almacena la aplicación se encuentran: el nombre de usuario, la dirección de correo electrónico, la foto de perfil de usuario, la contraseña, los mensajes y los videos publicados en la cuenta de YouTube. Para ello se debe cumplir la protección de estos datos de carácter personal y como consecuencia a nivel de aplicación, las comunicaciones serán cifradas durante el transito y no serán cedidas a terceros para otros propósitos.

Por otra parte, las autenticaciones realizadas con cuentas de Gmail y las llamadas que se realizaran a los diferentes servicios implementados, se realizan utilizando el protocolo de autorización OAuth 2.0. De esta forma la aplicación accede de forma segura a ciertos datos del usuario. El acceso a estos datos se realiza mediante un *token* genérico, una vez obtenida la autorización pertinente por parte del usuario.

4. IMPLEMENTACIÓN

4.1. Configuración inicial

a) Entorno de desarrollo utilizado

Antes de comenzar con la implementación de la aplicación, se ha decidido utilizar Visual Studio Code²¹ como editor de código debido a sus amplias herramientas que aumentan la productividad a la hora de escribir y compilar este tipo de proyecto. Visual Studio Code también proporciona diferentes *plugins* y documentación que permite la compilación, detección y corrección de errores en tiempo de escritura. De esta forma se están siguiendo las buenas practicas definidas en los archivos de configuración de un proyecto en Angular.

Entre las diferentes extensiones utilizadas para el desarrollo se ha configurado Angular 7 Snippets, Bootstrap 4, Font Awesome, VS Code JavaScript (ES6) snippets, Prettier formatter, TSLint y TypeScript Hero entre otros.

Antes de dar el paso a la creación y definición de la estructura de un proyecto en Angular se necesitan realizar una serie de configuraciones. Siguiendo la documentación oficial de Angular, observamos que se deben cumplir una serie de prerequisites. Se debe tener instalada la versión 8.x o 10.x de NodeJS²² y el npm²³ (Node Packet Manager). A continuación, se comprueba que se cumplen estos requisitos:

```
>> $ node -v → v10.15.3  
>> $ npm -v → 6.4.1
```

b) Estructura de un proyecto en Angular

Con respecto a la documentación del *framework* Angular esta proporciona un tutorial donde se indican los primeros pasos para crear una aplicación. Para ello se debe tener instalado el Angular CLI y se realiza a través del gestor de paquetes npm.

```
>>$ npm install -g @angular/cli
```

Después de realizar la instalación del Angular CLI se procede a la creación de una estructura básica mediante el comando:

```
>>$ ng new StreamEDU
```

Una vez finalizado este proceso en el directorio de trabajo se obtiene la siguiente estructura de ficheros y directorios:

²¹ Visual Studio Code – Editor de código: <https://code.visualstudio.com/>

²² NodeJS – Entorno de ejecución para JavaScript: <https://nodejs.org/es/>

²³ npm – Gestor de paquetes de Javascript de NodeJS: <https://www.npmjs.com/>

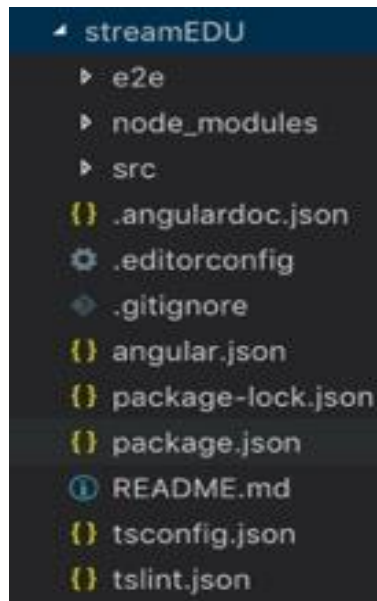


Imagen 18: Estructura de directorios y ficheros del proyecto

En primer lugar, en la anterior imagen se observa la estructura básica de ficheros y carpetas de un proyecto en Angular. Varias de estas carpetas incluyen una multitud de ficheros, pero no todos forman parte del código de la aplicación. Gran parte de estos ficheros y carpetas son necesarias para crear la infraestructura básica del proyecto, que el NodeJS necesita para gestionar el desarrollo de la aplicación.

El principio de la anterior imagen se encuentra el directorio *e2e*. Este directorio está destinado a realizar las pruebas de comunicación *end to end*. El contenido de este directorio sirve para realizar pruebas unitarias y de integración de los componentes forma automática. El alcance de este proyecto no contempla la realización de dichas pruebas y por lo tanto el contenido de esta carpeta no sería necesario para el desarrollo.

Después en la misma imagen se encuentra la carpeta *node_modules*. Esta carpeta incluye todos los módulos necesarios que la consola de Angular instala para facilitar el desarrollo. Entre los módulos instalados destaca el módulo *reload*, que básicamente es un servidor web que escucha los cambios que se realiza a lo largo del desarrollo para mostrar los cambios de forma casi instantánea por la pantalla. El contenido de esta carpeta es pesado y contiene una infinidad de paquetes. A la hora de realizar la compilación de la aplicación, el *framework* Angular se queda solo con las funcionalidades necesarias.

Una buena práctica es ignorar el contenido de la carpeta de *node_modules* a la hora de guardar los cambios en un repositorio de control de versiones. El contenido de esta carpeta puede ser recuperado utilizando el comando *npm install*. Con este comando el gestor de paquetes *npm* realiza una lectura del fichero *package.json* y restablece el contenido de esta carpeta. Una vez finalizada la instalación genera el fichero *package-lock.json* donde deja indicado los módulos que se encuentran instalados en el directorio del proyecto.

Por otra lado en la imagen anterior se encuentra el archivo *angular.json* sobre el cual se especificó su importancia previamente. Como bien se comentó este archivo indica al *framework* Angular el comportamiento y la forma en la cual se muestra la aplicación. Además, en este fichero se puede indicar la ubicación de los archivos externos añadidos a la aplicación como, por ejemplo:

los *assets*, las hojas de estilo y los scripts. En el momento de crear la aplicación final, el *framework* toma estos archivos y los unifica para que puedan ser utilizados en la aplicación.

En la carpeta SRC se almacena la lógica de los componentes que se desarrollan. En la siguiente imagen se muestra el contenido de esta carpeta y se ira explicando a continuación:

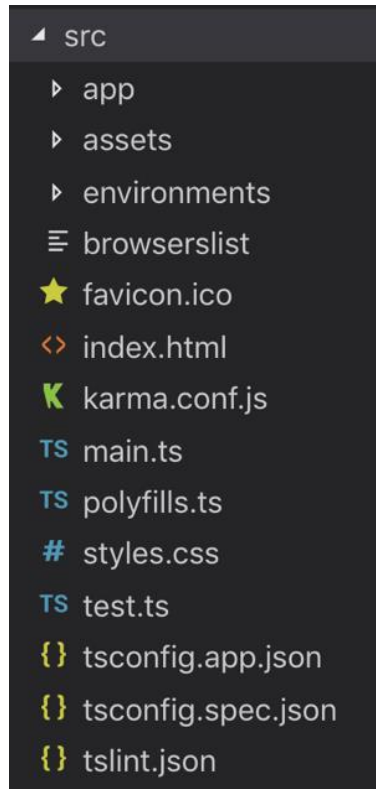


Imagen 19: Estructura de directorios y ficheros de la carpeta *src* del proyecto

Por un lado, en la anterior imagen se encuentran las carpetas *app*, *assets* y *environments*. Estas carpetas contienen la estructura básica del proyecto que la consola de Angular genera por defecto. También se puede apreciar en la anterior imagen la presencia del archivo *index.html*. Al tratarse de una aplicación del tipo S.P.A este es el archivo principal y sirve para mostrar cualquier ruta de la aplicación. Además, en esta carpeta *src* se encuentra el archivo *styles.css* que almacena las definiciones generales de la hoja de estilos en cascada.

4.2. Patrones de arquitectura de software

Modelo-Vista-Controlado (MVC)

El Modelo-Vista-Controlado (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos. Se trata de un modelo maduro que con los años ha demostrado su validez en todo tipo de aplicaciones. Es un patrón de diseño utilizado por una multitud de lenguajes y plataformas de desarrollo [24]. Este patrón de desarrollo de software ha sido diseñado con el propósito de reducir el esfuerzo de programación.

De esta forma este patrón de diseño permite que la implementación se realice de forma modular. Además, con las vistas la información que se muestra siempre esta actualizada. Por otra parte, cualquier modificación de la cantidad de métodos o datos contenidos implica solo una modificación en el modelo y las interfaces de las vistas [25].

Ahora bien, el Modelo contiene la representación de los datos que se trabajan en el sistema, la lógica de negocio y todos los mecanismos para asegurar la persistencia de los datos. Por otro lado, la Vista, o interfaz de usuario, es la encargada de representar la información que se envía al cliente y también implementa los mecanismos de interacción. Por ultimo el Controlador es el intermediario que gestiona el flujo de la información entre el modelo y la vista.

No obstante, además del patrón MVC existen otros patrones de diseño de arquitectura. Vale la pena decir que la mayoría de esto patrones de diseño de arquitectura son similares al MVC y tratan de mejorar la interfaz del usuario y la separación de responsabilidades. En definitiva, algunas variaciones del patrón de diseño MVC serian: el Modelo Vista Presentador (MVP) o el Modelo Vista VistaModelo (MVVM).

Modelo Vista Presentador (MVP)

El principal objetivo del Modelo Vista Presentador es proveer una implementación mas clara de la conexión entre el modelo de aplicación y la vista. Este modelo surge para ayudar a realizar pruebas automáticas de la interfaz grafica. Para ello con este modelo se trata codificar una interfaz de usuario lo mas simple posible, teniendo de esta forma la menor cantidad de código posible, para que en definitiva no merezca la pena probarla. En esta ocasión toda la lógica de la interfaz de usuario se realiza en una clase aparte, que es conocida como Presentador. La idea de esta clase es no depender de la interfaz grafica y de esta forma se facilita ejecución de pruebas sobre la clase de Presentador [26].

Modelo Vista VistaModelo (MVVM)

Este modelo se caracteriza por separar los datos de la aplicación de la interfaz de usuario, pero a la vez controlar manualmente los cambios en la vista o en los datos. Estos se actualizan automáticamente cuando sucede un cambio en ellos. Por ejemplo, si la vista actualiza un dato que esta presentado, se actualiza el modelo automáticamente y viceversa [27].

