



FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

TÍTULO: Estrategias de comunicación en escenarios educativos híbridos: implementación y mejoras al sistema de notificaciones push de la aplicación de Moodle para AulasWebColegios de la UNLP

AUTORES: Ungaro, Lucas Gustavo

DIRECTOR: González, Alejandro Héctor

ASESOR PROFESIONAL: Romanut, Leandro Matías

Resumen

La Universidad Nacional de La Plata (UNLP) utiliza el entorno virtual basado en Moodle denominado AulasWebColegios para generar aulas virtuales para los colegios dependientes de la UNLP. Durante la pandemia, se observó un aumento en el uso de aulas virtuales y una demanda creciente de acceso vía dispositivo móvil. Se propone la puesta en marcha de una aplicación móvil personalizada basada en Moodle para mejorar la comunicación docente-estudiantes, enfocándose en la integración y mejora del módulo de notificaciones push. La misma puede descargarse para el celular y está disponible para docentes y estudiantes

Palabras Clave

Escenarios educativos híbridos, Entorno virtual de enseñanza y aprendizaje, Moodle, Notificaciones push, aplicación móvil, integración

Conclusiones

Es posible implementar y mejorar el módulo de notificaciones push en una aplicación personalizada basada en Moodle, es decir, utilizando el código fuente open source de la plataforma. Esta implementación permitió agilizar la comunicación entre docentes y estudiantes en el entorno virtual. Este trabajo sienta las bases para continuar avanzando sobre la mejora de la aplicación personalizada basada en Moodle. Si bien se hace foco en la mejora de un medio de comunicación específico, como son las notificaciones push, se busca fomentar la mejora de todas las demás funcionalidades y características faltantes en la aplicación móvil.

Trabajos Realizados

- Se realizó un análisis a nivel comunicacional del entorno AulasWebColegios, analizando las tecnologías de comunicación utilizadas, así como los requerimientos y percepciones de los usuarios (docentes y estudiantes de los colegios de la UNLP) respecto a dichas tecnologías.
- Se implementó un módulo de notificaciones push compatible con la aplicación móvil de Moodle
- Se mejoró el módulo de notificaciones push mediante el desarrollo de dos nuevas funcionalidades.

Trabajos Futuros

Extender la compatibilidad de la aplicación a sistemas operativos iOS y continuar con la mejora del módulo de notificaciones. Se destacan los siguientes trabajos futuros: explorar el registro y desarrollo de nuevos eventos de notificación dentro del sistema Moodle, desarrollo sobre la personalización completa para la función de silenciar notificaciones, mejora sobre la interacción con AirNotifier y desarrollo de un plugin en Moodle para la configuración genérica de las notificaciones por parte del administrador.

Universidad Nacional de La Plata

Facultad de informática



Estrategias de comunicación en escenarios educativos híbridos: implementación y mejoras al sistema de notificaciones push de la aplicación de Moodle para AulasWebColegios de la UNLP

Tesina de Licenciatura en Informática

Tesista: Ungaro, Lucas Gustavo
Director: Alejandro Héctor Gonzalez
Asesor Profesional: Leandro Matías Romanut

Julio 2023

AGRADECIMIENTOS

En primer lugar, quiero agradecer al arquitecto del universo, Dios Padre, por ser mi fortaleza en todo este trayecto, por su paciencia y amor incondicional. Gracias a ti, culminar mi carrera y ser un profesional adquiere sentido y propósito.

Agradecer a mis padres y hermano, Silvia, Gustavo y Mariano, por su apoyo incondicional. Ustedes hicieron que todo esto sea posible. Nunca voy a terminar de agradecerles.

A ella, Lucila, por motivarme constantemente a concluir mis estudios. Gracias por tu ayuda y apoyo en los momentos difíciles.

Por último, agradecer a mi director y asesor, Alejandro y Leandro, por haberme guiado y aconsejado durante todo este proyecto.

ÍNDICE DE CONTENIDO

AGRADECIMIENTOS	2
ÍNDICE DE CONTENIDO	3
INTRODUCCIÓN	5
Objetivos	7
Estructura del trabajo	7
CAPÍTULO 1: Contexto	9
1.1 Historización de los entornos virtuales y la educación a distancia de la Universidad de La Plata	10
1.2 Entornos virtuales en el contexto de pandemia	11
1.3 Entornos virtuales en contexto de pandemia en la Universidad Nacional de La Plata	12
1.4 Diagnóstico inicial	13
1.4.1 Análisis de viabilidad	14
1.4.1.1 Perspectiva de los usuarios sobre los medios de comunicación	15
1.4.1.2 Información general sobre los medios de comunicación que utilizan los usuarios	16
1.4.1.3 Requerimientos de los usuarios sobre los medios de comunicación	18
1.4.2 Análisis tecnológico	20
1.4.2.1 Estudio de las herramientas de comunicación en AulasWebColegios	20
1.4.2.2 Documentación sobre la puesta en marcha de una aplicación móvil basada en Moodle e incorporación de notificaciones push	25
CAPÍTULO 2: Marco teórico	28
2.1 Blended learning (b-learning)	28
2.2 Entorno virtual de enseñanza y aprendizaje (EVEA)	29
2.3 Arquitectura de software	30
2.3.1 Notificaciones Móviles y su clasificación	30
2.3.2 Tecnología push	32
2.3.3 Notificaciones push	34
2.4 Tecnologías específicas	34
2.4.1 Moodle	35
2.4.2 AirNotifier	38
2.4.3 Firebase Cloud Messaging (FCM)	40
2.4.4 Moodle App	43
2.4.5 Servicios web	44
2.4.6 Ionic Framework	45
CAPÍTULO 3: Desarrollo	47
3.1 Implementación del módulo de notificaciones push	48
3.2 Implementación de la funcionalidad “Borrar notificación”	52
3.3 Implementación de la funcionalidad “Destacar Notificaciones”	57
CAPÍTULO 4: Pruebas con los usuarios	65

4.1	Introducción	65
4.2	Pruebas	66
4.3	Encuesta de evaluación	67
CAPÍTULO 5: Conclusiones y trabajos futuros		69
BIBLIOGRAFÍA		74
Anexo 1: Formulario de sondeo para estudiantes		79
Anexo 2: Formulario de sondeo para docentes		83
Anexo 3: Formulario para validación de funcionalidad con los usuarios		87

INTRODUCCIÓN

La práctica educativa consta de ser un proceso en permanente construcción y reconstrucción, bajo el tiempo y el espacio institucional, una práctica que se desenvuelve entre lo constituyente y lo constituido. Así, el modo relacional del proceso de enseñanza aprendizaje, indica que, por un lado, se encuentra influenciado y determinado por diversos factores sociales, sociotécnicos, económicos, culturales, entre otros; y, por otro lado, se resalta el intercambio colaborativo entre docentes y estudiantes dentro de un marco institucional, en el que entran en juego los procesos comunicativos. Así, de acuerdo a Litwin (2012), es relevante señalar aquellas tecnologías creadas específicamente para la enseñanza, y aquellas que no lo son pero que son incorporadas en las instituciones educativas y las prácticas docentes. Siendo dos epistemologías diferentes, las tecnologías diseñadas para el proceso educativo, constan de pensarse en soporte didáctico y pedagógico, reconfigurando así el espacio de aprendizaje. Es decir, se destaca el uso pedagógico de la tecnología en cuestión (Fagina, 2018) en el marco de la era de la digitalización social (Mosquera Gende, 2019).

Entonces, puede decirse que el despliegue de las Tecnologías de Información y Comunicación (TIC) fueron y son una fuente de recursos para el sistema educativo, al impactar en los procesos, y en las transformaciones de los entornos de enseñanza. Estos entornos pueden definirse como entornos virtuales de enseñanza y aprendizaje (EVEA), los que utilizan tecnología educativa, y en diversas ocasiones, en combinación con la presencialidad educativa.

Teniendo en cuenta el desarrollo tecnológico en los entornos virtuales de enseñanza, el presente trabajo aborda la utilización de la aplicación oficial de Moodle¹ para dispositivos móviles, en el marco de AulasWebColegios², en instituciones educativas de educación media de la Universidad Nacional de la Plata (UNLP), con el fin de establecer posteriores mejoras a la misma. En este sentido se resalta que, este desarrollo tiene incidencia en la comunicación, en el proceso de enseñanza y de aprendizaje, dentro del entorno virtual

¹ <https://moodledev.io/general/app>

² <https://aulaswebcolegios.ead.unlp.edu.ar/>

educativo.

Desde hace tiempo, la Universidad Nacional de La Plata (UNLP) desarrolla propuestas de Educación a Distancia (EaD). Si bien en un principio se utilizó un desarrollo propio para llevar a cabo la propuesta de aulas virtuales, desde el 2008 se comienza a utilizar la plataforma Moodle.

Actualmente, la Dirección General de Educación a Distancia y Tecnologías (DGEaDyT) coordina las acciones del Sistema Institucional de Educación a Distancia (SIED)³, que entre otras tareas administra varios entornos virtuales basados en el sistema Moodle⁴.

Moodle como entorno virtual de enseñanza y aprendizaje (EVEA) facilita la constante demanda de personalización de las aulas virtuales, necesaria para abordar las diversas propuestas educativas. Esto es posible a través de las distintas opciones de configuraciones, herramientas disponibles, su capacidad para extenderse (gracias a su característica open source) y su posibilidad de integración con otros sistemas.

En este marco, durante la pandemia, la DGEaDyT registró un aumento en la creación y uso de aulas virtuales, creciendo el número de usuarios. Como medida de emergencia, una de las acciones que se llevó a cabo para evitar la sobrecarga de los entornos fue migrar las aulas virtuales de los colegios de la UNLP a un EVEA específico, denominado "AulasWebColegios".

En este caso particular, se registró una creciente demanda de acceso a la plataforma vía dispositivo móvil, los estudiantes utilizan frecuentemente el celular para acceder al recorrido de las aulas virtuales de los entornos de la UNLP (Jaime, 2020). Este acceso a través del celular, por parte de los estudiantes, se realiza mediante un navegador o la aplicación móvil de Moodle, estos medios están limitados en funcionalidad y no permiten una mejora continua.

Ante esta problemática, surgen tres aspectos a considerar. En primer lugar, la experiencia de la DGEaDyT en el desarrollo sobre entornos Moodle a lo largo de varios años. En segundo lugar, la necesidad de priorizar la compatibilidad con el EVEA AulasWebColegios, el cual está basado en Moodle. Y en tercer lugar, el uso intensivo de los celulares por parte de los estudiantes.

³ <https://sied.ead.unlp.edu.ar/>

⁴ <https://moodle.org/>

En base a estos aspectos, la propuesta de esta tesina consiste en mejorar la comunicación docente-estudiantes a través de la puesta en marcha de una aplicación móvil personalizada basada en Moodle, cuya personalización implica la integración y mejora del módulo de notificaciones push.

Objetivos

La presente tesina propone agilizar la comunicación docente-estudiante a través del análisis e implementación del módulo notificaciones en una aplicación personalizada basada en Moodle para dispositivos móviles, en el entorno AulasWebColegios de la Universidad Nacional de La Plata. Para cumplir el objetivo propuesto, se despliegan los siguientes objetivos específicos:

- Realizar un análisis del sistema de comunicación actual en el entorno de AulasWebColegios.
- Implementar un sistema de notificaciones push compatible con la aplicación móvil de Moodle.
- Identificar funcionalidades a mejorar del módulo de comunicación
- Desarrollar funcionalidades nuevas en el módulo notificaciones dentro de la app.