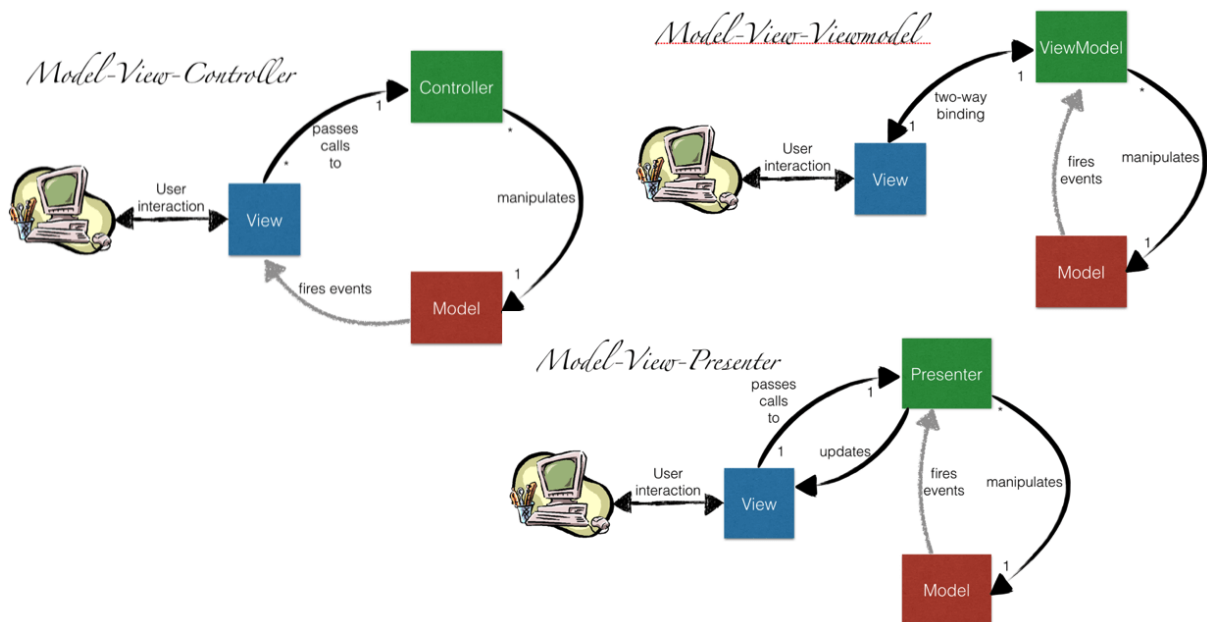


Imagen 20: Representación de las arquitecturas: MVC, MVP y MVVM [28]

a) El MVVM en Angular

A partir de este patrón, el *framework* Angular genera el paradigma usado en el entorno de desarrollo de aplicaciones del tipo SPA. Este patrón de diseño ofrece una abstracción de vista, que contiene el estado de la vista y su comportamiento. En comparación con el MVC, en este patrón de diseño, el controlador es reemplazado por el Vista Modelo. Las principales diferencias entre estos dos componentes son la dirección de dependencia entre la Vista de un lado y el Controlador o ViewModel en el otro lado. A continuación, se realiza una explicación del uso de este patrón de diseño con un ejemplo práctico.

El Modelo

Un modelo es un contenedor que define la estructura de una entidad, siguiendo el principio orientado a objetos. En Angular por norma general se usan clase o interfaces para definir el modelo. En definitiva, los modelos son simplemente objetos de JavaScript. Un claro ejemplo de modelo aplicado en el desarrollo de la aplicación sería la interfaz mensaje:

```
export interface Mensaje {
  nombre: string;
  mensaje: string;
  fecha: number;
  uid: string;
  videoId: string;
}
```

Hay que tener en cuenta que los modelos no deben incluir ninguna lógica y por lo tanto la lógica de negocio debe estar definida en la ViewModel.

Vista

La vista es la capa visual de la aplicación e incluye todos los elementos visuales que se muestran en la pantalla del usuario. En el caso del *framework* Angular la vista se define en una plantilla o *template*. Estos *templates* son escritos en HTML y CSS, pero también contienen elementos específicos del *framework* Angular. Los *templates* están conectados a la VistaModelo a través de enlaces de datos o *data-binding*. Los *templates* siempre forman parte de un componente.

VistaModelo

La VistaModelo es una parte abstracta de la Vista, que se encarga de administrar los modelos y las propiedades que se muestran por la pantalla. De esta forma contiene toda la lógica de la página. La Vista y VistaModelo a menudo están conectadas a través del *data-binding*. Esto significa que todos los cambios en la VistaModelo, como por ejemplo el cambio de una propiedad, se aplican de forma instantánea en la Vista, cambiando de esta forma por ejemplo el texto que la Vista muestra por pantalla. En el *framework* Angular, la VistaModelo está representada por la parte TypeScript de un componente. De modo que un componente de Angular está formado por dos partes: la plantilla o Vista, formada por los archivos HTML y CSS, y por la clase o VistaModelo.

Los componentes en Angular

Como se referencio anteriormente un componente básico en Angular consiste en dos partes. La primera parte consta de la definición del componente. Esta definición se realiza con la ayuda del decorador `@Component({})`. En la definición de este decorador se puede especificar las propiedades de este componente como por ejemplo un selector. De esta forma Angular permite añadir este componente a través de una etiqueta HTML. Por razón de no complicar estas definiciones el siguiente decorador define la vista principal de la aplicación.

```
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.css']  
})
```

Esta vista puede definirse agregando estructura de código HTML en línea o usando, como en este caso, una ruta que especifica la ubicación del archivo HTML. Por norma general se recomienda utilizar la segunda opción debido a que se mejora la legibilidad del código y una clara separación de estos elementos.

La segunda parte de un componente, utilizando el *framework* Angular, es la clase que permite implementar la lógica. Esta clase se encuentra en el mismo archivo justo debajo de la anterior definición. Hay que destacar que se trata de una clase TypeScript básica donde se puede implementar cualquier lógica de negocio. A continuación, se muestra un ejemplo que ilustra esta definición:

```
export class AppComponent {  
  constructor() { }  
}
```


Para poder mostrar los componentes del *framework* Angular en la pantalla, este debe ser insertado en una vista HTML que ya se muestra por pantalla. Como bien se comento anteriormente Angular permite crear aplicaciones del tipo SPA que contienen un único archivo HTML. De esta forma haciendo referencia a la estructura de un proyecto en Angular, este genera un archivo index.html y es este el archivo donde se debe añadir la etiqueta definida en el selector. La definición de esta etiqueta se realiza de la siguiente forma:

```
<app-root></app-root>
```

Teniendo en cuenta su definición, solo queda por añadir esta etiqueta en el cuerpo del archivo HTML principal. A continuación se muestra un ejemplo donde esta etiqueta principal es añadida al archivo index.html :

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Streamedu</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="/assets/logo_tfg.png">
</head>
<body >
  <app-root></app-root>
</body>
</html>
```

Hay que tener en cuenta que de forma análoga los componentes que se definen en la aplicación pueden añadirse al componente principal. De esta forma Angular permite un diseño modular aplicando el patrón de diseño MVVM.

4.3. Configuración de los servicios de back-end

Para comenzar a utilizar los distintos servicios comentados anteriormente, las APIs de YouTube y Firebase requieren de una serie de configuraciones previas. A continuación, se explican como se han llevado a cabo estas configuraciones e implementaciones en el proyecto.

a) YouTube Data API

En lo que la YouTube Data API atañe, cada una de las solicitudes que se realizan a esta API requieren de un *token* de Auth2.0²⁴. Para poder conseguir este tipo de clave de acceso, previamente se debe disponer de una cuenta de Gmail. Siguiendo la documentación oficial de esta API se debe acceder a la consola de API de Google conseguir este *token*. Hay que destacar que ambos servicios, YouTube y Firebase, son propiedad de la compañía Google y que, al crear un nuevo proyecto en el panel de administración, este será visible en ambos servicios que se están implementando.

²⁴ OAuth 2.0 – Protocolo de autorización <https://oauth.net/2/>

De modo que se debe crear primero un nuevo proyecto en esta consola de administración de APIs de Google. Es en este punto es donde se define el proyecto StreamEDU en la consola de administración. Después se habilita el uso de la ultima versión de la YouTube Data API y en el panel de administración del proyecto se facilita la creación de las credenciales necesarias para utilizar las funcionalidades de esta API.

Por último, estas credenciales deben configurarse en el proyecto Angular. Siguiendo las buenas practicas del *framework*, estas credenciales deben ser guardadas en fichero *enviroments.ts* dentro de la carpeta *enviroments* del proyecto. Además del fichero *enviroments.ts* en la misma carpeta, el *framework* proporciona otro fichero llamado *enviroments.prod.ts* . Como puede apreciarse a continuación el contenido de los ficheros es prácticamente idéntico y queda a disposición de Angular la administración de estos en el momento de realizar la compilación del código.

```
// environment.ts
export const environment = {
  production: false
};
//environment.prod.ts
export const environment = {
  production: true
};
```

Para poder utilizar estas credenciales en función del entorno utilizado, estas deben ser importadas en los componentes que lo requieren. A continuación, siguiendo la documentación de YouTube se muestra un ejemplo de un servicio en Angular que devuelve una lista de videos asociados a un canal de YouTube.

```

import { Injectable } from '@angular/core';
import { map } from 'rxjs/operators';
import { environment } from './../../environments/environment';
import { HttpClient, HttpParams } from '@angular/common/http';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class YouTubeService {
  private apiKey: string;
  private urlYoutube: string;
  private playList: string;
  private nextPageToken: string ;
  private paramsGetVideos = new HttpParams();

  constructor( public http: HttpClient ) {
    this.apiKey = environment.youtube.key;
    this.urlYoutube = environment.youtube.url;
    this.playList = environment.youtube.playList;
  }

  getVideos() {
    this.paramsGetVideos = this.paramsGetVideos.set('part', 'snippet');
    this.paramsGetVideos = this.paramsGetVideos.set('maxResults', '1');
    this.paramsGetVideos = this.paramsGetVideos.set('playlistId', this.playList);
    this.paramsGetVideos = this.paramsGetVideos.set('key', this.apiKey);

    if (this.nextPageToken) {
      this.paramsGetVideos = this.paramsGetVideos.set('pageToken',this.nextPageToken);}
    else {this.paramsGetVideos = this.paramsGetVideos.set('pageToken', "");}
    return this.http.get(`${this.urlYoutube}/playlistItems`, { params: this.paramsGetVideos })
      .pipe(map((data: any) => {
        this.nextPageToken = data.nextPageToken;
        let videos: any[] = [];
        for (let video of data.items) {
          let snippet = video.snippet;
          videos.push(snippet);
        }
        return videos;
      })))
  }
}

```

b) FireBase

Por lo que se refiere a FireBase, este ofrece una librería llamada AngularFire²⁵ que facilita la implementación de los servicios en Angular. Antes de entrar en detalles sobre esta librería, al igual que en el caso de los servicios de la API de YouTube se debe disponer de una cuenta de Google con la cual se debe dirigir a la Consola de Administración de Firebase²⁶ para registrar un nuevo proyecto.

Como bien se comento previamente, en el momento de añadir un nuevo proyecto de Firebase se selecciona el proyecto StreamEDU creado para la API de YouTube. Una vez creado el proyecto se deben obtener las credenciales necesarias para que la aplicación pueda usar los servicios ofrecidos por Firebase. Estas credenciales se deben añadir de forma similar a los ficheros *enviroments.ts* o *enviroments.prod.ts* . De forma similar a continuación se presenta un ejemplo de algunos métodos del servicio de Autenticación utilizando la librería AngularFire.