Estructura del trabajo

El presente trabajo se organiza en 6 capítulos que van desde análisis del contexto, el planteo del marco teórico, la presentación del desarrollo, las pruebas de testeo con los usuarios finales, la conclusión y los trabajos futuros.

En el capítulo 1 se plantea la información necesaria para comprender el entorno, la problemática y la justificación del trabajo. Comienza con una historización de los EVEAs en la UNLP, sobre su origen y evolución. Describe el impacto que tuvo la pandemia sobre el EVEA como herramienta en el ámbito de la educación, sus transformaciones y limitaciones. Destaca el caso particular de los EVEAs de la UNLP en el contexto pandémico, que medidas

se tomaron, los aspectos relevantes que llevaron a la propuesta de esta tesina y detalla un diagnóstico inicial sobre las potencialidades y limitaciones de los usuarios finales y las tecnologías subyacentes.

En el capítulo 2 se abordan los fundamentos y las tecnologías involucradas en el trabajo. Los primeros conceptos abordados ayudan a comprender el marco del desarrollo, mientras que los demás se centran en el propio desarrollo, desde las abstracciones teóricas hasta la definición de las tecnologías específicas.

En el capítulo 3 se explican las tres funcionalidades desarrolladas. La puesta en marcha de la aplicación móvil personalizada es inherente a la implementación y mejora del módulo de notificaciones.

En el capítulo 4 se detalla el proceso que se llevó a cabo para evaluar y validar las funcionalidades implementadas. Las pruebas que se realizaron, los datos recabados y los usuarios involucrados.

En el capítulo 5 y 6 se trata la conclusión y los trabajos futuros respectivamente.

CAPÍTULO 1: Contexto

En este capítulo se analizan los Entornos Virtuales de Enseñanza y Aprendizaje (EVEA) dentro de la Universidad Nacional de La Plata. Los EVEAs, aplicados a los procesos de enseñanza, que dependen de la Dirección General de Educación a Distancia y Tecnologías (DGEaDyT).

Se analiza no solo su génesis histórica sino también sus desarrollos, adaptaciones y modificaciones a partir del inesperado contexto de pandemia. Ante la crisis sanitaria, se crea el Programa de Ayuda a la Educación a Distancia (PAED). El rápido aumento de la cantidad de usuarios y de la demanda comunicacional de docentes y alumnos de los Colegios de la UNLP, requirió de la migración de sus aulas virtuales, generando un entorno específico llamado AulasWebColegios.

A partir de esto, surge la propuesta de que docentes y estudiantes puedan acceder al entorno de forma fácil e intuitiva a través del celular, planteándose entonces como objetivo la creación de una aplicación móvil basada en Moodle.

Este objetivo requirió de un diagnóstico inicial para evaluar potencialidades y limitaciones de la tecnología Moodle. Dicho diagnóstico se dividió en dos aspectos bien diferenciados, a saber: el análisis de los usuarios, docentes y estudiantes de los Colegios, y por otro lado el análisis de las herramientas Moodle implicadas: el entorno AulasWebColegios y la aplicación móvil de Moodle.

En base a las observaciones del diagnóstico se propuso, a partir de la App Moodle Oficial, implementar una nueva aplicación Moodle personalizada orientada a facilitar la comunicación, llamada "AulasWebColegiosNotificaciones". El trabajo de personalización consistió en la puesta en marcha y la mejora del módulo de Notificaciones Push.

1.1 Historización de los entornos virtuales y la educación a distancia de la Universidad de La Plata

La Universidad Nacional de La Plata (UNLP) desarrolla formalmente propuestas de Educación a Distancia desde la década de los 90. En el año 2002 se comenzaron a desarrollar los primeros cursos en opción pedagógica a distancia utilizando un desarrollo propio denominado WebLIDI. (Sanz & et al, 2004). Luego, en el año 2008 comenzó a utilizarse a nivel de presidencia de la Universidad, el entorno virtual de enseñanza y aprendizaje Moodle bajo la denominación de AVA (Ambiente Virtual de Aprendizaje) (Principe et al, 2009).

Desde el año 2017 se desarrolla el Sistema Institucional de Educación a Distancia de la UNLP (SIED-UNLP), como el conjunto de acciones, normas, procesos, equipamiento, recursos humanos y didácticos que permiten el desarrollo de propuestas a distancia. En la actualidad la Dirección General de Educación a Distancia y Tecnologías (DGEaDyT) administra el SIED-UNLP y varios entornos virtuales para alojar las diversas propuestas educativas, como e-learning y b-learning (de tipo híbrido), entre otras, que emergen de los distintos niveles (pregrado, grado, posgrado, extensión, capacitación). (Gonzalez & Martin, 2017)

Para la gestión de sus Aulas Virtuales, la DGEaDyT, adopta la plataforma e-learning Moodle a partir de 2014 bajo la denominación de AulasWeb. Esta Dirección administra los siguientes entornos (Gonzalez, 2019), que ofrecen diversas herramientas de trabajo y comunicación entre docentes y estudiantes, como ser foros de debate, creación de tareas, cuestionarios, entre otros:

- AulasWebGrado⁵
- AulasWebPosgrado⁶
- AulasWebFormacion⁷
- AulasWebColegios

⁵ <https://aulaswebgrado.ead.unlp.edu.ar/>

⁶ <https://aulaswebposgrado.ead.unlp.edu.ar/>

⁷ <https://aulaswebformacion.ead.unlp.edu.ar/>

- AulasWebOficios⁸
- Cursos Externos⁹

La estrategia de creación de los diferentes entornos AulasWeb radica en organizar las aulas de acuerdo al nivel de propuesta educativa, cantidad de participantes y cantidad de cursos. El contenido educativo es específico y las herramientas/recursos se adaptan a sus requisitos particulares.

1.2 Entornos virtuales en el contexto de pandemia

A partir del año 2020, la pandemia COVID-19 provocó una crisis sanitaria sin precedentes, lo que implicó la puesta en marcha de medidas de prevención como el Aislamiento Social Preventivo y Obligatorio (ASPO) y el posterior Distanciamiento Social Preventivo y Obligatorio (DISPO) mediante Decretos de Necesidad de Urgencia (DNU). Dichas medidas de prevención, han ocasionado tanto rupturas como continuidades en la dinámica educativa, y en especial se han profundizado debates sobre los medios y las herramientas para llevar adelante las prácticas docentes en el proceso de enseñanza y aprendizaje.

La reconfiguración del tiempo y el espacio institucional educativo a partir de la pandemia se conformó sobre desigualdades sociales y asimetrías previas, e implicó la necesidad de repensar lo educativo. Así, desde el punto de vista pedagógico y didáctico, los medios y herramientas disponibles, por ejemplo, las plataformas, pueden convertirse en obstáculos o aceleradoras del desarrollo cognitivo, al poner en marcha factores como la motivación de los estudiantes (Picherili y Tolosa, 2020).

Marquez (2021) evidenció que, en el contexto latinoamericano, la implementación de tecnologías de información y comunicación en el contexto de emergencia sanitaria se centró en la masificación de la educación a distancia, con la característica distintiva de asimetrías en los accesos y utilidades de los medios y herramientas. Particularmente, las estrategias regionales se centraron en:

⁸ <https://aulasweboficios.ead.unlp.edu.ar/>

⁹ <https://cursosexternos.ead.unlp.edu.ar/>

- Creación de plataformas TIC, donde el acceso y la conectividad son claves.
- Uso de medios masivos tradicionales (radio y televisión), desde los que se transmiten programas para dar continuidad a los procesos de enseñanza aprendizaje.
- Acompañamiento docente para la educación a distancia.

Así, los entornos virtuales de enseñanza sufrieron grandes transformaciones debido a su utilización intensiva y extensiva en el ámbito educativo, lo que implicó la necesidad de repensar la interacción comunicativa y abordar los cambios sucedidos en la comunicación entre docentes y estudiantes.

1.3 Entornos virtuales en contexto de pandemia en la Universidad Nacional de La Plata

Para atender los programas de educación digital, la Universidad Nacional de la Plata cuenta con la Dirección General de Educación a Distancia (DGEADyT), con aprobación del SIED-UNLP. De esta Dirección depende el Centro de Producción Multimedial (CEPROM).

Ante la emergencia surgida en 2020 con la suspensión de clases presenciales y su continuidad en formato virtual, se organizó el Programa de Ayuda a la Educación a Distancia (PAED). Su ejecución estuvo a cargo de la Secretaría de Asuntos Académicos de la UNLP, coordinada desde la DGEADyT. El objetivo era interactuar rápidamente y responder a la nueva demanda de educación remota, generada en el contexto de pandemia, de todas las Unidades Académicas de la institución.

Se crearon propuestas formativas como cursos de capacitación en formato webinar, formación en educación a distancia y continuidad pedagógica; y se brindaron asesorías técnico-pedagógicas a docentes y estudiantes. Además, se asesoró en desarrollo de materiales digitales interactivos, guías de clase, edición de videos educativos, organización de eventos totalmente en línea, con capacitaciones específicas para los Colegios de la UNLP. Se adquirieron licencias Zoom y Webex, y se ampliaron tanto los servidores como el hardware en general.

La cantidad de aulas virtuales creadas en el período marzo 2020 - mayo 2021 desde la DGEADyT es de 4.706 (Gonzalez, 2022).

Ante el aumento tanto de aulas virtuales como del número de usuarios, se redefinieron y ampliaron los entornos virtuales de la DGEADyT y se decidió la creación de un nuevo entorno, para descomprimir los ya existentes. Para apoyar específicamente a los colegios de la UNLP, migrando y transformando su contenido previo, se creó el entorno AulasWebColegios (Jaime, 2020).

Con la aparición de esta nueva situación surge la propuesta de que docentes y estudiantes puedan acceder al entorno de forma fácil e intuitiva a través del celular, planteándose entonces como objetivo la creación de una aplicación personalizada basada en Moodle para el acceso al nuevo entorno. Además, teniendo en cuenta las características y requerimientos específicos de los usuarios, se plantea como personalización la incorporación de notificaciones push en la aplicación.

Dado que los usuarios de este entorno tienen características y requerimientos específicos; la estrategia planteada fue la permanente revisión de la arquitectura y los contenidos del entorno, mediante sucesivos acercamientos a los docentes y estudiantes. Tomar nota de sus valoraciones sobre los aspectos pedagógicos y tecnológicos, con el fin de facilitar sus experiencias en la virtualidad.

1.4 Diagnóstico inicial

Este objetivo de creación de una aplicación Moodle que implemente notificaciones push requirió diseñar e implementar un diagnóstico inicial para evaluar potencialidades y limitaciones tanto de la experiencia de los usuarios como de las tecnologías a utilizar. Por lo que dicho diagnóstico se dividió en dos aspectos bien diferenciados.

Por un lado, se realizó el análisis de viabilidad con los usuarios, docentes y estudiantes de los Colegios de la UNLP, el cual se trata en el ítem 1.4.1.

Por otro lado, se llevó a cabo un análisis tecnológico que consistió en investigar el entorno AulasWebColegios a nivel comunicacional y realizar un examen detallado de la aplicación móvil de Moodle. Estos aspectos se tratan en el ítem 1.4.2

1.4.1 Análisis de viabilidad

Los usuarios finales del entorno virtual, son los docentes y estudiantes de los colegios de la UNLP. De la gestión del entorno y el acompañamiento a estos dos grupos de usuarios, surgen necesidades específicas que dan origen a las propuestas de mejora presentadas en este trabajo. Se hace necesario entonces sondear con ellos su viabilidad, y tomar nota del grado de factibilidad y apropiación de los nuevos recursos. Para conocer sus perspectivas y requerimientos respecto a los medios de comunicación en el marco del EVEA, se planificó una encuesta de sondeo.

Los colegios que participaron de dicha encuesta fueron:

- Bachillerato Bellas Artes “Francisco Américo de Santo”
- Liceo “Victor Mercante”
- Colegio Nacional “Rafael Hernández”
- Escuela Graduada “Joaquín V. Gonzalez”

Esta modalidad de diagnóstico se diseñó aprovechando el contexto pedagógico específico dado por la relación de docentes y alumnos, que por tratarse de dos grupos de usuarios bien diferenciados, permite evitar una encuesta genérica y ajustar las preguntas de sondeo según el tipo de usuario. Con lo cual en pos de analizar las particularidades de cada uno se decidió realizar dos encuestas específicas, una para estudiantes y otra para docentes (**ver Anexo 1 y 2**)

Las preguntas de ambas encuestas de sondeo se dividieron en tres categorías:

- 1.4.1.a Perspectiva de los usuarios sobre los medios de comunicación.
- 1.4.1.b Información general sobre los medios de comunicación que utilizan los usuarios.
- 1.4.1.c Requerimientos de los usuarios sobre los medios de comunicación.

A continuación, se analizan los datos más relevantes que se obtuvieron en cada una

de estas categorías, presentando datos y observaciones.

1.4.1.1 Perspectiva de los usuarios sobre los medios de comunicación

En el caso de los estudiantes, primero se consultó por la omisión de tareas y por su percepción sobre la comunicación de novedades en el aula virtual. Estos aspectos aluden a la efectividad de los medios de comunicación utilizados.

Los resultados obtenidos se presentan en los gráficos 1 y 2:

2 - ¿Alguna vez te olvidaste o no te enteraste de alguna tarea por no haber ingresado a tiempo a la plataforma AulasWebColegios?

271 respuestas

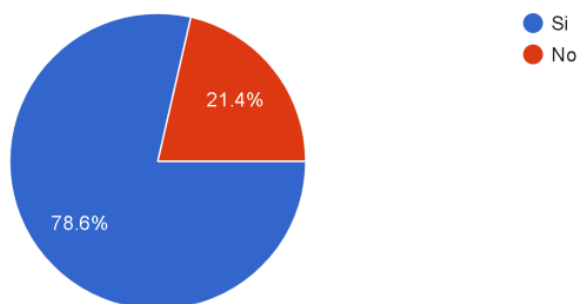


Gráfico 1. Distribución de omisión de tareas escolares por no ingresar a tiempo a la plataforma AulasWebColegios

3 - ¿Consideras difícil estar al tanto de las novedades y tareas nuevas dentro del curso?

271 respuestas

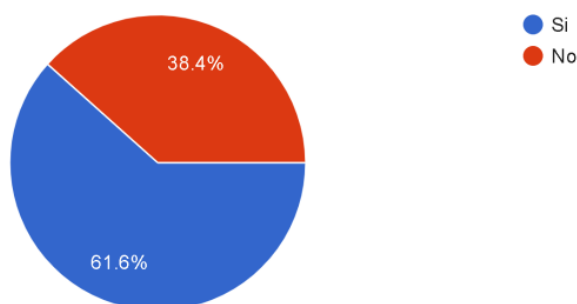


Gráfico 2. Dificultad para estar al tanto de las tareas y novedades del aula virtual

En el gráfico 1 se puede observar que la mayoría de los estudiantes alguna vez

perdió la posibilidad de completar una tarea por no haber ingresado a tiempo al entorno virtual, en concreto el 78.6%. Y en el gráfico 2 se muestra que más de la mitad (el 61.6%), considera difícil estar al tanto de las novedades del curso.

En el caso de los docentes, se indagó específicamente sobre su perspectiva respecto a adicionar más medios de comunicación.

Los resultados obtenidos se presentan en el gráfico 3:

3 - ¿Considera necesario agregar más vías de comunicación para notificar las novedades dentro de un curso en AulasWebColegios?

53 respuestas

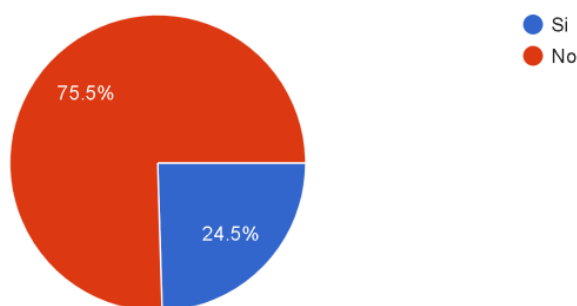


Gráfico 3. Percepción docente sobre la ampliación de las vías de comunicación.

En el gráfico 3 se puede observar que el 75.5% de los docentes no considera necesario agregar más vías de comunicación para notificar. No obstante, se destaca que un pequeño porcentaje (el 24.5%) apoya la idea de agregar más vías de comunicación. Estos docentes piensan en el beneficio que podría significar para el estudiante (dato en base a las respuestas de la pregunta 4 del formulario para docentes, **Anexo 2**)

1.4.1.2 Información general sobre los medios de comunicación que utilizan los usuarios

La información que se recolectó para esta categoría se sintetiza en el gráfico 4, donde tanto para estudiantes como para docentes se muestran los porcentajes positivos obtenidos.

Al preguntar por la comunicación utilizada a nivel software, se consultó específicamente por: “Whatsapp”, “Ingreso a la app Moodle”, “Correo electrónico”, “Otros” e “Ingreso al entorno virtual”. El parámetro “Otros” se refiere a redes sociales, blog o comunicación presencial. Y cuando se preguntó a nivel hardware, se consultó por: “Celular”, “Computadora”, “Notebook” y “Tablet”.

Medios / Canales de comunicación utilizados

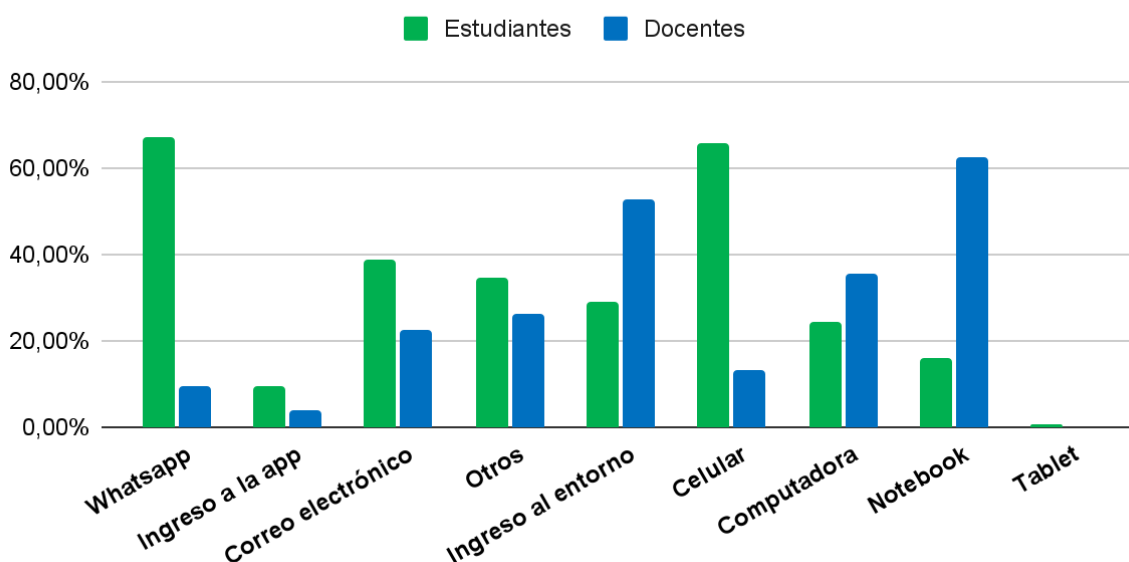


Gráfico 4. Medios / Canales de comunicación utilizados

Como se observa en el gráfico 4: La mayoría de los estudiantes encuestados (67.2%) se entera de las novedades dentro del aula virtual a través de Whatsapp (a nivel software) El 65.7% de los estudiantes prioriza el celular para ingresar al aula virtual (a nivel hardware) y en porcentajes muy similares el estudiante usa el correo electrónico, “otros” e ingresa al entorno (38.7%, 34.7% y 29.2% respectivamente) para enterarse de las novedades dentro del aula virtual.

La mitad de los docentes, específicamente el 52.8%, hace uso del entorno para enviar avisos e interactuar con los estudiantes. En menor medida utiliza el correo electrónico

y “otros”, 22.6% y 26.4% respectivamente (a nivel software).

El 62.3% de los docentes elige a la notebook como dispositivo para acceder al aula y un 35.8%, también, ingresa desde computadora de escritorio. Solo un 13.2% ingresa a través del celular (a nivel hardware).

Observaciones

- Se destaca, por parte de los estudiantes, el gran uso del celular y la utilización de Whatsapp.
- En el caso de los docentes, se destaca el gran uso de la notebook y la utilización del entorno para comunicar avisos y novedades.
- Se señala la gran diferencia en el uso del celular.
- Existe un bajo porcentaje tanto de docentes como de estudiantes que hacen uso de la “app Moodle”, pero esto se relaciona con que más de la mitad, en ambos casos, no conoce sobre la existencia la app (50.6% de estudiantes y 66% de docentes)
- De los estudiantes y docentes que usan el celular para ingresar a AulasWebColegios, una gran cantidad posee “Android” como sistema operativo. (65.7% y 39.6% respectivamente - Anexo “encuestas sondeo”)

1.4.1.3 Requerimientos de los usuarios sobre los medios de comunicación

Al igual que con los datos sobre “información general”, los datos recolectados para esta categoría se sintetizan en un gráfico general (gráfico 5), donde tanto para estudiantes como para docentes se muestran solo los porcentajes positivos obtenidos.

Este gráfico se descompone en tres partes, la preferencia por el medio para la recepción de las notificaciones (“Mail”, “Celular”, “Ambos” y/o “Ninguna”), la aprobación sobre la implementación de las notificaciones push y la unificación de los canales de comunicación.