```
import { Injectable } from '@angular/core';
import { map } from 'rxjs/operators';
import { AngularFireAuth } from '@angular/fire/auth';
import { auth } from 'firebase/app';

@Injectable({ providedIn: 'root' })

export class AuthService {
  public usuario: any = {};
  constructor(public afsAuth: AngularFireAuth) {
    this.afsAuth.authState.subscribe(user => {
      if (!user) { return; }
      if (user.displayName) { this.usuario.nombre = user.displayName; }
      else { this.usuario.nombre = user.providerData[0].displayName; }
      this.usuario.uid = user.uid;
    })
  }
  loginGoogleUser() {
    return this.afsAuth.auth.signInWithPopup(new auth.GoogleAuthProvider())
      .then(credentials => this.updateUserData(credentials.user))
  }
  logoutUser() {
    this.usuario = {};
    return this.afsAuth.auth.signOut();
  }
  isAuth() { return this.afsAuth.authState.pipe(map(auth => auth)); }
}
```

²⁵ Librería AngularFire – Documentación: <https://github.com/angular/angularfire2>

²⁶ Consola de Administración de Firebase – Documentación: <https://console.firebase.google.com/u/0/>

4.4. Despliegue de la aplicación

Otro punto por tratar en esta ocasión es el despliegue de la aplicación y a continuación se detallarán los pasos seguidos con el fin de conseguir este objetivo. Se comenzará explicando como se adquirió el dominio web y la migración de la gestión del dominio a la plataforma AWS. Después se explicarán las configuraciones que se han llevado a cabo para alojar la aplicación web en el servicio AWS S3. Posteriormente se habla de los pasos seguidos para llevar a cabo la integración de la CDN con el servicio AWS S3 y el uso del certificado SSL ofrecido por el servicio AWS Certificate Manager. Para concluir se compila y se despliega de forma manual el contenido de la aplicación web diseñada.

a) Gestión del dominio web

Como se comento previamente se adquirió el dominio web a la plataforma de Piensa Solutions. Para ello previamente se comprueba que el dominio que se desea contratar esta disponible y se procede a realizar el registro en la plataforma y la contratación. Una vez finalizado el proceso de contratación se accede al panel de administración y se realiza la migración de la administración de los dominios a la plataforma AWS.

SERVICIO *	FECHA ALTA	FECHA RENOVACIÓN	ACCESOS
streamedu.es Dominio	15/04/2019	15/04/2020	 
streamedu.es Parking	15/04/2019		 
streamedu.eu Dominio	13/04/2019	13/04/2020	 
streamedu.eu Parking	13/04/2019		 

Imagen 20: Listado de dominios registrados

Registro: streamedu.eu

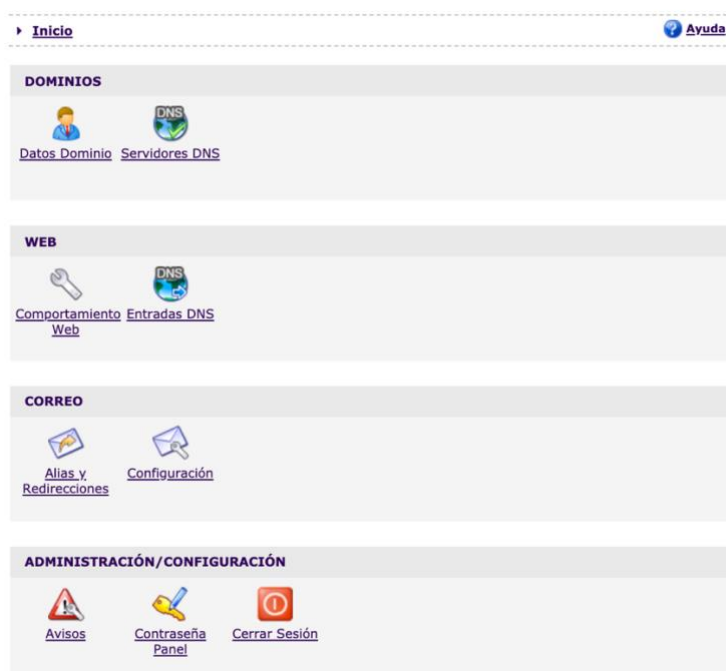


Imagen 21: Panel de administración de dominios Piensa Solutions

Como paso previo a la migración de la administración del dominio, se debe crear en el servicio Route53 de AWS una zona de hospedaje. El principal objetivo de esta zona de hospedaje, como bien se comentó en los apartados anteriores, es la administración del DNS permitiendo de esta forma una integración fácil con los demás servicios AWS.

Para finalizar la migración de dominio se requieren los registros del tipo NS para registrar la migración del dominio en la plataforma de Piensa Solutions. En las siguientes imágenes se muestran los registros del tipo NS utilizados en el proceso de migración.


<input type="checkbox"/>	Nombre	Tipo	Valor	Evaluar el estado del destino
<input type="checkbox"/>	streamedu.eu.	SOA	ns-1611.awsdns-09.co.uk. awsdns-hostmaster.amaz	-
<input checked="" type="checkbox"/>	streamedu.eu.	NS	ns-81.awsdns-10.com. ns-1203.awsdns-22.org. ns-1611.awsdns-09.co.uk. ns-882.awsdns-46.net.	-

Imagen 22: Panel de administración Route 53 AWS

Registro: streamedu.eu

► [Inicio](#) > [Listado Servidores DNS](#) ? [Ayuda](#)

LISTADO DE DNS

 [Añadir Servidor DNS](#)









Nombre de servidor DNS:	IP:	Modificar	Eliminar
ns-1203.awsdns-22.org	205.251.196.179		
ns-81.awsdns-10.com	205.251.192.81		
ns-882.awsdns-46.net	205.251.195.114		
ns-1611.awsdns-09.co.uk	205.251.198.75		

Imagen 23: Panel de administración DNS Piensa Solutions

b) Alojamiento web

Con respecto al alojamiento web este se ha configurado utilizando el servicio AWS S3. Para ello se requiere crear un *bucket* que servirá el contenido de la aplicación web. Para llevar a cabo este proceso se ha de acceder al servicio S3 a través de la consola de administración de AWS. Se han de seguir una serie de pasos en los cuales se ha de especificar en nombre del *bucket* y la región AWS en la cual se va a crear este *bucket*. En la siguiente imagen se ilustra el proceso de creación de este *bucket*.

Imagen 24: Proceso creación *bucket* AWS S3

Posteriormente con la configuración por defecto que ofrece este *bucket* solo se permite el almacenaje de objetos y por esta razón la configuración del *bucket* se ha de modificar. Para poder alojar un sitio web estático, se debe configurar este *bucket* para el alojamiento web y subir el contenido de la aplicación al *bucket*. Para llevar a cabo este proceso se accede a las propiedades del *bucket* creado y se configura el alojamiento a sitios web estáticos.

Imagen 25: Proceso configuración alojamiento web estático en AWS S3

Posteriormente es necesario generar el contenido estático de la aplicación web previamente desarrollada. Para esto se utiliza la CLI del *framework* Angular que será el encargado de generar los archivos estáticos necesarios. Este proceso se lleva a cabo mediante la siguiente sentencia de comando:

```
>>$ ng build --prod
```

Una vez finalizada la generación de los archivos necesarios, en la carpeta raíz del proyecto se encuentra una carpeta llamada *dist*. En ella se encuentran los archivos minificados por el *framework* y deben ser subidos al *bucket*.

Acto seguido, antes de realizar la subida de estos archivos se de añadir una política de *bucket* para que el contenido sea publico. Para ello se accede al panel de propiedades del *bucket* y en la sección de permisos añadir la siguiente política proporcionada por la documentación de AWS²⁷.

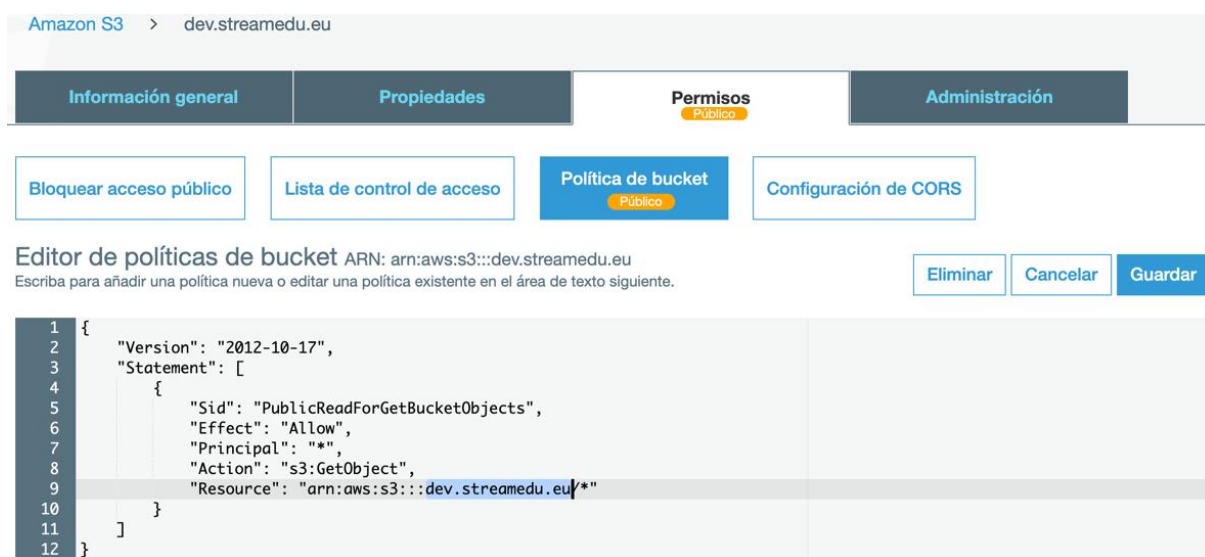


Imagen 26: Proceso configuración de la política de acceso al *bucket*

Por ultimo se realiza la subida de los archivos estáticos de la aplicación web al *bucket* y estos pasan a se accesibles a través del registro DNS que proporciona AWS S3 en la Imagen 26. Prestando atención al punto de enlace anteriormente mencionado, se puede observar que se trata de una conexión del tipo HTTP. A continuación, se realizan los ajusten necesarios para que la aplicación solo sea accesible a través de conexiones del tipo HTTPS.

c) Configuración de la CDN y certificado SSL

Como bien se comento previamente, es importante ofrecer una buena experiencia de navegación segura a los usuarios. En esta ocasión, para llevar a cabo estos requisitos, se explican los pasos seguidos para implementar una red de distribución de contenido. Como paso final se explica como se realizo la integración de certificado SSL a estos servicios y la creación de registros DNS necesarios para esta aplicación web.

En primer lugar, antes de comenzar la creación de la distribución CDN es conveniente solicitar los certificados SSL. Para llevar a cabo este proceso se utiliza el servicio Certificate Manager a través de la consola de administración de AWS.

Una vez situados en la consola del servicio Certificate Manager, se solicita un certificado publico que, de forma predeterminada, los navegadores y los sistemas operativos confían en los certificados públicos emitidos por la autoridad certificadora de AWS. En particular a la hora de solicitar este certificado existe la posibilidad de solicitarse para: un nombre de dominio completo (FQDN), para un nombre de dominio desnudo (APEX) y además también se permite

²⁷ Configuración de sitios web estáticos :

https://docs.aws.amazon.com/es_es/AmazonS3/latest/dev/HostingWebsiteOnS3Setup.html

usar el asterisco (*) como comodín en la posición de la izquierda para poder proteger varios subdominios del dominio APEX solicitado.

En esta ocasión se solicita de forma conjunta un certificado para el APEX de *streamedu.eu* y el **.streamedu.eu*. El proceso de validación de los certificados SSL se ha realizado creando los registros de tipo CNAME en el DNS para cada uno de los certificados solicitados. En la siguiente imagen se ilustra los certificados anteriormente mencionados.

<input type="checkbox"/>	Nombre ▾	Nombre de dominio ▾	Nombres adicionales	Estado ▾	Tipo ▾	¿En uso? ▾	Requisitos para la renovación ▾
<input type="checkbox"/>	▾	*.streamedu.eu	streamedu.eu	Emitido	Emitido por Amazon	Sí	Válido

Estado	
Estado	Emitido
Estado detallado El certificado se emitió a las 2019-04-18T16:37:58UTC	
Dominio	Estado de validación
▸ *.streamedu.eu	Correcto
▸ streamedu.eu	Correcto

[Exportar la configuración de DNS a un archivo](#) Puede exportar todos los registros CNAME a un archivo

Detalles	
Tipo	Emitido por Amazon
¿En uso?	Sí
Nombre de dominio	*.streamedu.eu
Número de nombres adicionales	1
Nombres adicionales	streamedu.eu
Identificador	b6638a00-6706-49c6-b5f3-88f4122f70
Número de serie	0a:a6:8c:99:92:e4:12:90:60:db:d5:fd:12:29:ce:0b
Recursos asociados	arn:aws:cloudfront::999349766468:distribution/E30EJUDXQG8X4G
Solicitado	2019-04-18T16:36:13UTC
Emitido	2019-04-18T16:37:58UTC
No anterior a	2019-04-18T00:00:00UTC
No posterior a	2020-05-18T12:00:00UTC
Información de clave pública	RSA 2048 bits
Algoritmo de firma	SHA256WITHRSA
ARN	arn:aws:acm:us-east-1:999349766468:certificate/b6638a00-6706-49c6-b5f3-88f4122f70
Estado de validación	Ninguno

Imagen 27: Certificado SSL para la aplicación web

A continuación, el siguiente paso es la creación y configuración de la red de distribución de contenido. Para ello se ha de acceder al servicio CloudFront desde el panel de administración de AWS y se crea una nueva distribución de contenido. En primer lugar, se ha de indicar de donde se provee el contenido que se distribuirá a través de esta CDN. En este caso el origen de contenido es el *bucket dev.streamedu.eu* creado previamente. Como parámetros extra de configuración se restringirá el acceso al contenido del *bucket* solo para la CDN y se concede permisos para que realice estas actualizaciones. A continuación, se adjunta una imagen que ilustran las configuraciones realizadas.

Origin Settings		
Origin Domain Name	dev.streamedu.eu.s3.amazonaws.com	?
Origin Path		?
Origin ID	S3-dev.streamedu.eu	?
Restrict Bucket Access	<input checked="" type="radio"/> Yes <input type="radio"/> No	?
Origin Access Identity	<input checked="" type="radio"/> Create a New Identity <input type="radio"/> Use an Existing Identity	?
Comment	access-identity-dev.streamedu.eu.s3.ar	?
Grant Read Permissions on Bucket	<input checked="" type="radio"/> Yes, Update Bucket Policy <input type="radio"/> No, I Will Update Permissions	?
Origin Custom Headers	Header Name	Value ?
	<input type="text"/>	<input type="text"/>

Imagen 28: Configuración del origen del contenido para la CDN

Posteriormente se debe indicar la configuración del comportamiento de cache de la CDN. En primer lugar, es en este punto donde se realiza la configuración necesaria para que todas las comunicaciones se realicen utilizando el protocolo HTTPS. Por lo tanto, se redirigen con la ayuda de la CDN todas las solicitudes del tipo HTTP a través de HTTPS y de esta forma se cumple el requisito establecido anteriormente.

Otro punto para tener en cuenta, son los métodos HTTP permitidos, pero al tratarse de una aplicación orientada a servicios, donde estos se ejecutan desde el navegador web, se tiene suficiente con los métodos GET y HEAD. Posteriormente se configura que se realice la compresión de los objetos de forma automática para mejorar los tiempos de carga de la aplicación web. A continuación, se junta una imagen que ilustra las configuraciones comentadas previamente.

Default Cache Behavior Settings

Path Pattern	Default (*)	i
Viewer Protocol Policy	<input checked="" type="radio"/> HTTP and HTTPS <input type="radio"/> Redirect HTTP to HTTPS <input type="radio"/> HTTPS Only	i
Allowed HTTP Methods	<input checked="" type="radio"/> GET, HEAD <input type="radio"/> GET, HEAD, OPTIONS <input type="radio"/> GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE	i
Field-level Encryption Config	<div>▼</div>	i
Cached HTTP Methods	GET, HEAD (Cached by default)	i
Cache Based on Selected Request Headers	<div>None (Improves Caching) ▼</div> Learn More	i
Object Caching	<input checked="" type="radio"/> Use Origin Cache Headers <input type="radio"/> Customize	i
	Learn More	
Minimum TTL	<div>0</div>	i
Maximum TTL	<div>31536000</div>	i
Default TTL	<div>86400</div>	i
Forward Cookies	<div>None (Improves Caching) ▼</div>	i
Query String Forwarding and Caching	<div>None (Improves Caching) ▼</div>	i
Smooth Streaming	<input type="radio"/> Yes <input checked="" type="radio"/> No	i
Restrict Viewer Access (Use Signed URLs or Signed Cookies)	<input type="radio"/> Yes <input checked="" type="radio"/> No	i
Compress Objects Automatically	<input checked="" type="radio"/> Yes <input type="radio"/> No	i
	Learn More	
Lambda Function Associations		i
	<div>CloudFront Event</div> <div>Select Event Type ▼</div>	<div>Lambda Function ARN</div> <div></div>

Imagen 29: Configuración del comportamiento de la cache para la CDN

Otro paso que se realiza en esta parte de configuración es la definición de dominios alternativos a los cuales debe responder esta distribución y para ello se configura el CNAME *dev.streamedu.eu*. Posteriormente se elige la opción de usar el certificado SLL personalizado creado previamente. A continuación, se adjunta una imagen que ilustra estas últimas configuraciones realizadas y finalmente se procede a la creación de la CDN.

Imagen 30: Configuración de la distribución CDN

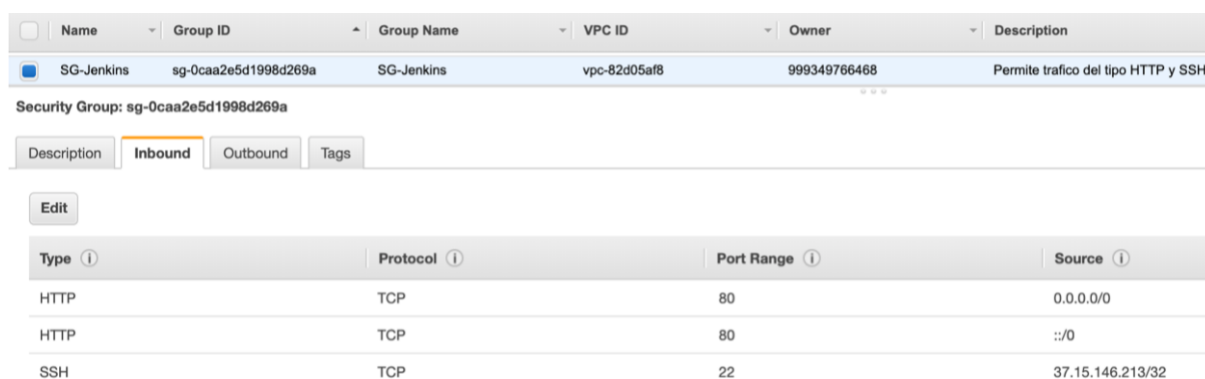
4.5. Configuración de la herramienta de automatización

Con respecto al proceso de configuración de la herramienta de automatización este estará dividido tres partes. Se comenzará hablando de la creación del servidor que aloja la herramienta Jenkins en el entorno AWS, después se realiza la instalación con una configuración básica del Jenkins y finalmente se explica el proceso de automatización implementado.

a) Despliegue de una instancia de computo en AWS

Para empezar, cabe recordad que, en el apartado de análisis de costes se ha propuesto implementar la herramienta de automatización utilizando el servicio EC2 (Elastic Cloud Computing) del mismo proveedor AWS. Con este motivo se considera oportuno explicar las configuraciones mas relevantes que se deben tener en cuenta antes de lanzar una nueva instancia.

Ahora bien, como la seguridad en todo sistema de comunicación es un factor importante, para el servidor que se va a desplegar se necesita configurar un firewall y que a nivel de instancias de computo de AWS este proceso se lleva a cabo a través de los grupos de seguridad. Por defecto en AWS cada vez que se lanza una nueva instancia de computo, todo el trafico entrante para la instancia es denegado por defecto. Por lo tanto, se debe permitir el trafico entrante a través del puerto 80, para tener acceso a la interfaz grafica del Jenkins, y por otro lado se debe permitir de forma temporal el trafico a través del puerto 22 para establecer conexiones SSH con el propósito de realizar las instalaciones y configuraciones oportunas en la instancia. A continuación, se adjunta una imagen que ilustra el grupo de seguridad creado.



Name	Group ID	Group Name	VPC ID	Owner	Description
SG-Jenkins	sg-0caa2e5d1998d269a	SG-Jenkins	vpc-82d05af8	999349766468	Permite trafico del tipo HTTP y SSH

Security Group: sg-0caa2e5d1998d269a

Description Inbound Outbound Tags

Edit

Type	Protocol	Port Range	Source
HTTP	TCP	80	0.0.0.0/0
HTTP	TCP	80	:::0
SSH	TCP	22	37.15.146.213/32

Imagen 31: Grupo de seguridad asociado a la instancia Jenkins

Por otro parte, con el propósito de autorizar el uso de los distintos servicios de AWS desde la instancia es necesario configurar un rol utilizando el servicio AWS IAM (Identity and Access Management). Los servicios con los cuales debe interaccionar esta instancia son: AWS S3, para poder actualizar al contenido de la aplicación web, y AWS CloudFront para poder invalidar el contenido almacenado en los distintos puntos finales de la red de distribución de contenido. En definitiva, este rol puede ser añadido a la instancia en el momento de crearla y de esta forma se asegura la integridad de las credenciales al no tener que guardarlas de forma manual en la instancia. En la siguiente imagen se muestra el rol IAM creado para esta instancia utilizando políticas de uso de servicios provista por AWS.

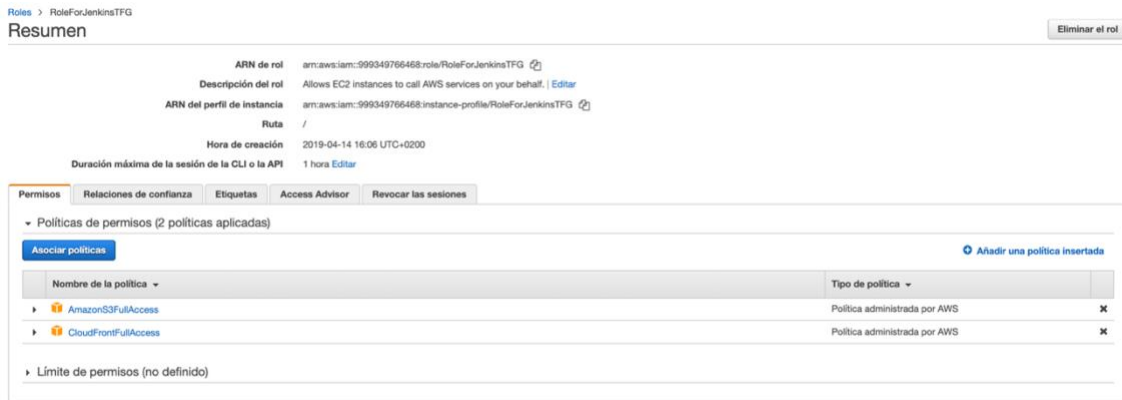


Imagen 32: Rol IAM para la instancia Jenkins

Para concluir con estas configuraciones se crea una instancia de computo en la cual se procede a la instalación y configuración del Jenkins.

b) Instalación y configuración básica del Jenkins

La instancia de computo lanzada en el anterior apartado se basa en una imagen de Amazon Linux que implementa el gestor de paquetes *yum*. Para facilitar la instalación del Jenkins se crea el siguiente *bash script* que se ejecuta en la instancia a través de una conexión SSH. En pocas palabras con este script se instala y configura un servidor web *nginx*, la versión de *java* que necesita *jenkins* para ser instalado, se instala *git* y se elimina la versión de java que el sistema operativo trae por defecto.