Requerimientos

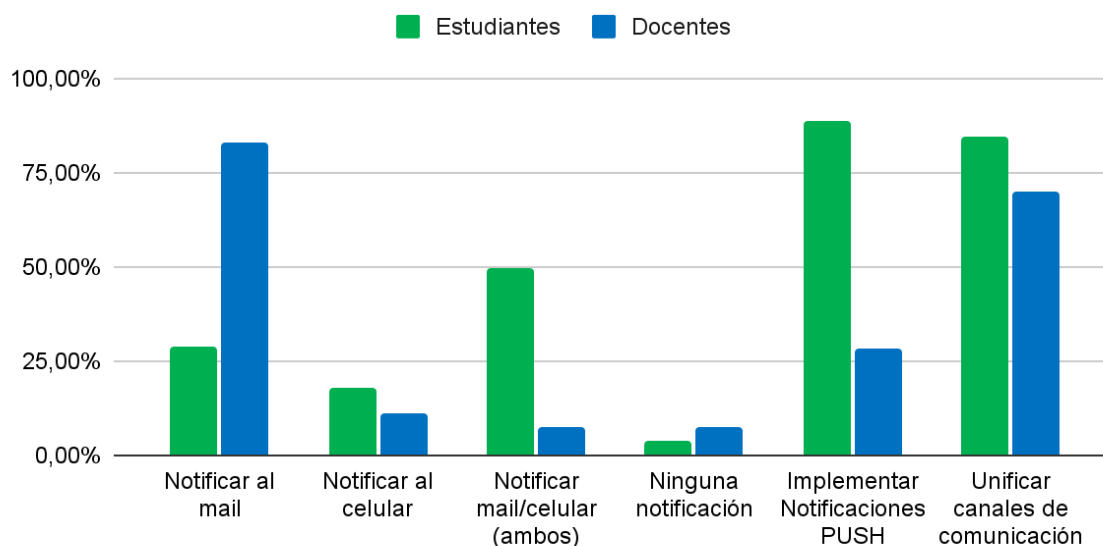


Gráfico 5. Requerimientos

El 88.6% de los estudiantes cree necesarias las notificaciones push y el 84.5% considera que sería de ayuda unificar los canales de comunicación existentes. El 49.8% prefiere recibir notificaciones tanto al mail como al celular. El 71.7% de los docentes cree innecesarias las notificaciones push, pero el 69.8% considera que sería de ayuda unificar los canales de comunicación. El 83% prefiere las notificaciones al correo electrónico.

Observaciones

- Ambos grupos tienen en común el enfatizar sobre la importancia de unificar los canales de comunicación.
- El 28.3% de los docentes está a favor de la implementación de las notificaciones a los dispositivos móviles, pero cabe aclarar que su aprobación pasa por ponerse del lado de los estudiantes, pensar en la ventaja que podría significar para ellos. Esto se relaciona con el dispositivo y el canal más utilizado (Notebook y correo electrónico).
- Los porcentajes obtenidos sobre los estudiantes tienen relación con el dispositivo y el canal más utilizado.

En términos específicos, hoy en día los estudiantes necesitan entrar al entorno virtual,

localizar el aula virtual correspondiente y allí buscar la notificación de la actividad/recurso particular. Contar con un sistema de notificaciones push, aportaría comodidad y seguridad a los estudiantes a la hora de estar al tanto, en tiempo real, de las novedades que surgen en los cursos.

Por lo cual dado el uso intensivo de celulares por parte de los estudiantes se hace necesario mejorar la comunicación docentes-estudiantes en la aplicación móvil de Moodle. La motivación del presente trabajo consta de la implementación de un sistema de notificaciones móviles que pueda aportar a la mejora de la comunicación en la app y lograr una mejor interacción de los estudiantes con el aula virtual.

1.4.2 Análisis tecnológico

Se examinaron las diferentes tecnologías disponibles para completar el desarrollo priorizando la compatibilidad con el sistema Moodle. En este sentido, se llevó a cabo un análisis detallado sobre la documentación oficial de Moodle, en específico sobre la versión 3.9.

El análisis consistió en examinar:

- Las herramientas de comunicación presentes en el entorno AulasWebColegios. (ítem 1.4.2.a)
- La documentación relacionada sobre la puesta en marcha de una aplicación móvil basada en Moodle y la incorporación de notificaciones push. (ítem 1.4.2.b)

1.4.2.1 Estudio de las herramientas de comunicación en AulasWebColegios

El entorno virtual cuenta con varias herramientas de comunicación, que son propias del sistema Moodle.

Hay tres maneras (a nivel sistema) en las que se puede dar la comunicación docente-estudiantes:

- Actividades
- Mensajería interna
- Notificaciones

Las actividades son elementos interactivos que se incorporan al aula virtual y buscan fomentar la participación y el aprendizaje involucrando a los estudiantes en diferentes formas de interacción, colaboración y evaluación. En específico, las actividades que priorizan la interacción/comunicación se pueden dividir en actividades sincrónicas y asincrónicas.

Actividades Moodle para comunicación sincrónica

- **BigBlueButton**¹⁰: Actividad (open source) que se integra al aula virtual y aporta al aprendizaje colaborativo. Permite crear salas/conferencias, en tiempo real, donde los participantes pueden compartir audio, vídeo, diapositivas, pizarra, chat y pantalla. Además, todas las sesiones pueden ser grabadas.

Los participantes pueden tomar diferentes roles y cada rol posee diferentes permisos.

- **Chat**: Actividad propia del sistema Moodle que permite interacción fluida mediante texto síncrono. Soporta direcciones URL, emoticonos, integración de HTML, imágenes ,etc. Todas las sesiones quedan registradas para verlas posteriormente, y pueden ponerse a disposición de los estudiantes. Pueden programarse sesiones periódicas que aparecerán en el calendario del curso.

Actividades Moodle para comunicación asincrónica

- **Foros**: permite presentar anuncios y generar debates, entre los participantes del aula, a partir de la publicación de comentarios como parte de un “hilo” dentro de un foro. Lo importante a destacar es la posibilidad que tienen los participantes de suscribirse a un foro (incluso un docente puede forzar la suscripción de sus estudiantes al foro), ya que luego de ello cada publicación será notificada a “Moodle app” y/o al correo electrónico.

¹⁰ <https://bigbluebutton.org/>

- **Board**¹¹: Esta actividad, que se incorpora como extensión al sistema, consiste en la utilización de un tablero/pizarra virtual en donde docentes y estudiantes pueden realizar aportes y compartir notas. Las publicaciones se realizan sobre “columnas” dentro del tablero y se pueden agregar textos, imágenes e insertar URLs.

En el caso particular del entorno AulasWebColegios, no se utiliza la mensajería interna que proporciona por defecto el sistema Moodle sino que se implementa un sistema similar a través de un plugin denominado Local Mail¹², que permite la comunicación directa dentro de la plataforma. Cuenta con la interfaz y la funcionalidad básica propia de un sistema de mensajería, como se muestra en la imagen 1 presenta una sección para “bandeja de entrada”, “destacados”, entre otros.

¹¹ https://moodle.org/plugins/mod_board

¹² https://moodle.org/plugins/local_mail



Imagen 1. Interfaz gráfica del plugin Local Mail

Este plugin suma otras características, tales como:

- Capacidad para separar los mensajes por cursos. (En la imagen 1 se pueden ver, sobre la interfaz, a los cursos “TEST”, “modelo” y “PMI”)
- Comunicación privada: los mensajes sólo son visibles para el destinatario y el remitente.
- Historial de mensajes: queda un registro de los mensajes enviados por los usuarios.
- Notificaciones: Cuando se remite un mensaje, el mismo se envía por mensajería interna y también se genera una notificación al email del destinatario. Es muy importante destacar que el plugin de mensajería utiliza el “sistema de notificaciones” para enviar al email (en específico, se está “notificando el email”).

Las notificaciones en Moodle se generan a partir de “eventos”. Los eventos son acciones registradas por Moodle que ocurren dentro del sistema. Estos, pueden ser generados automáticamente por el sistema o por la interacción del usuario con el entorno.

Del conjunto de eventos totales solo algunos son notificados a los usuarios. Esta es una decisión de diseño que toma Moodle buscando evitar la sobrecarga de información.

- En *Administración del sitio>Informes>Listado de eventos* se encuentra la lista de eventos totales que registra el sistema.
- En *Administración del sitio>Mensajería>Ajustes de notificación* está el listado de eventos que son notificados a los usuarios.

En la imagen 2 se muestra la sección preferencias de notificaciones. Se puede ver como los eventos a notificar se agrupan por contextos (Tarea/Módulo de encuesta/Foro).

Preferencias de notificación

Desactivar las notificaciones

	Web		Email		Móvil	
Tarea	En línea	Descorrido	En línea	Descorrido	En línea	Descorrido
Notificaciones de tareas	No	No	No	No	Sí	Sí
Módulo de encuesta	En línea	Descorrido	En línea	Descorrido	En línea	Descorrido
Notificaciones de encuesta	No	No	No	No	Sí	Sí
Recordatorio de encuesta	No	No	No	No	Sí	Sí
Foro	En línea	Descorrido	En línea	Descorrido	En línea	Descorrido
Mensajes suscritos del foro	No	No	No	No	Sí	Sí
Resúmenes del foro suscrito	No	No	No	No	Sí	Sí

Imagen 2. La sección preferencias de notificaciones dentro del entorno

Además, en la imagen 2 se visualiza como por cada evento se ofrece la posibilidad de habilitar el envío a tres medios diferentes (Web, Email y Móvil).

Habilitar la notificación al Email implica el envío de un correo electrónico a la cuenta que haya agregado el usuario en su perfil personal dentro de la plataforma. El sistema Moodle utiliza un servidor SMTP, previamente configurado por el administrador, para realizar el envío del mensaje.

Las opciones Web y Móvil implican el envío de una notificación push al navegador web y/o a la aplicación Moodle. En el caso de la notificación web, el sistema utiliza la tecnología “web push” para hacer el envío del mensaje y en el caso de la notificación al móvil se utiliza AirNotifier¹³ junto con Firebase Cloud Messaging¹⁴ (FCM). En ambos casos la notificación se visualiza como una ventana emergente que muestra el aviso correspondiente.

Se destaca que las notificaciones son el único medio disponible en el entorno AulasWebColegios que permite la comunicación desde la plataforma hacia el exterior (por fuera del entorno).

En específico, las notificaciones al móvil son la única forma de notificación directa hacia un dispositivo móvil. Porque si bien las notificaciones al email pueden llegar al celular, es necesario que el usuario utilice en su teléfono el mismo email que usa en el entorno y que tenga habilitada las notificaciones push correspondientes a la aplicación móvil de correo electrónico.

1.4.2.2 Documentación sobre la puesta en marcha de una aplicación móvil basada en Moodle e incorporación de notificaciones push

Moodle ofrece de manera oficial y gratuita la aplicación móvil “Moodle App”, pero la misma carece de mucha funcionalidad. A continuación se mencionan algunas limitaciones:

- Soporte técnico: Si bien la comunidad Moodle brinda documentación y foros de ayuda, no es factible la comunicación directa con un desarrollador oficial.

¹³ <https://github.com/dcai/airnotifier>

¹⁴ <https://firebase.google.com/>

- Integración con otros sistemas: No es posible agregar herramientas externas, como herramientas de trabajo colaborativo o comunicación.
- Personalización avanzada: los usuarios se ven limitados a la hora de realizar cambios en la apariencia, diseño o estructura de la aplicación según sus necesidades.
- La configuración de la interfaz gráfica se encuentra restringida, no es posible agregar logotipos personalizados, colores, estilos o diseños específicos de una institución en particular.
- Los recursos y actividades se encuentran condicionados, en características como “restricciones de acceso” o “condiciones de finalización”.
- No es posible deshabilitar el acceso sin conexión, el uso de bloques, los comentarios y las etiquetas.
- El único paquete de idioma disponible es el inglés.
- Funciones avanzadas: existen varias funcionalidades complejas que se encuentran restringidas, como:
 - El acceso offline al contenido del curso implica descargar solo dos cursos completos.
 - Registrando el sitio moodle solo se pueden habilitar las notificaciones push a 50 dispositivos y las mismas no pueden ser personalizadas. Registrar el sitio implica compartir todos los datos del entorno con la empresa Moodle.
 - Login por QR

Moodle plantea cuatro formas para solventar estas limitaciones en diferentes niveles:

- El plan pago “Pro”, supone utilizar la aplicación móvil oficial y que Moodle habilite ciertas características propias de este nivel.
- El plan pago “Premium”, funciona de la misma manera que el plan pro pero involucra más características disponibles. Existe la posibilidad de acceder a este plan de manera gratuita calificando como organización sin fines de lucro.