```
#!/bin/bash
yum -y update
yum remove -y java-1.7.0
yum install -y nginx java-1.8.0 git
wget -O /etc/yum.repos.d/jenkins.repo http://pkg.jenkins-ci.org/redhat-stable/jenkins.repo
rpm --import http://pkg.jenkins-ci.org/redhat-stable/jenkins-ci.org.key
yum -y update
yum install -y jenkins
service jenkins start
adminpasswd=`sudo cat /var/lib/jenkins/secrets/initialAdminPassword`
echo -e " Contraseña para desbloquear el Jenkins: $adminpasswd "
conf=/etc/nginx/conf.d/jenkins.conf
cat >> $conf << EOF
upstream jenkins {
    server 127.0.0.1:8080 fail_timeout=0;
}
server {
    listen 80 default_server;
    access_log /var/log/nginx/jenkins.access.log;
    error_log /var/log/nginx/jenkins.error.log;
    proxy_buffers 16 64k;
    proxy_buffer_size 128k;
    location / {
        proxy_pass http://jenkins;
        proxy_next_upstream error timeout invalid_header http_500 http_502 http_503 http_504;
        proxy_set_header Host $host:$server_port;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
EOF
sed -e 's/^listen / s/^#/#/' -i /etc/nginx/nginx.conf
chkconfig nginx on
service nginx start
```

Una vez finalizado este proceso de instalación se accede a la interfaz grafica de Jenkins con la contraseña facilitada por el anterior script. Se procede a crear un usuario administrador y se instalan los *plugins* recomendados por defecto. Como paso final de este proceso se instala a través de la interfaz grafica del Jenkins el *plugin* de *BlueOcean* mediante el cual se crea proyecto de desarrollo de software conectado al repositorio *git* que almacena el código de la aplicación web desarrollada.

c) Proceso de automatización implementado

Con el fin la facilitar el proceso de actualización de la aplicación web se ha implementado un pipeline de entrega de software. En su esencia un pipeline de entrega de software implementa de forma automatizada la construcción, el testing, el despliegue y el proceso de puesta en producción de la aplicación web desarrollada. Entre los principales propósitos de estos pipelines se encuentra la visibilidad de cada uno de los pasos de despliegue de software mencionados anteriormente. Por otro lado, se facilita el *feedback* constante de forma instantánea, ayudando de esta forma la detección de errores o fallos de integración de las nuevas funcionalidades implementadas en la aplicación. Por lo ultimo, facilita la entrega del software compilado y probado en un entorno similar al entorno productivo, dejado de esta forma preparada la aplicación web para que esta sea desplegada en el entorno productivo.

En cuanto a la definición de estos pipelines de desarrollo de software la herramienta Jenkins permite definirlos a través de la interfaz grafica o escribiendo un script de forma declarativa. El script se recomienda guardarlo en la raíz del repositorio, bajo el nombre de *Jenkinsfile*, junto con el código de la aplicación. En pocas palabras el contenido del script *Jenkinsfile* trata de automatizar los pasos de compilación de código, actualización de código compilado en el *bucket* S3 y la invalidación del contenido cacheado en los puntos extremos de la CDN.

Por otro lado, en la herramienta Jenkins se define un proyecto de software en el cual se realizan una serie de configuraciones, en las cuales se especifica por ejemplo la URL donde el repositorio debe notificar a Jenkins que han ocurrido cambios en el proceso de desarrollo. De forma automática, Jenkins al recibir esta notificación procede al clonado del repositorio y ejecuta la serie de sentencias definidas en el script *Jenkinsfile*.

Para ilustrar este proceso a continuación se adjuntan unas imágenes que ilustran los pasos que Jenkins siguen en función de las tareas definidas en el script *Jenkinsfile*.

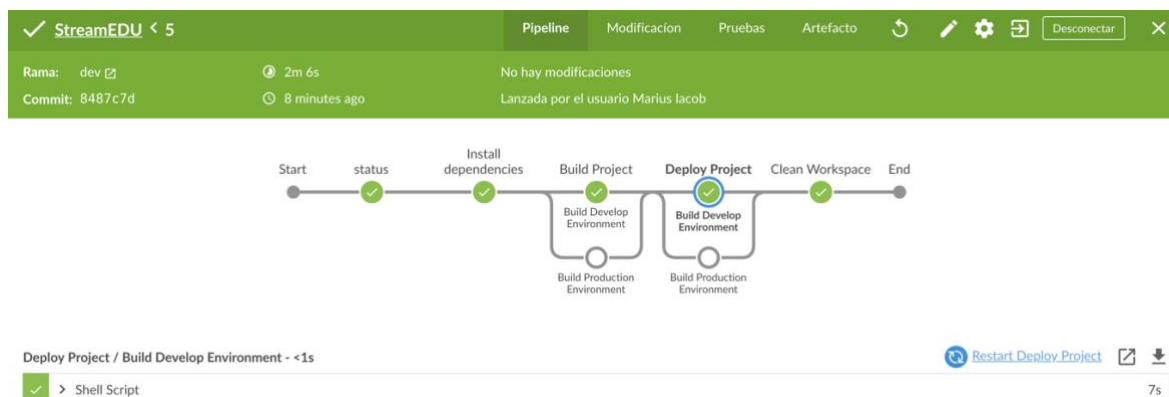


Imagen 33: Representación del pipeline de la rama *dev* del repositorio

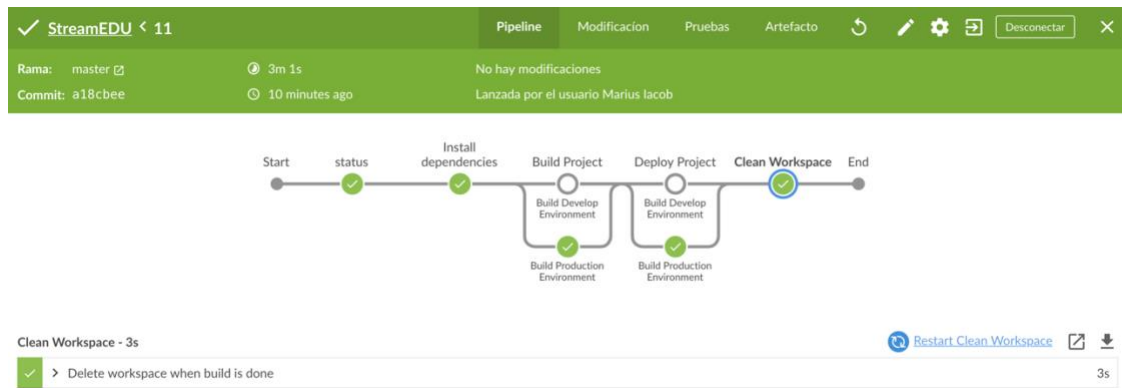


Imagen 34: Representación del pipeline de la rama *master* del repositorio

A continuación, se adjunta el contenido del script *Jenkinsfile* que hace posible el proceso de automatización y representación grafica de los mismos.