- La opción paga Branded Moodle App¹⁵, consiste en contactar con la empresa para crear una aplicación totalmente a medida, incluyendo la publicación en las tiendas de Android y iOS.
- La opción de crear una aplicación móvil personalizada, consiste en ir desarrollando cada una de las características faltantes a partir del código fuente de la aplicación oficial de Moodle, aprovechando que el mismo es de código abierto.

La opción de personalizar la aplicación móvil aporta mayor flexibilidad, control, adaptabilidad y la oportunidad de aprender sobre el desarrollo de aplicaciones móviles. Ya que al desarrollar las funcionalidades faltantes de acuerdo a las necesidades específicas de los usuarios se logra mejor personalización apuntando al desarrollo propio y a la mejora continua.

Igualmente, se destacan las ventajas de las otras tres opciones:

- No se requiere invertir tanto tiempo y recursos.
- La responsabilidad respecto al mantenimiento, los problemas técnicos y la compatibilidad están sujetos a Moodle.
- Acceso a las actualizaciones oficiales de Moodle.

Por otra parte, una opción para implementar notificaciones push en una aplicación personalizada basada en Moodle, priorizando la compatibilidad con el mismo, consiste en la puesta en marcha del servicio de AirNotifier junto al servicio de Firebase Cloud Messaging, puesto que existe una integración oficial dentro del sistema web Moodle.

¹⁵ <https://moodle.com/branded-app/>

CAPÍTULO 2: Marco teórico

En este capítulo se presentan los conceptos necesarios para comprender los fundamentos y las tecnologías involucradas en este desarrollo.

En primer lugar, se menciona el aprendizaje híbrido, debido a que el desarrollo busca adaptarse a un escenario de este tipo. Este escenario, o también conocido como enfoque, tiene lugar en el entorno basado en Moodle, AulasWebColegios.

El entorno virtual de enseñanza y aprendizaje (EVEA) es otro concepto fundamental, ya que hace referencia al sistema Moodle (González y Chicote, 2016). Se presenta un punto de vista particular que busca ayudar a comprender el concepto en su totalidad.

Posteriormente, se conceptualiza lo referido al desarrollo, la incorporación de notificaciones push en la aplicación personalizada y la mejora del mismo a través de la creación de dos nuevas funcionalidades. Se plantean los conceptos sobre la arquitectura de software involucrada, y luego se presentan las tecnologías en particular que se utilizaron para este proyecto.

2.1 Blended learning (b-learning)

Desde el año 2020 se comenzó a utilizar con mayor énfasis la palabra “híbrido” como respuesta a escenarios de enseñanza que intercalan presencialidad y virtualidad. Según Andreoli (2021) este concepto no es nuevo en educación y hace referencia a otros términos que describen situaciones similares como blended learning (b-learning) o aprendizaje combinado.

El b-learning procura tomar en el contexto de aplicación las mejores prácticas de la educación presencial y las mejores prácticas de educación a distancia y combinarlas. Entre esas prácticas podemos mencionar el uso del aula virtual como espacio de comunicación y desarrollo de tareas para los tiempos asincrónicos. También, el b-learning se puede plantear, según Ibañez (2018), como una estrategia que combina el aprendizaje presencial y el aprendizaje a distancia. Y remarca algunos de los aspectos que la educación a distancia aporta al b-learning, como la posibilidad de ampliar la comunicación a través de las

Tecnologías de la Información y Comunicación (TIC), la creación de nuevas formas de interacción docente-alumno, y la flexibilidad en términos de tiempo y espacio.

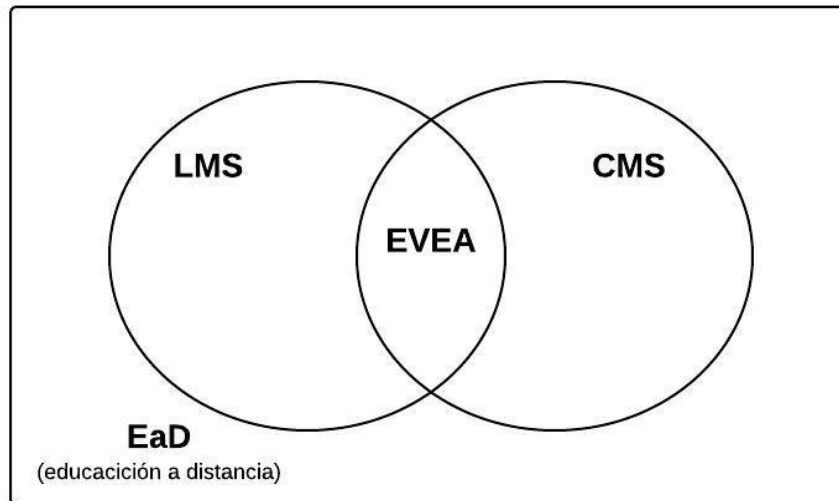
2.2 Entorno virtual de enseñanza y aprendizaje (EVEA)

De acuerdo a González y Chicote (2016) los EVEA refieren a espacios comunicativos que “integran un extenso grupo de materiales y recursos desarrollados para facilitar y optimizar el proceso de enseñanza y, por ende el aprendizaje de los alumnos mediados ambos por TIC” (p. 2). Además, los autores destacan las herramientas que proporcionan estos entornos, que permiten la administración, seguimiento y evaluación de actividades educativas en diferentes modalidades.

Para completar la comprensión de este concepto se plantea el siguiente punto de vista: por un lado, un sistema de gestión de enseñanza (LMS) es una plataforma diseñada para administrar, dar seguimiento y evaluación a diferentes actividades educativas. Romanut (2016) menciona que un LMS “es un software online que permite administrar, distribuir, monitorear, evaluar y apoyar las diferentes actividades previamente diseñadas y programadas dentro de un proceso de formación completamente virtual (e-learning) o de formación semipresencial (b-learning)” (p. 38). Por otro lado, un sistema de gestión de contenidos (CMS), según Cabrera Medina, Sanchez Medina y Rojas (2016) se puede entender como a una aplicación que permite la creación, la publicación y la gestión de contenido en línea. Los CMS se emplean para administrar diferentes tipos de contenido como texto, imágenes, videos y documentos, en un sitio web. Además, proporciona herramientas para la gestión de usuarios y la personalización del sitio a nivel diseño.

Ahora bien, un EVEA comparte las características de un LMS y un CMS. Ya que el mismo integra las funcionalidades de gestión para las actividades educativas y las funcionalidades de gestión de contenido en línea. Si consideramos los conjuntos LMS y CMS, el concepto EVEA se ubicaría en la intersección de ambos como se muestra en la imagen 3.

Estas características que se combinan en un EVEA le ofrecen facilidad al docente para diseñar materiales educativos, y al mismo tiempo facilita la interacción y el aprendizaje del estudiante.



*Imagen 3. La intersección entre el conjunto LMS y CMS
Fuente. Gráfico basado en Mujica Sequera, 2020*

2.3 Arquitectura de software

Según Fernandez (2006), la arquitectura de software abarca lo referido a la estructura del software, las piezas que lo componen, su organización y cómo interactúan entre sí. Este concepto proporciona una mirada que ayuda a comprender la complejidad y la comunicación de los componentes del sistema. En este caso particular, nos sirve para comprender la estructura y el funcionamiento de las notificaciones push en el marco de una aplicación móvil basada en Moodle, permitiendo identificar el modelo y los componentes implícitos en el diseño.

Siguiendo esta perspectiva, en este apartado se abordaron temas como:

- Notificaciones móviles y su clasificación
- Tecnología push, estructura estándar y componentes
- Notificaciones push

2.3.1 Notificaciones Móviles y su clasificación

De acuerdo con Vila Ruiz (2015), las notificaciones móviles son mensajes de alerta que se envían al dispositivo del usuario para brindar información puntual sobre la aplicación

o el sistema, o para dar aviso sobre recordatorios o eventos. Son un medio para la comunicación directa y eficiente, que beneficia a la fidelización de usuarios. Una vez que llega la notificación, el usuario puede presionar sobre la misma para abrir la aplicación o ejecutar una acción determinada.

Hay dos cuestiones a considerar sobre las notificaciones móviles en general, las formas de envío y su clasificación. Las notificaciones pueden ser enviadas por mensajes de texto (SMS), correo electrónico o a través del sistema de notificaciones push presente en los smartphones y tablets. En cuanto a la clasificación, como menciona Zuazo (2018), Google¹⁶ y Apple¹⁷ forman parte de las plataformas más importantes a nivel mundial, y cada una clasifica a las notificaciones móviles de diferente manera, pero ambas coinciden en las siguientes clasificaciones:

Procedencia

- Tipo locales: Las notificaciones que provienen de la misma aplicación móvil. La aplicación programa la notificación con el sistema operativo del dispositivo local, configura los detalles de la notificación localmente y pasa esos detalles al sistema, que luego maneja la entrega de la notificación cuando su aplicación no está en primer plano.

Las notificaciones locales brindan una forma de alertar al usuario cuando su aplicación tiene nueva información para presentar. Por ejemplo, una aplicación que extrae datos de un servidor en segundo plano puede programar una notificación local cuando se recibe información interesante. Las notificaciones locales también son adecuadas para aplicaciones como el calendario y la lista de tareas que necesitan alertar al usuario en un momento específico o cuando se alcanza una ubicación geográfica específica.

- Tipo remotas: Las notificaciones provienen de un servidor remoto. Los datos provienen de un servidor y se envían a los dispositivos de los usuarios a través de los servicios de notificaciones móviles. En el caso de los dispositivos con sistemas

¹⁶ <https://www.google.com/>

¹⁷ <https://www.apple.com/>

operativos iOS, estos interactúan con el servicio de APNs¹⁸ (Apple Push Notification service) y en el caso de Android lo hacen con FCM. Las notificaciones remotas se pueden transmitir en cualquier momento, incluso cuando la aplicación no se está ejecutando.

Propósito

- Tipo transaccional: Las notificaciones transaccionales proporcionan contenido que un usuario debe recibir en un momento específico para realizar una de las siguientes acciones: “Permitir la interacción de persona a persona”, “Responder a un evento o recordatorio” o “Controlar o resolver estados transitorios del dispositivo”.
- Tipo no-transaccional: Las notificaciones no transaccionales deben ser opcionales, ya que es posible que no resulten atractivas para todos los usuarios. Hay dos formas de manejar la opcionalidad:
- Opt-out: Los usuarios reciben notificaciones de “opt-out” de forma predeterminada, pero pueden dejar de recibirlas desactivando una configuración. Se relacionan con el contexto del usuario y aportan un valor concreto al mismo.
- Opt-in: Los usuarios solo reciben notificaciones de “opt-in” activando una configuración en su aplicación.