```

pipeline { agent any
  stages {
    stage('status') { steps { sh 'ls -la' } }
    stage('Install dependencies') { steps { sh "'npm install'" } }
    stage('Build Project') {
      parallel {
        stage('Build Develop Environment') { when { branch 'dev' }
          steps { sh "'ng build'" }
        }
        stage('Build Production Environment') { when { branch 'master' }
          steps { sh "'ng build --prod'" }
        }
      }
    }
    stage('Deploy Project') {
      parallel {
        stage('Build Develop Environment') { when { branch 'dev' }
          steps { sh "'
            aws s3 rm s3://dev.streamedu.eu --recursive
            aws s3 sync ./dist/streamedu s3://dev.streamedu.eu \
              --acl public-read --cache-control \"public, max-age=60\"
            aws s3api copy-object --copy-source dev.streamedu.eu/index.html --cache-control \
              \"public, max-age=60\" --content-type \"text/html\" \
              --bucket dev.streamedu.eu --key index.html --metadata-directive=\"REPLACE\"
            aws configure set preview.cloudfront true
            aws cloudfront create-invalidation \
              --distribution-id=E2OBWN1AFE4OTK --paths /index.html
          '" }
        }
        stage('Build Production Environment') { when { branch 'master' }
          steps { sh "'
            aws s3 rm s3://streamedu.eu --recursive
            aws s3 sync ./dist/streamedu s3://streamedu.eu \
              --acl public-read --cache-control \"public, max-age=60\"
            aws s3api copy-object --copy-source streamedu.eu/index.html --cache-control \
              \"public, max-age=60\" --content-type \"text/html\" \
              --bucket streamedu.eu --key index.html --metadata-directive=\"REPLACE\"
            aws configure set preview.cloudfront true
            aws cloudfront create-invalidation \
              --distribution-id=E30EJUDXQG8X4G --paths /index.html
          '" }
        }
      }
    }
    stage('Clean Workspace') {
      steps { cleanWs(cleanWhenAborted: true, cleanWhenFailure: true, cleanWhenNotBuilt: true,
        cleanWhenSuccess: true, cleanWhenUnstable: true, cleanupMatrixParent: true,
        deleteDirs: true, disableDeferredWipeout: true)
    }
  }
}

```


5. ORGANIZACIÓN DEL TRABAJO

Acerca de la organización del trabajo, inicialmente se ha llevado a cabo una planificación basada en un diagrama de Gantt a partir del cual ha sido posible evaluar los diferentes hitos del trabajo y cual era el progreso, así como los posibles avances y/o retrasos. Con referencia a la organización de las tareas diarias se ha utilizado la herramienta Trello. En relación con el diagrama de Gantt, este ha servido para realizar una aproximación de las principales tareas que se debían llevar a cabo y una estimación del tiempo que se le debía dedicar. En cuanto a esta primera aproximación del diagrama de Gantt, como era de esperar, han ido sucediendo dificultades a la hora de cumplir los plazos estimados. A continuación, se adjunta una imagen donde se ilustra la planificación del trabajo mediante el diagrama de Gantt:

	Nombre de la tarea	Fecha de Ini	Fecha final	Duración	Predecesores
1	StreamEDU				
2	Seguimiento del TFG	10/12/18	23/08/19	185d	
3	Revisiones	10/12/18	23/08/19	185d	
4	Reuniones	11/12/18	11/12/18	1d	
5	Generación de la Memoria TFG	12/12/18	10/06/19	129d	
6	Estudio previo	12/12/18	14/12/18	3d	
7	Descomposicion de tareas	12/12/18	25/12/18	10d	6CC
8	Estimación de tiempo	12/12/18	13/12/18	2d	7CC
9	Documentación	12/12/18	10/06/19	129d	7CC
10	Defensa del TFG	10/06/19	14/06/19	5d	
11	Preparación	10/06/19	13/06/19	4d	
12	Defensa	14/06/19	14/06/19	1d	11
13	Formación	14/12/18	07/03/19	60d	8
14	Análisis	14/03/19	10/04/19	20d	
15	Definición del sistema	14/03/19	15/03/19	2d	14CC
16	Análisis de requerimientos	18/03/19	20/03/19	3d	15
17	Casos de uso	21/03/19	25/03/19	3d	16
18	Diagramas de actividad	26/03/19	27/03/19	2d	17
19	Identificación de componentes	28/03/19	01/04/19	3d	18
20	Revision del análisis	02/04/19	03/04/19	2d	19
21	Diseño	04/04/19	15/05/19	30d	20
22	Estudio previo	04/04/19	05/04/19	2d	21CC
23	Diseño BBDD NoSql	04/04/19	10/04/19	5d	21CC
24	Diseño interfaces	04/04/19	05/04/19	2d	21CC
25	Diseño Mockups	04/04/19	08/04/19	3d	21CC
26	Revision Diseño	11/04/19	15/04/19	3d	23
27	Desarrollo	16/05/19	16/05/19	1d	21
28	Estudio previo	16/05/19	17/05/19	2d	21
29	Implementacion BBDD NoSQL	20/05/19	22/05/19	3d	28
30	Implementación Autenticación	23/05/19	03/06/19	8d	29
31	Revision de código	04/06/19	06/06/19	3d	30
32	Despliegue aplicación	07/06/19	10/06/19	2d	31

Imagen 35: Diagrama de Gantt [Anexo 2]

Por lo que se refiere a la herramienta Trello, esta ha servido para desglosar en tareas pequeñas los grandes bloques del anterior diagrama de Gantt. Hay que destacar que: Trello es una herramienta administradora de tareas comúnmente utilizada en el desarrollo de proyectos de todo tipo. Además, Trello es una herramienta basada en el método Kanban y ha servido para ver la planificación de las tareas a corto plazo. De esta forma se ha podido detectar si algunas de las tareas planificadas a corto plazo necesitaban mas dedicación y ha servido para el desarrollo del proyecto.

En cuanto al funcionamiento de esta herramienta se puede decir que es muy fácil e intuitiva. Se aproxima al uso de unas tarjetas de colores donde cada una de estas tarjetas debe formar parte de una lista, siendo ambas totalmente personalizables. En cuanto al uso de las tarjetas, se definían pequeñas tareas con una breve descripción que iba acompañada en ocasiones de un enlace que preveía documentación sobre la tarea. Por otro lado, los colores de las tarjetas han servido para ubicar y enlazar las tareas que se llevaban a cabo en la memoria. Cabe destacar que a las tarjetas se les añadía una alarma que notificaba si se llegaba a la fecha límite de finalización, permitiendo de esta forma medir si se cumplían los plazos estimados.

Acerca de las listas, se han creado un total de cinco listas: el backlog o la cantidad de tareas a realizar, la lista de esta semana, las tareas en progreso, las tareas finalizadas y por lo ultimo las tareas validadas. La siguiente imagen ilustra un ejemplo la organización llevada a cabo a través de esta herramienta.

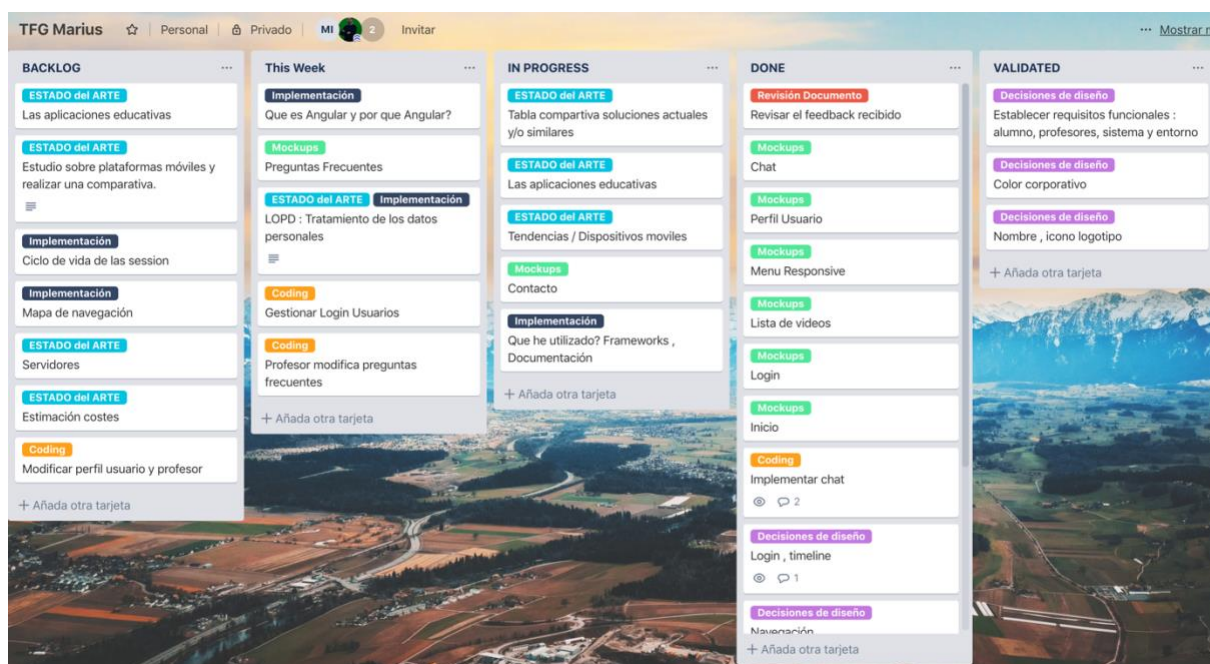


Imagen 36: Organización de trabajo con Trello

6. TRABAJO FUTURO

Una vez finalizado este proyecto surgen varias ideas a partir de las cuales se puede seguir desarrollando y mejorando distintas funcionalidades de esta aplicación web. A continuación, se realiza un breve resumen de las principales ideas y funcionalidades que se deberían mejorar.

Implementar las funcionalidades restantes de YouTube

Durante el trabajo se ha dado mucha importancia a la API de YouTube puesto que es uno de los servicios básicos de la aplicación. Para el prototipo se ha implementado solamente la obtención de los videos de un canal específico de YouTube. Hay que destacar que para poder implementar las funcionalidades de subir, editar y eliminar videos se debería utilizar la librería JavaScript *gapi* que provee la API de Google para sus servicios. En comparación con el servicio de autenticación de Firebase, esta librería permite solicitar los permisos necesarios a los usuarios para poder de esta forma implementar las funcionalidades restantes.

Escalabilidad de la aplicación

Por lo que se refiere a la escalabilidad de la aplicación, esta debería evaluarse al menos en términos de arquitectura y numero de colegios. Ahora bien, en términos de arquitectura, esta aplicación ha sido diseñada para soportar un numero ilimitado de alumnos. Esto es posible gracias a los distintos servicios y tecnologías utilizadas, pero hay que destacar que no se ha realizado una planificación sobre como organizar y mostrar el contenido que se ira publicando en la aplicación.

En consecuencia, la aplicación diseñada podría resultar deficiente a la hora de mostrar el contenido educativo a los alumnos de un curso concreto. De esto se desprende que por una parte se debería tener en cuenta que curso esta cursando cada uno de los alumnos registrados en la aplicación. Esta parte organizativa de los alumnos se podría llevar a cabo a través del panel de administración de usuarios.

Otro punto que se debería tener en cuenta es la publicación del contenido video y la asociación de este contenido a distintos cursos. Esta organización del contenido video se podría llevar a cabo a través del uso de los *tags* que se solicitan a la hora de publicar el contenido video. Con la ayuda de estos *tags* se podría realizar un filtrado de los videos teniendo en cuenta el curso de cada uno de los alumnos. Como alternativa al uso de los *tags* se podría guardar un registro en Firebase de los videos que se publican y a que curso van destinados. Esta ultima solución sigue siendo valida, pero se le añade mas lógica a la aplicación y esto implica que se realice al menos una consulta extra a base de datos de Firebase. De esto se desprende que el uso de los *tags* podría resultar mas eficiente para proporcionar una mejor experiencia de navegación.

Por ultimo, en cuanto a la escalabilidad de la aplicación en términos de distintos colegios, esto se podría realizar tanto de forma horizontal como vertical. Ahora bien, con referencia al escalado horizontal a distintos colegios, la mayoría de ellos disponen de una pagina web y por lo tanto de un dominio web registrado. Con esto la aplicación web desarrollada se podría distribuir a través de un subdominio del colegio, pero se debería estudiar la viabilidad de la integración con los distintos servicios de AWS utilizados. En cambio, con el escalado vertical a distintos colegios se dispone de una única aplicación para todos los colegios. De esta forma se podrían reducir de forma considerable los costes que repercutirían, en cuanto a la solución

desplegada o al desarrollo continuo de la aplicación propuesta, a cada uno de los colegios. Para poder llevar a cabo este tipo de escalado vertical a distintos colegios, se debería analizar e implementar un mecanismo que permita distinguir a qué colegio pertenece cada alumno. En cuanto a los profesores sería interesante permitirles reutilizar los contenidos educativos publicados por la comunidad de profesores y poder asignarlos a un grupo de alumnos específicos.

Integración de otros métodos de autenticación

En relación con la integración de otros métodos de autenticación, el prototipo diseñado permite la autenticación de los usuarios a través de una cuenta de Gmail o mediante una dirección de correo y contraseña, que previamente deben estar registrados en la aplicación. Puesto que Firebase ofrece la posibilidad de autenticarse con otras redes sociales, sería interesante poder implementar algún método de autenticación distinto a los mencionados anteriormente. Para esto se debería tener en cuenta la compatibilidad de estos métodos de autenticación en términos de compatibilidad con las funcionalidades de YouTube o los mecanismos de escalabilidad que por el momento no se han implementado.

Tratamiento de los datos

Con respecto al cumplimiento de la LODP sería conveniente definir los términos y condiciones de uso del servicio y obligar la lectura y el consentimiento del tratamiento de estos datos antes de facilitar el uso de la aplicación. Por otra parte, se debería facilitar a los usuarios la opción de eliminar su cuenta de usuario y todos los datos que se han generado a lo largo de su ciclo de vida en la aplicación.