En adición a esto, se explica la tecnología push, que como se mencionó anteriormente es una forma en la que se pueden enviar las notificaciones móviles.

2.3.2 Tecnología push

Permite enviar notificaciones directamente a nuestros dispositivos móviles sin que el usuario tenga que hacer una solicitud activa para ello. Las notificaciones aparecen directamente en la pantalla de inicio o en la barra de notificaciones del dispositivo, captando la atención del usuario.

Según Quizhpe (2019), la tecnología push es una forma de comunicación

¹⁸ https://developer.apple.com/documentation/usernotifications/registering_your_app_with_apns/

cliente-servidor, que implica varias características, como:

- El servidor es quien inicia la comunicación.
- El servidor envía mensajes (push, empuja información) a los clientes previamente suscritos al servicio de notificaciones. De esta forma el servidor sabe a dónde enviar la push y con quién generar, luego, una comunicación constante a través de un socket.
- Aunque la aplicación móvil esté cerrada o en segundo plano la notificación llegará al dispositivo.
- Los clientes se suscriben al servidor de aplicaciones mediante un token de registro que obtienen del servicio de mensajería móvil correspondiente.

Por otra parte, tanto Quizhpe (2019) como Aguín (2011) plantean que una estructura push estándar, especialmente cuando se trata de notificaciones, se constituye de un servidor de aplicación, un servidor push y un terminal móvil. En la imagen 4 se puede visualizar esta estructura junto a su funcionamiento estándar representado por una serie de pasos.

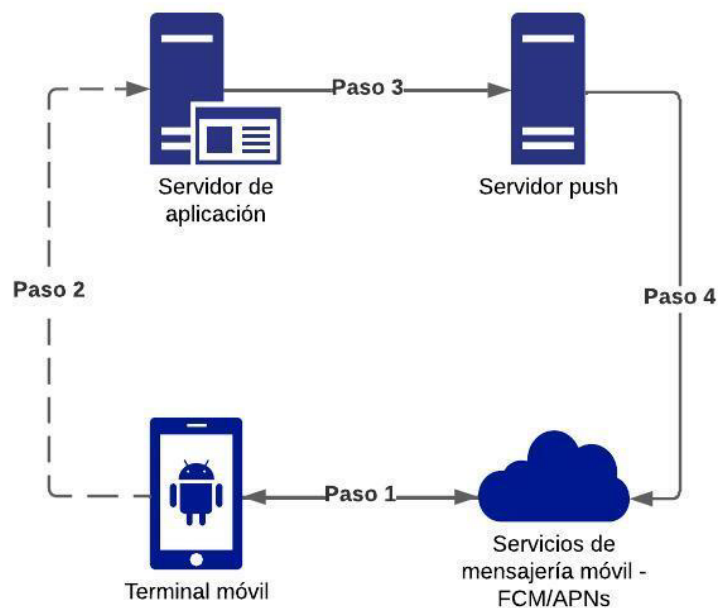


Imagen 4. Estructura y funcionamiento push estándar

Fuente. Basado en el gráfico de Lenin Alcivar Gordillo Quizhpe y en el gráfico de José Manuel Alarcón Aguín, 2019 y 2011 respectivamente.

A continuación se detallan cada uno de los elementos involucrados en la imagen 4:

- Un servidor de aplicación: Quien genera la notificación push, es decir, donde se originan las notificaciones y se decide a qué instancias de aplicación se deben enviar.
- Un servidor push: Quien gestiona la transmisión de las notificaciones, que recibe del servidor de aplicación, para su posterior envío a los dispositivos móviles. Este envío hacia los dispositivos se lleva a cabo mediante protocolos de comunicación, como APNs o FCM.
- Un terminal móvil: Quien recibe la notificación. En concreto, la instancia de la aplicación presente en el terminal actúa de receptor de las mismas.

La serie de pasos planteados en la imagen 4 sería la siguiente:

- Paso 1: La aplicación móvil recibe del servicio de mensajería correspondiente un token de registro. Este paso solo se realiza la primera vez que se inicia la aplicación.
- Paso 2: El token de registro, que identifica unívocamente a la instancia de aplicación, es enviado al servidor de aplicación para que una aplicación concreta lo almacene (suscripción al servicio push).
- Paso 3: Cuando la aplicación web lo decide, emite las notificaciones al servidor push para que este gestione el envío final de las mismas. Junto a la notificación se transmite el token de registro.
- Paso 4: Una vez que el servidor push recibe la notificación verifica si el terminal al cual corresponde el token de registro se encuentra online. Si el dispositivo no se encuentra conectado a internet se encola la notificación, caso contrario se envía.

2.3.3 Notificaciones push

En relación a lo expuesto anteriormente, el concepto de notificaciones push se puede entender como aquellas notificaciones móviles, necesariamente remotas, que ocurren en el marco de una tecnología push.

2.4 Tecnologías específicas

Las tecnologías que van a ser mencionadas encajan con la arquitectura de software planteada:

- Moodle: Es la aplicación web alojada en un servidor de aplicaciones, que genera las notificaciones push.
- AirNotifier: Es el servidor push que gestiona las notificaciones, interactuando con Moodle y los dispositivos móviles.
- Firebase Cloud Messaging: Es el protocolo de comunicación que utiliza el servidor push para transmitir a los dispositivos móviles.
- Moodle App: Es la instancia de aplicación que se suscribe al servicio push que ofrece la aplicación web Moodle. Quien recibe las notificaciones.
- Servicios web: Es el medio de comunicación entre la aplicación móvil y la aplicación web.
- Ionic Framework¹⁹: Es la herramienta con la que está programada Moodle App.

En este punto, cabe destacar que se aborda la explicación de Moodle App porque la puesta en marcha de la aplicación personalizada se basa en ella. A su vez, se trabaja el concepto de servicios web ya que es el medio que utiliza la aplicación para suscribirse al servicio de notificaciones push y por donde interactúan las dos funcionalidades nuevas desarrolladas con Moodle.

2.4.1 Moodle

Es una plataforma de tipo EVEA que permite la gestión de aulas virtuales y usuarios facilitando a los educadores trabajar en distintos ámbitos educativos e instancias académicas a través de diferentes herramientas y contenidos (González y Chicote, 2016). También, se puede pensar como un aplicación web multiplataforma que permite crear espacios de aprendizaje personalizados en línea (Romanut, 2016).

Se destaca su característica de código abierto, la cual posibilita que cualquier desarrollador pueda adaptar, extender o modificar su código. Se encuentra bajo la Licencia Pública General GNU²⁰.

Existen tres elementos clave en Moodle: los usuarios, las aulas virtuales y las distintas herramientas.

¹⁹ <https://ionicframework.com/>

²⁰ <https://moodledev.io/general/license>

Usuarios en Moodle

Si bien la plataforma se centra en docentes y estudiantes, es necesario usuarios que administren el entorno y que los docentes puedan asumir diferentes roles de acuerdo a los requerimientos del curso. Por ello, en el sistema, los usuarios pueden tomar diferentes roles que determinan sus permisos dentro del mismo. Un usuario puede tener uno o más roles y cada rol puede tener uno o más permisos.

Los roles estándar en Moodle son los siguientes:

- Administrador del sitio: Posee todos los permisos. Si bien el usuario que posea este rol puede hacer cualquier tipo de alta, baja o modificación (ABM), generalmente, es un rol que se utiliza para hacer cambios a nivel sistema.
- Manager (Gestor): Es un rol de administrador de nivel inferior. La diferencia con el "administrador del sitio" es que a este rol se le pueden modificar sus permisos.
- Creador de cursos: Es un rol específico para el usuario encargado del ABM (alta, baja y modificación) de los cursos.
- Profesor: Este rol otorga casi todos los permisos a nivel de curso (aula virtual). Esto incluye a todos los permisos que involucren a la actividades, los recursos y a los usuarios (con sus roles y permisos).
- Profesor no-editor: Es el rol de profesor pero sin permisos de edición. Puede ver, calificar, interactuar pero no editar/modificar actividades o recursos.
- Estudiante: Participa en el curso, interactúa con los recursos y actividades propuestas por el profesor.
- Invitado: Este rol te permite visualizar contenido pero no interactuar. Es un rol de "solo lectura".

Herramientas en Moodle

Las herramientas sirven para gestionar la interacción con los usuarios y administrar el

contenido dentro del aula virtual, se pueden configurar y adaptar a la propuesta de enseñanza. Estas herramientas se dividen en dos categorías: las actividades y los recursos. A continuación se describe cada una:

- Las actividades tienen como objetivo buscar la interacción del estudiante con otros estudiantes y/o el profesor. Las principales actividades Moodle son:
 - Tarea: permite cargar archivos o tareas que posteriormente pueden ser calificadas por los profesores.
 - Chat: Permite el intercambio de mensajes, de manera sincrónica, entre los participantes del curso.
 - Foro: Permite la comunicación asincrónica entre los participantes, a través de aportes en un hilo de debate (publicaciones en el foro).
 - Glosario: Permite crear y mantener una lista de definiciones.
 - Taller: Permite evaluar a los participantes de a pares.
 - Wiki: Permite crear una colección de páginas de manera colaborativa entre los participantes.
 - Base de datos: Permite a los participantes crear, mantener y buscar dentro de un banco de entradas. Las entradas pueden tener diferentes formatos (imágenes, archivos, urls, texto).
 - Consulta: Permite que el profesor pueda cargar una pregunta y una variedad de respuestas de tipo opción múltiple.
 - Cuestionario: Herramienta que permite diseñar exámenes. Estos exámenes se componen de diferentes tipos de preguntas (Verdadero/Falso, Opción múltiple, Ensayo, entre otros) brindando la posibilidad de generar diferentes formatos de examen (como trabajos complejos de autoevaluación con retroalimentación detallada o “multiple choice”).
 - SCORM: Permite la carga de paquetes SCORM (objetos pedagógicos estructurados).
 - Encuesta: Permite recolectar datos de los estudiantes de manera eficiente.
 - H5P: Permite la carga de contenido H5P desde el “banco de contenidos” (dentro del aula virtual) o desde la página oficial “h5p.com”.

- BigBlueButton: Permite generar instancias de videollamadas para la comunicación sincrónica dentro de Moodle.
- Los recursos son objetos que se utilizan para cargar contenido, como un archivo o enlace. Los principales recursos Moodle son los siguientes:
 - Archivo: permite la carga de un elemento tipo imagen, documento PDF, hoja de cálculo, archivo de sonido o video.
 - Etiqueta: imagen o texto útil para separar los recursos o actividades dentro del curso.
 - Carpeta: Permite agrupar archivos. A su vez, una carpeta puede contener subcarpetas.
 - Página: permite generar una página HTML a partir de un editor propio del sistema.
 - Libro: permite agrupar un conjunto de páginas, es decir, se forma un libro de páginas.
 - Página de contenido IMS: permite añadir material estadístico desde otros recursos en el formato IMS estándar.
 - URL: permite la carga de un enlace hacia un sitio externo. Un ejemplo sería un enlace a un video en la plataforma YouTube²¹.

Aula virtual

Es el sitio donde los usuarios, las actividades y los recursos convergen. En este espacio, los estudiantes acceden a las herramientas proporcionadas por los docentes para llevar a cabo actividades de aprendizaje y comunicación en línea.

Un aula virtual puede ir desde un simple contenedor que almacena contenido hasta un conjunto complejo de tareas en donde la enseñanza va en aumento, esto depende del diseño y la propuesta del docente.

El diseño del aula no está compuesto solo de actividades y recursos, también existen

²¹ <https://www.youtube.com/>

otros aspectos de configuración, como:

- Formato del curso: Las aulas se pueden dividir en secciones, pestañas o mosaicos.
- Método de matriculación: existen diferentes métodos para registrar a los estudiantes en el aula.
- Gestión de los participantes: Los participantes, sean docentes o estudiantes, se pueden organizar en “grupos”. Las secciones, las actividades y los recursos se pueden restringir para trabajar en grupos.

2.4.2 AirNotifier

Es importante resaltar que Moodle en su versión 3.9 añade de manera oficial la posibilidad de conexión con un servidor AirNotifier, brindando estabilidad y compatibilidad con la integración de notificaciones push.

Malek (2020) define a AirNotifier como un servidor de notificaciones push que gestiona el envío de notificaciones, en tiempo real, a aplicaciones móviles y de escritorio.

Visto de otra manera, AirNotifier actúa como gateway entre la aplicación que genera la notificación push y los servicios móviles de mensajería (como FMC o APNs).

Algunas características de este servicio son:

- Proporciona una interfaz gráfica de usuario vía web, esto permite claridad a la hora de visualizar los informes de las notificaciones (envíos, recepción y posibles errores).
- Comunicación a través de API REST.
- Compatibilidad con sistemas operativos Android, iOS y Windows 10 (aplicaciones de escritorio / Navegador Chrome).
- Soporte para los protocolos FCM, APNs y WNS.
- Es open source.
- Requiere/emplea Python 3.6 y MongoDB 4.0+.

Funcionamiento básico

El acceso a la interfaz web de configuración (panel de control) se hace a través de un

navegador web apuntando a la dirección ip del servidor y al puerto en el que se ubico el servicio.

En el panel de control se ubica la función principal, *crear una aplicación*. Esta última, permite crear una instancia de aplicación que encapsula la conexión del servicio de AirNotifier con otros dos componentes, la aplicación web y los servicios de mensajería móvil.

Dentro de la instancia de *aplicación*, se encuentra la opción para configurar la conexión anteriormente mencionada que posibilita el flujo final que muestra en la imagen 4. La configuración se divide en dos partes:

- La conexión con la aplicación web: consiste en generar un “access key”, un código de acceso que posteriormente usará la aplicación para comunicarse con AirNotifier. La aplicación a través de la API enviará las notificaciones utilizando el código de acceso.
- La conexión con los servicios de mensajería: consiste en cargar los parámetros de conexión que brindan los servicios. En el caso del servicio APNs es necesario cargar el certificado de notificaciones push de desarrollo y la clave privada. Y en el caso de FCM el ID de proyecto y la clave JSON.

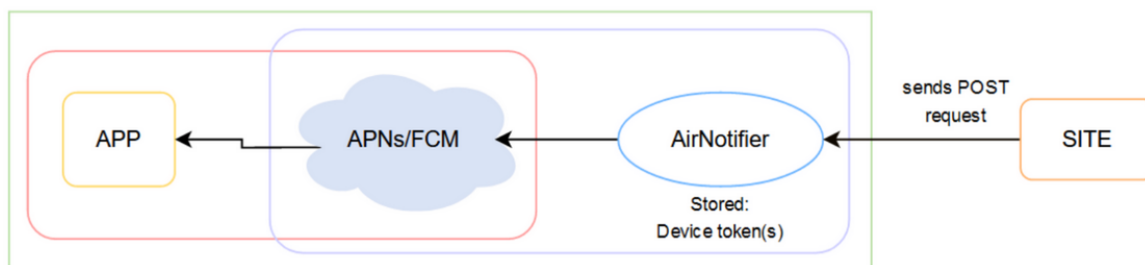


Imagen 4. Flujo de ejecución final de AirNotifier
Fuente. Nabilah Malek, 2020

La imagen 4 exhibe el flujo de ejecución normal desde la generación de la notificación push en la aplicación web hasta la recepción final de la aplicación móvil. Para que pueda ocurrir este flujo final son necesarios los siguientes pasos:

1. Conexión con los servicios de mensajería.
2. Conexión con la aplicación web (“SITE” en la imagen 4).
3. Conexión de la aplicación móvil con el servicio de mensajería y la aplicación web.

Para ello es necesario que la aplicación envíe al SITE el token de registro que le otorga FCM. Esto último ocurre cuando la aplicación móvil se inicia por primera vez, a través de servicios web, la aplicación obtiene un token y lo envía al SITE para que lo guarde. Finalmente, cuando la aplicación web desea enviar una notificación a una instancia de aplicación móvil específica utiliza el token correspondiente.

2.4.3 Firebase Cloud Messaging (FCM)

En el marco de este desarrollo, AirNotifier utiliza FCM como protocolo de comunicación para el envío de las notificaciones push.

Este es un servicio multiplataforma que ofrece Google para el envío de mensajes y notificaciones de forma segura entre dispositivos a través de Internet. Compatible con Android, iOS y aplicaciones web.

A la hora de integrar este servicio con otro sistema es necesario conocer sobre su arquitectura para optimizar el rendimiento y depurar con mayor facilidad. La estructura y los componentes de la arquitectura se pueden visualizar en la imagen 5.

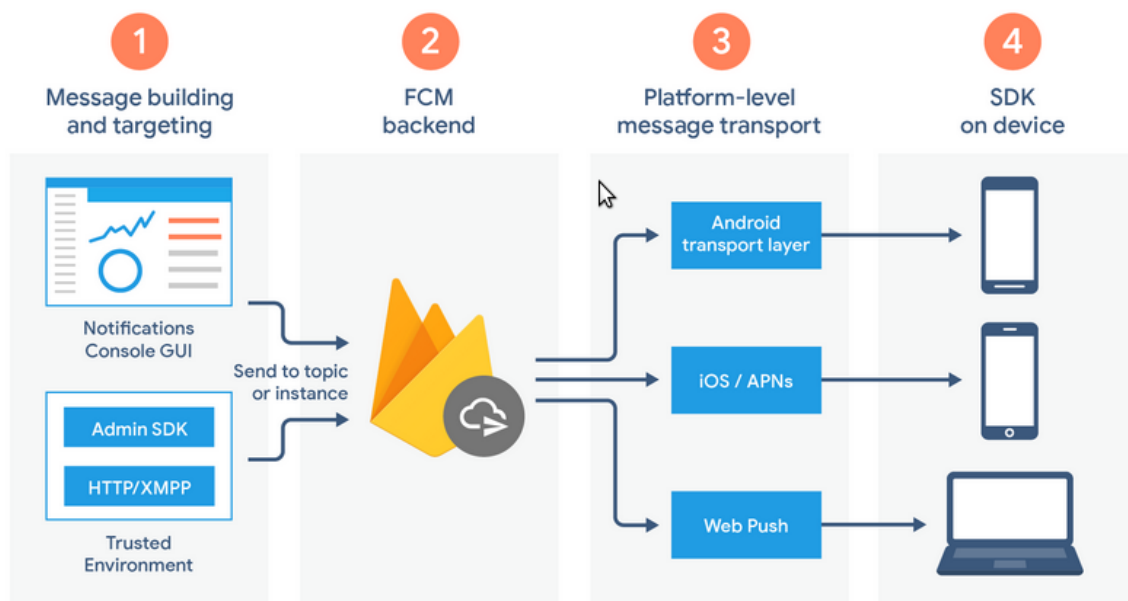


Imagen 5. Arquitectura de Firebase Cloud Messaging.
Fuente. Documentación oficial de Firebase Cloud Messaging, 2022.

En la imagen 5 se presentan cuatro componentes esenciales:

- Entorno de confianza: El entorno que genera y transmite la notificación. Ya sea un servidor de aplicaciones (que admita HTTP o XMPP) o "Cloud Functions", la cual es una herramienta propia de Firebase que permite generar notificaciones o mensajes).
- Backend FCM: Es la infraestructura que acepta las solicitudes de mensaje, genera metadatos como el ID del mensaje y realiza, opcionalmente, la distribución de los mensajes a través de temas. Gestiona el envío y la entrega de los avisos a los dispositivos clientes.
- Capa de transporte a nivel de plataforma: Enruta el mensaje al dispositivo cliente, controla la entrega y aplica la configuración específica (ATL/Android, APNs/iOS o Protocolo Web Push/Aplicación Web) de acuerdo al dispositivo destino.
- El SDK de FCM: Es la biblioteca de software que se integra en la aplicación móvil y permite la interacción con FCM.

Además de la arquitectura, otro aspecto a considerar sobre FCM son los tipos de mensajes que se pueden transmitir. Existen los mensajes de notificación, cuya sintaxis se muestra en la imagen 6 y los mensajes de datos, cuya sintaxis se muestra en la imagen 7.

```
{
  "message": {
    "token": "bk3RNwTe3H0:CI2k_HHwgIpoDKCIZvvDMExUdFQ3P1...",
    "notification": {
      "title": "Portugal vs. Denmark",
      "body": "great match!"
    }
  }
}
```

Imagen 6. Sintaxis de mensaje tipo notificación en FCM.

Fuente. Documentación oficial de Firebase Cloud Messaging, 2022.

```
{
  "message": {
    "token": "bk3RNwTe3H0:CI2k_HHwgIpoDKCIZvvDMEUdFQ3P1...",
    "data": {
      "Nick": "Mario",
      "body": "great match!",
      "Room": "PortugalVSDenmark"
    }
  }
}
```

Imagen 7. Sintaxis de mensaje de datos en FCM.

Fuente. Documentación oficial de Firebase Cloud Messaging, 2022.

Los mensajes de notificación son manejados automáticamente por el SDK. Estos, contienen un conjunto de claves visibles para el usuario y siguen un formato JSON. El cuerpo del mensaje se encuentra encapsulado en la palabra clave “notification”, como se muestra en la imagen 6.

Los mensajes de datos son por completo responsabilidad de la aplicación cliente. Contienen pares clave-valor personalizados definidos por el usuario, presenta formato JSON y el cuerpo del mensaje se encuentra encapsulado en la palabra clave “data”, como se muestra en la imagen 7.

En relación a los tipos de mensajes, FCM permite accionar de dos maneras diferentes a partir de la recepción del mensaje:

- Mensajes no contraíbles (Non-collapsible messages): Son los mensajes que siempre son enviados al dispositivo destino. Mensajes que no se actualizan ni se reemplazan.
- Mensajes contraíbles (Collapsible messages): Son los mensajes que pueden ser reemplazados si no llega a destino antes que aparezca un mensaje nuevo.

Por último, se destaca que a través del SDK de FCM se proporciona, a la instancia de aplicación que se ejecuta en el dispositivo, un token de registro que lo identifica unívocamente en el sistema. Este token se incluye en todas las peticiones a la API de FCM, y por esto es importante guardar los tokens en el servidor de aplicación junto a una marca de tiempo la cual evita que se sigan almacenando dispositivos inactivos.

2.4.4 Moodle App

Al abordar la personalización y el desarrollo de una aplicación basada en el código fuente de la aplicación móvil oficial de Moodle, resulta importante comprender sus características de diseño y su contexto.

Nicols (2022), en la documentación oficial, explica que Moodle app es la aplicación oficial de Moodle para celulares con sistemas operativos Android o iOS. La misma se ofrece en diferentes planes y cada uno ofrece distintos niveles de funcionalidad.