Integración de podcasts

Por lo que se refiere a los podcasts serían otro tipo de servicio que se podrían integrar en esta aplicación web. Con este tipo de servicio se podría distribuir contenido educativo como por ejemplo clases de inglés o clases de historia entre otras. Spotify o iTunes son algunas de las plataformas que ofrecen una API para poder integrar este tipo de contenido educativo.

Integración de las notificaciones

Por otro lado, se deberían implementar mecanismos que permitan las notificaciones a los alumnos cuando un nuevo contenido educativo ha sido publicado en la aplicación. Estas notificaciones podrían realizarse a través de mensajes SMS o email. De modo que entre los posibles servicios que ofrecen estos tipos de funcionalidades y que se pueden integrar a la aplicación web, vale la pena enumerar los servicios ofrecidos por Mandrill, SendGrid o AWS Simple Email Service entre otros.

Además, también se debería implementar las notificaciones del tipo PUSH para que, tanto los alumnos como los profesores sean notificados cuando se envía una consulta al chat de los videos.

7. CONCLUSIONES

Si se toma la frase introductoria del primer capítulo de la memoria *“El objetivo de este trabajo es desarrollar una aplicación web con el propósito de solucionar un problema de accesibilidad en los colegios. Será proporcionada de forma gratuita y con el objetivo de ser una aplicación de software libre. Con este trabajo se pretende aplicar los conocimientos adquiridos durante el grado para analizar, planificar, diseñar, desarrollar, gestionar el proyecto y entregar a tiempo un producto mínimo viable (M.V.P.), fácil de mantener y evolucionar de manera constante.”* y se valora de forma global si se han cumplido los objetivos, la respuesta es si. Pero hay que tener en cuenta que este “si” tiene algunos matices que se explicaran a continuación.

Con respecto a las conclusiones se ha decidido dividir las en tres partes: los aspectos positivos, que hacen referencia a lo que se ha realizado de forma adecuada durante el proyecto, los aspectos mejorables, que hacen referencia a la perspectiva global una vez finalizado el proyecto y por ultimo, las lecciones aprendidas, que hacen referencia a todo el proceso de aprendizaje adquirido durante este proyecto.

Aspectos positivos

Para comenzar a evaluar lo que ha sido apropiado para este proyecto, se podría hablar de la duración y organización del proyecto. La magnitud de este proyecto se aproxima a situaciones que se dan en el mundo laboral y ha servido para poder medir, evaluar y priorizar las distintas entregas que se han ido realizando. Por otra parte, la organización en un proyecto es clave (como bien se ha comentado en el capítulo que hace referencia a la Organización del trabajo) si además se ha de compaginar con un trabajo a tiempo completo u otras responsabilidades que uno puede llegar a tener.

Otro de los aspectos positivos a la hora de realizar este trabajo de final de grado ha sido la definición desde un principio de las fases y las secciones que deberían cubrir un proyecto de esta envergadura. Hay que destacar que la mayoría de las fases y secciones que se han llevado a cabo en este proyecto son transversales y se pueden aplicar todo tipo de proyectos que pueden surgir en el entorno laboral.

Por otra parte, con referencia al problema propuesto a solucionar y al estado del arte, han servido para mejorar los hábitos para analizar una situación actual, de los colegios del entorno rural, que se podría decir que se encuentran en una situación de olvido o marginación. Además, se ha realizado un estudio sobre las distintas soluciones educativas, donde la mayoría de estas soluciones son enfocadas a un uso masivo y con una multitud de funcionalidades que podrían resultar difíciles de usar para ese tipo de público.

En cuanto al análisis y diseño de la aplicación, ha sido una parte complicada que al mismo tiempo ha resultado positiva después de haber analizado las distintas plataformas aprendizaje educativo. De modo similar se ha tenido que realizar un esfuerzo a la hora del desarrollo de la aplicación utilizando el *framework* Angular, por lo que se han tenido que dedicar una parte del tiempo de este proyecto a este aprendizaje. Sin embargo, se tenía una base conocimiento de los lenguajes de marcado y programación, que en tal sentido se han visto ampliados todos estos conocimientos.

Al mismo tiempo, el proceso de implementación y despliegue de la aplicación ha resultado interesante y atractivo, al ver como la implementación de un sistema de control de versiones notifica a una herramienta de automatización de software los cambios y que siguiendo el *pipeline* la nueva versión de la aplicación es comprobada y actualizada de forma casi instantánea en una CDN.

Aspectos mejorables

En cambio, con referencia a la planificación de este proyecto, se ha comentado que esta ha sido correcta, pero con ciertos matices, ya que una vez finalizado el proyecto se ha observado que no se han podido desarrollar todas funcionalidades propuestas para este prototipo. En particular estas funcionalidades no se han implementado debido a la falta de experiencia, cierta complejidad y limitación en términos de uso de la API de YouTube. Al mismo tiempo se ha realizado un esfuerzo, que a lo mejor no ha sido suficiente, para entender y aplicar los medios de seguridad en las peticiones HTTP que se realizan a la API de YouTube utilizando los Json Web Token (JWT).

Otro punto mejorable, aunque en todo momento se ha tratado de seguir el análisis y diseño previo realizado, esto no ha sido del todo posible, ya que este en ocasiones ha resultado un poco deficiente y ha implicado que se vuelvan a plantear varios puntos que previamente se habían decidido.

Lecciones aprendidas

Este proyecto, en gran parte ha sido enfocado en el desarrollo y despliegue de software y ha sido un reto a nivel personal. Se ha comenzado con el pretexto de aprender programar y desplegar aplicaciones web multiplataforma, cosa que al final no había profundizado del todo durante el grado.

Además, para poder cumplir este reto, durante este proyecto se ha tenido la oportunidad de aplicar los conocimientos teóricos y prácticos adquiridos a lo largo del grado. Esto ha servido para experimentar y adquirir ciertas habilidades en las técnicas y tecnologías que se usan actualmente.

Hay que destacar que se ha reforzado la capacidad de análisis y gestión de proyectos acordes a las necesidades detectadas. Se han aprovechado las tecnologías que se han considerado mas adecuadas para este proyecto y se ha hecho uso de forma practica de los distintos servicios provistos por las principales compañías de Cloud.

En cuanto a la implementación, ha sido una buena forma de iniciarse en el desarrollo y despliegue de aplicaciones multiplataforma. No obstante, queda un largo camino por recorrer, pero al menos se han adquirido unos conocimientos básicos de Angular en cuanto al desarrollo usando este *framework*. Del mismo modo se han descubierto los beneficios de las herramientas de automatización, arquitecturas de diseño y arquitecturas *serverless* presentes cada vez mas en el desarrollo de todo tipo de soluciones asociadas a las tecnologías de información y comunicación.

8. GLOSARIO

Agile

Es una de las varias metodologías que facilitan la incorporación de los cambios con rapidez en el desarrollo de proyectos. A menudo esta metodología es utilizada en el mundo del desarrollo de software.

AMI

Amazon Machine Image es una imagen de una configuración de un sistema operativo, que proporcionada de esta forma la información necesaria para lanzar una instancia.

Angular CLI

Es la herramienta de línea de comandos que proporciona el *framework* Angular para crear, depurar y compilar aplicaciones .

APEX

Es un termino utilizado para describir el punto de nivel superior de un espacio de nombres de DNS.

API

Application Programming Interface o Interfaz de Programación de Aplicaciones, es un subconjunto de rutinas, procedimientos y funciones que se utilizan para diseñar e integrar el software de las aplicaciones.

Archivos minificados

Es una técnica con la cual los ficheros se comprimen al máximo posible sin perder sus funcionalidades. Con esta técnica se reduce el tamaño de los ficheros a través de la eliminación de los espacios, saltos de línea, comentarios o tabulación. Como resultado se obtiene un fichero de una sola línea.

Autoridad Certificadora

Es una entidad de confianza que se responsabiliza de la emisión o revocación de certificados para garantizar que las comunicaciones se realizan con la persona o el servicio que dice que es.

BaaS

Back-end as a Service es un tipo de servicio que ofrece la posibilidad de prescindir de la programación de un *back-end*. Con este tipo de servicios se garantiza el respaldo y la accesibilidad de los datos.

Back-end

Es la capa que permite el acceso a los datos de un software. Esta capa de abstracción no es directamente accesible por los usuarios y contiene la lógica de la aplicación.

Blue Ocean

Es un proyecto que replantea la experiencia de usuario de la plataforma Jenkins. Con este proyecto se presenta y se modela el proceso de entrega de software al proporcionar la información relevante al equipo de desarrollo.

Bootstrap

Es un *framework* de CSS y JavaScript que facilita la creación de interfaces de usuario con un diseño que se adaptan a cualquier dispositivo.

CircleCI

Es una plataforma moderna de integración continua y entrega continua de software que automatiza la automatización, prueba y despliegue de software. Es una solución propietaria que puede ser utilizada tanto en centros de datos como en entornos Cloud.

CMS

Content Management System es un software de desarrollado que facilita a los usuarios crear, administrar y gestionar contenido web con cierta facilidad.

CNAME

Canonical Name o registro de nombre canónico, es un tipo de registro que se utilizan en un DNS. Este tipo de registro permite al usuario definir un alias a un nombre de dominio.

Codeship

Es una herramienta que facilita a los programadores la integración continua y entrega continua de software en el Cloud con aplicaciones que se desarrollan utilizando repositorios de código de GitHub, Bitbucket o GitLab.

Data-binding

Este concepto en el mundo de la programación es un mecanismo que enlaza los datos entre la interfaz grafica con la lógica de la aplicación.

DNS

Domain Name System o Sistema de Nombres de Dominio es el mecanismo que facilita la traducción de los nombres de domino a direcciones IP que se corresponden servidores que alojan los servicios.

EC2

Elastic Compute Cloud es el servicio Amazon Web Services que ofrece capacidad de computo en la nube.

Font Awesome

Es un *framework* de iconos vectoriales y hojas de estilo CSS que se adaptan y se integran con facilidad en el proceso de desarrollo de aplicaciones web

FQDN

Fully Qualified Domain Name es la dirección de dominio web completa que se asocia a una aplicación. Esta dirección es necesaria para tener presencia en internet a través de un sistema de resolución de nombres.

Framework

Un *framework* en el mundo del desarrollo se define como un conjunto de: herramientas, librerías y buenas practicas, que permiten resolver una serie de problemas.

Front-end

Es la parte de una aplicación web que permite al usuario acceder e interactuar a través de ella. Esta parte de la aplicación es la que se ejecuta en el navegador del cliente.

Git

Es un software de control de versiones y su principal propósito es llevar un registro de los cambios que se realizan en los archivos.

Heroku

Es una plataforma como servicio que facilita realizar los despliegues de aplicaciones a los desarrolladores sin la necesidad de configurar la base de datos o el sistema operativo. Además, facilita contralar la disponibilidad y escalabilidad, a nivel de infraestructura, de la aplicación desplegada.

Hosting

El hosting o el alojamiento web es tipo de servicio que se ofrece para desplegar un sitio web o una aplicación. Con este servicio se alquila un espacio determinado en un servidor.

IAM

Identity and Access Management es un tipo de servicio ofrecido por las plataformas de Cloud para la administración de identidades y para el acceso a los servicios y recursos de forma segura.

Kanban

Es un método que permite la gestión de trabajo, ideado por Toyota, y que comúnmente es utilizado para mejorar los procesos de producción a través de tarjetas que permiten identificar las necesidades de la cadena de producción.

Mock-up

Es una maqueta o un modelo utilizado comúnmente por los desarrolladores para demostrar y evaluar con el cliente los diseños realizados.

Nginx

Es un servidor web de código abierto que también es utilizado como proxy inverso, balanceador de carga o como cache de HTTP.

NoSQL

Not only SQL o No solo SQL es una clase de sistemas de gestión de base de datos que difiere del modelo clásico de los sistemas de bases de datos relacionales. Este tipo de bases de datos NoSQL permiten una mejor adaptación a problemas de escalabilidad y rendimiento en comparación con las tradicionales bases de datos SQL.

OAuth 2.0

Es un protocolo de autorización que permite a los usuarios autorizar a terceros para acceder a los datos que se les concede permiso sin necesidad de facilitar las credenciales del usuario.

SDK

Software Development Kit o Kit de Desarrollo de Software es un conjunto de herramientas que permiten la programación de aplicaciones móviles.

SSH

Secure Shell es un protocolo que facilita a los usuarios la administración de los servidores a través de internet utilizando técnicas de encriptación.

SSL

Secure Sockets Layer es un protocolo de comunicación diseñado para permitir a las aplicaciones transmitir información de forma segura.

Starvation

Es un problema relacionado a los sistemas multitarea donde a alguno de los procesos o de los hilos en ejecución siempre se le deniega el acceso a un recurso compartido.

VPS

Virtual Private Server o Servidor Virtual Privado es maquina virtual alojada en un servidor físico. Es uno de los servicios de hosting mas populares a la hora de implementar un sitio web.

9. BIBLIOGRAFIA

- [1] Santamaría Conde, R. M. S. C. Rosa M^a, De la Fuente Anuncibay, R. F. A. Raquel, & Universidad de Burgos. (s.f.). LA DIRECCIÓN EN UN COLEGIO RURAL AGRUPADO. Recuperado 14 marzo, 2019, de <http://www.uco.es/dptos/educacion/congresolider/comunica8.htm>
- [2] Ministerio de Educación y Ciencia. (1986, 24 diciembre). BOE.es - Documento BOE-A-1987-353. Recuperado 15 marzo, 2019, de <https://www.boe.es/eli/es/rd/1986/12/24/2731>
- [3] Colaboradores de Wikipedia. (2019, 14 febrero). [Colegio Rural Agrupado]. Recuperado 8 mayo, 2019, de https://es.wikipedia.org/wiki/Colegio_Rural_Agrupado
- [4] Lacort Navarro, L. C. R. Laura. (2014, 18 julio). La identidad del alumnado en la practica docente diaria actual en un Colegio Rural Agrupado. Recuperado 8 mayo, 2019, de <https://reunir.unir.net/bitstream/handle/123456789/2569/lacort.navarro.pdf?sequence=1>
- [5] WeAreSocial, & HootSuite. (s.f.). Digital en 2019 en España - We Are Social ES. Recuperado 15 abril, 2019, de <https://wearesocial.com/es/digital-2019-espana>
- [6] Instituto Nacional de Estadística. (s.f.). España en cifras 2018 [Conjunto de datos]. Recuperado 16 marzo, 2019, de https://www.ine.es/prodyser/espa_cifras/2018/files/assets/common/downloads/publication.pdf?uni=4f7e7b429c56ccbc4bf56b3e93ebc47b
- [7] Artemio Almonacid *, A. A. (2018b, 7 mayo). Si se coopera de aprende más y mejor. Recuperado 13 abril, 2019, de <https://www.esdiario.com/532713300/Aprendizaje-cooperativo-en-la-educacion.html>
- [8] COMUNICAWEB. (s.f.). Top 10 mejores plataformas e-learning. Recuperado 20 abril, 2019, de https://www.comunica-web.com/verarticulo-plataformas-elearning_849.php
- [9] Moodle. (s.f.). Moodle [Foto]. Recuperado 24 abril, 2019, de <https://moodle.org/>
- [10] CANVAS. (s.f.). Guía del Esencial de Canvas | Guías de Canvas (es) [Foto]. Recuperado 2 mayo, 2019, de <https://es.guides.instructure.com/m/70283>
- [11] Rafael Alejandro Martínez-Campoblanco, R. A. M. C. (2016, 14 abril). El seguimiento y control de la gestión de alumnos mediante el uso de la plataforma Canvas LMS. Aportes desde la experiencia de tutoría virtual de la maestría en Educación de la Universidad de San Martín de Porres | Martínez-Campoblanco | EduTicInnova. Revista de Educación Virtual. Recuperado 2 mayo, 2019, de <https://www.aulavirtualusmp.pe/ojs/index.php/eduticinnova/article/view/800/630>
- [12] Chamilo. (s.f.). Chamilo.org – Asociación Chamilo [Foto]. Recuperado 2 junio, 2019, de <https://chamilo.org/es/>
- [13] Sakai. (s.f.). Sakai Learning Management System | Higher Education [Foto]. Recuperado 2 mayo, 2019, de <https://www.sakailms.org/>

- [14] WordPress. (2019, 22 mayo). LearnPress – WordPress LMS Conector [Foto]. Recuperado 25 mayo, 2019, de <https://es.wordpress.org/plugins/learnpress/>
- [15] Blackboard Inc. (s.f.). Blackboard | Servicios y tecnología para la educación [Foto]. Recuperado 2 mayo, 2019, de <https://es.blackboard.com/index.html>
- [16] Educativa. (s.f.). Campus | Educativa [Foto]. Recuperado 2 mayo, 2019, de <https://www.educativa.com/campus/>
- [17] OpenText. (s.f.). [Logo Imagen OpenText]. Recuperado 2 mayo, 2019, de <https://www.opentext.com/>
- [18] [Servicios Firebase] [Foto]. (s.f.). Recuperado 3 mayo, 2019, de <https://www.periscopix.co.uk/analytics/web-analytics/firebase-analytics/>
- [19] Google. (2017, 10 febrero). YouTube API Services Terms of Service (EMEA) | YouTube | Google Developers. Recuperado 15 mayo, 2019, de <https://developers.google.com/youtube/terms/api-services-terms-of-service-emea-es>
- [20] Firebase. (s.f.). Firebase Pricing. Recuperado 15 mayo, 2019, de <https://firebase.google.com/pricing/?hl=es-419>
- [21] Amazon Web Services. (s.f.). AWS | Almacenamiento de datos seguro en la nube (S3). Recuperado 16 mayo, 2019, de <https://aws.amazon.com/es/s3/>
- [22] Colaboradores de Wikipedia. (2019c, 20 abril). Red de distribución de contenidos - Wikipedia, la enciclopedia libre. Recuperado 16 mayo, 2019, de https://es.wikipedia.org/wiki/Red_de_distribuci%C3%B3n_de_contenidos
- [23] Amazon Web Services. (s.f.-b). Free Tier Terms and Conditions. Recuperado 17 mayo, 2019, de <https://aws.amazon.com/es/free/free-tier-terms/>
- [24] Universidad de Alicante. (s.f.). Modelo vista controlador (MVC). Servicio de Informática ASP.NET MVC 3 Framework. Recuperado 17 mayo, 2019, de <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>
- [25] Junta de Andalucía. (s.f.). Patrón Modelo Vista Controlador | Marco de Desarrollo de la Junta de Andalucía. Recuperado 17 mayo, 2019, de <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/122>
- [26] Patrón MVP [Publicación en un blog]. (2012, 2 febrero). Recuperado 18 mayo, 2019, de <https://www.imaginanet.com/blog/patron-mvp.html>
- [27] ARANGUREN, G. (s.f.). Comparación de los patrones de arquitectura MVC, MV-VM, MVP | Tópicos generales de Ingeniería de Software. Recuperado 20 mayo, 2019, de <https://ingsoftwarei2014.wordpress.com/category/comparacion-de-los-patrones-de-arquitectura-mvc-mv-vm-mvp/>
- [28] BeyondMaterial [Foto]. (s.f.). Recuperado 20 mayo, 2019, de <https://www.beyondjava.net/model-view-whatever>

- [29] Herrera, F. (s.f.). Curso Angular: De cero a experto creando aplicaciones [Archivo de vídeo]. Recuperado 10 enero, 2019, de <https://www.udemy.com/angular-2-fernando-herrera/>
- [30] Robles, V. (s.f.). <https://www.udemy.com/course/master-en-javascript-aprender-js-jquery-angular-nodejs-y-mas/> [Archivo de vídeo]. Recuperado 15 enero, 2019, de <https://www.udemy.com/course/master-en-javascript-aprender-js-jquery-angular-nodejs-y-mas/>
- [31] Domini Code. (2018, 12 diciembre). Angular 7, Firebase & Bootstrap 4 - YouTube [Archivo de vídeo]. Recuperado 2 febrero, 2019, de https://www.youtube.com/playlist?list=PL_9MDdjVuFjHgSCW7_15yoCAFUE5NAEjm
- [32] Amazon Web Services. (s.f.-c). Documentación de AWS. Recuperado 15 marzo, 2019, de <https://docs.aws.amazon.com/>
- [33] Google. (s.f.). YouTube Data API | Google Developers. Recuperado 20 mayo, 2019, de <https://developers.google.com/youtube/v3/>
- [34] Angular. (s.f.). Documentación Librería AngularFire2. Recuperado 7 marzo, 2019, de <https://github.com/angular/angularfire2>
- [35] Google. (s.f.-b). Firebase. Recuperado 20 abril, 2019, de <https://firebase.google.com/>
- [36] Blue Ocean. (s.f.). Recuperado 30 abril, 2019, de <https://jenkins.io/doc/book/blueocean/>
- [37] Khan Pathan, J. (s.f.). Contact Form on Lambda. Recuperado 7 junio, 2019, de <https://www.codekopp.com/blog/post/serverless-contact-form/>

10. ANEXOS

Anexo 1

Factura contratación dominio web



Tesys Internet S.L.U.
N.I.F: B-26309096
C/ Circunde 16 Pol. Industr. La Portalada
26006 Logroño (La Rioja) España

Datos de facturación:			
Marius iacob	SAbadell		
NIF: x8070284k	08206 Sabadell		
ID: 171438	BARCELONA - ESPAÑA		

Nº de factura:	Fecha de emisión:	Vencimiento:	Forma de pago:
2019B048850	13/04/2019	13/04/2019	TARJETA

Concepto	Importe	% Dto.	Importe total
Registro streamedu.es (13/04/2019-12/04/2020)	4.95	100% ⁽¹⁾	0.00 €
Protección de Dominio Registro .es streamedu.es (13/04/2019-12/04/2020)	3.95	100% ⁽²⁾	0.00 €
Registro .eu streamedu.eu (13/04/2019-12/04/2020)	4.95	79.8% ⁽³⁾	1.00 €
		Base Imponible	1.00 €
		IVA 21,00 %	0.21 €
		Total	1.21 €

⁽¹⁾ Descuento aplicado por: 2x1 en dominios

⁽²⁾ Descuento aplicado por: Protección de Dominio gratis 1 año al contratar un registro

⁽³⁾ Descuento aplicado por: Promoción dominios a 1 €

Sociedad inscrita en el Registro Mercantil de La Rioja, Tomo: 477, Folio: 76

Factura contratación servicios AWS



Account number:
999349766468

Bill to Address:
ATTN: Marius iacob
Sabadell
Barcelona
Sabadell , Barcelona , 08206 , ES

Request For Payment

Email or talk to us about your AWS account or bill, visit console.aws.amazon.com/support/

Bill Summary

Bill Number:	225888205
Bill Date:	June 3 , 2019
TOTAL AMOUNT DUE ON June 3 , 2019	\$0.68

This bill is for the billing period May 1 - May 31 , 2019

Greetings from Amazon Web Services, we're writing to provide you with a bill for your use of AWS services. Additional information about your bill, individual service charge details, and your account history are available on the Account Activity Page.

Summary	
AWS Service Charges	\$0.68
Charges	\$0.56
Credits	\$0.00
Tax *	\$0.12
Total for this bill	\$0.68

Anexo 2

Repositorio GitHub con el código de la aplicación:

<https://github.com/streamedu/app.git>

Cabe destacar que el repositorio no incluye las credenciales necesarias para acceder a los servicios implementados por la aplicación.