La app es open source y su código fuente se encuentra en un repositorio GIT²² de acceso público. Además, presenta una amplia comunidad de desarrolladores, quienes brindan ayuda y soporte continuo. Por esto, independientemente de los planes, cualquiera puede hacer un desarrollo propio a partir de su código (personalizaciones).

Uno de los aspectos a destacar en cuanto a su funcionamiento es el hecho de que la app no se acopla a un sitio Moodle en particular. Si bien no es posible el acceso simultáneo a diferentes sitios Moodle, las distintas sesiones quedan guardadas. Por ejemplo, esto permite recibir notificaciones push de los distintos sitios en los que se inició sesión independientemente de la sesión activa.

Se destacan las siguientes características respecto al diseño y la arquitectura de la aplicación:

- El código fuente sigue una arquitectura hexagonal, con módulos centrales (core modules) que incluyen la funcionalidad principal y módulos complementarios (addon modules) que brindan funciones adicionales. Esta característica de diseño favorece a la extensibilidad y a la mantenibilidad de la aplicación.
- La app se encuentra desarrollada con Ionic Framework (versión 5) en el marco de Angular²³. Ambas tecnologías facilitan el desarrollo de aplicaciones multiplataforma.
- En la aplicación se implementan varios conceptos propios de Angular, como:
 - Los módulos se definen como “Angular Modules”.

²² <https://github.com/moodlehq/moodleapp>

²³ <https://angular.io/>

- Estos, se ejecutan utilizando el “Angular dependency injection framework.”
- Las páginas y la navegación se definen utilizando “Angular Routing” haciendo uso de la carga diferida (lazy loading).

2.4.5 Servicios web

Los servicios web son el medio de comunicación entre la aplicación móvil de Moodle y el sistema Moodle.

Coggiola (2017) explica cómo los Servicios Web (WS), “en términos generales, son la manera de exponer información y funcionalidad de un sistema a través de tecnologías Web, respetando sus estándares. El uso de estos es clave, ya que reduce la heterogeneidad existente entre sistemas facilitando su integración. Esto nos permite utilizar los estándares Web como medio para comunicar sistemas.” (p.20)

A su vez, Albertengo (2020), define a los WS como interfaces programáticas para la comunicación entre agentes de software que utilizan, en la mayoría de los casos, el protocolo de transferencia de hipertexto (HTTP). Los agentes de software que brindan el servicio se denominan “service provider” y los consumidores son los “service requestor” (en el medio de la comunicación entre estos dos agentes existe el componente “service registry” pero no es muy utilizado). Las diferentes implementaciones de WS utilizan diferentes protocolos y estructuras de mensaje (JSON, XML). Los dos estilos arquitectónicos más conocidos son REST y SOAP.

Moodle denomina a sus servicios web como *servicios externos*. Hace uso de estos exponiendo información y funcionalidad (propia del sistema web) a los diferentes sistemas externos, a través de la web.

Los servicios externos se construyen a partir del *framework para servicios externos* y de la *API Externa*, que son tecnologías propias de Moodle. Es decir, a la hora de crear un servicio externo hay aspectos a respetar y una estructura a seguir, ya sea por el lado del framework o de la API.

En la mayoría de los casos la comunicación consiste en que la instancia de la aplicación cliente apunte a los endpoints definidos en los servicios externos.

Moodle soporta diferentes protocolos de comunicación en servicios web como REST/XML, REST/JSON, SOAP y XMLRPC.

Comprender el papel que toman los servicios web facilita el aprovechar las funcionalidades existentes en la aplicación. Incluso, modificar o desarrollar nuevas.

2.4.6 Ionic Framework

Al encarar un desarrollo sobre una aplicación existente es de vital importancia conocer sobre el lenguaje y/o herramienta con el que se encuentra programado, es clave para preparar el entorno de desarrollo. La versión 3.9.6 de la aplicación oficial de Moodle se encuentra programada con Ionic en su versión 5.

Este último, es un framework de código abierto enfocado en la Interfaz de Usuario (UI) que ofrece un conjunto de herramientas y servicios para desarrollar aplicaciones web progresivas, móviles (híbridas/multiplataforma) y de escritorio; con alta calidad y rendimiento.

Es un proyecto open source publicado bajo la licencia MIT (utilizada por jQuery, Ruby, entre otros), es decir, se puede utilizar en proyectos personales o comerciales de forma gratuita.

Lo destacable de Ionic es el hecho de poder construir aplicaciones móviles multiplataforma a partir de un único código base. Este código base se sustenta en tecnologías web como HTML, CSS y Javascript; actualmente lo más común es integrar a este código otros frameworks como Angular, React o Vue.

Otro aspecto a destacar es la posibilidad de emular los estándares que sigue un UI de app nativa y combinarlos con SDKs nativos. Esto es posible usando los plugins de Capacitor o Cordova, ambos permiten el acceso a las características nativas de un Sistema Operativo en particular.

Anteriormente, Ionic estaba estrechamente relacionado con Angular, a la hora de desarrollar se debía trabajar con este último. Sin embargo, a partir de la versión 4.0 hubo un rediseño en este sentido, para funcionar como una librería independiente de componentes web. Igualmente, se dejó la posibilidad de integrar con otros frameworks.

Por último, otro aspecto fuerte es la CLI de Ionic (interfaz de línea de comandos). Esta es una herramienta que permite un desarrollo rápido y sencillo a través de comandos preestablecidos. Además, viene con un servidor web de desarrollo incorporado y herramientas de compilación y depuración.

CAPÍTULO 3: Desarrollo

En el presente capítulo, se desarrollan las tres funcionalidades implementadas en la nueva aplicación personalizada basada en Moodle App. Se exponen los desafíos encontrados y las soluciones implementadas.

En la primera sección se detalla la implementación del módulo de notificaciones push, donde se presenta la estructura del desarrollo y se profundiza sobre los pasos requeridos para su integración.

En segundo lugar, para ampliar y mejorar el módulo de notificaciones, se desarrolló la funcionalidad que permite eliminarlas. Una vez que se pone en marcha el envío de notificaciones push, estas se muestran como un listado en la página *Notificaciones* (donde se listan todas las notificaciones de las aulas virtuales en las que el usuario está matriculado) dentro de la aplicación móvil. Con la nueva función eliminar, se plantea agregar un botón rojo junto a cada uno de las notificaciones, con la posibilidad de quitarlas del listado (esto se presenta en el apartado Frontend). Además, esto implica eliminar la notificación correspondiente de la base de datos de Moodle, para lo cual fue necesario crear un servicio web en Moodle que exponga esta acción (esto se presenta en el apartado Backend).

En la tercera sección se presenta otra mejora al módulo, la cual se basa en la

posibilidad de resaltar notificaciones. Este desarrollo se asemeja a la opción destacar correos en Gmail²⁴, pero en este caso, al destacar un aula virtual (dentro de la aplicación móvil) desde la página *Área personal>Vista general*, se resaltan las notificaciones provenientes de ese curso particular. El objetivo es permitir al usuario que pueda priorizar las notificaciones de un aula virtual específica, facilitando la identificación rápida y evitar que pasen desapercibidas.

Se decidió que las notificaciones destacadas se visualizan al principio del listado de la página *notificaciones* con una estrella dorada (esto se presenta en el apartado Frontend). Y se trabajó sobre el servicio web existente encargado de traer todas las notificaciones para un usuario. Se modificó dicho servicio para que considere las aulas virtuales destacadas y, de ese modo, resaltar las notificaciones correspondientes (esto se presenta en el apartado Backend).

Finalmente, se exponen detalles a considerar sobre el proceso de desarrollo y se presenta la aplicación móvil, con las nuevas funcionalidades ya integradas, publicada en la tienda de Google Play²⁵.

3.1 Implementación del módulo de notificaciones push

En la imagen 8 se ilustra la estructura push aplicada, con las tecnologías específicas que se emplearon. Esta imagen brinda una visión completa del proceso de implementación de la nueva funcionalidad.

²⁴ <https://gmail.com/>

²⁵ <https://play.google.com/>

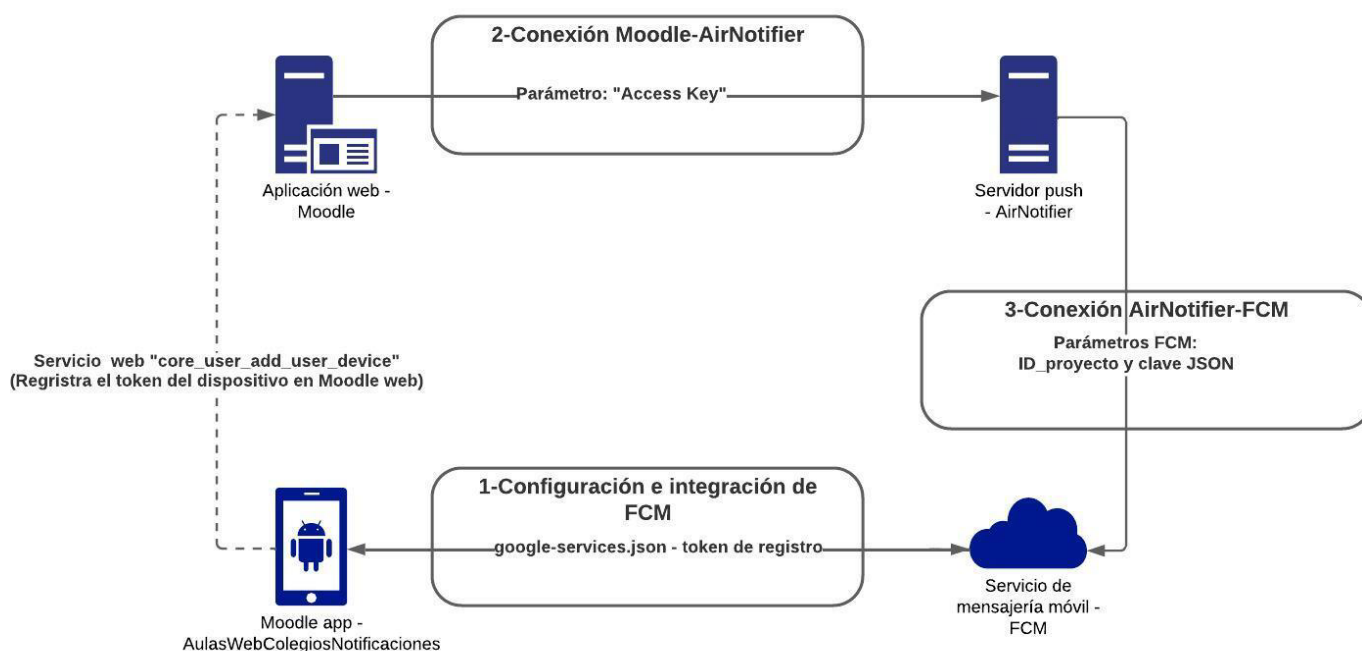


Imagen 8. Estructura push aplicada en el desarrollo

Como se puede visualizar, la notificación push se origina en la aplicación web Moodle (AulasWebColegios) para ser enviada al servidor push y que este las gestione. En este caso, AirNotifier es quien recibe y gestiona todas las notificaciones transmitidas por el sistema Moodle. Finalmente, AirNotifier utiliza FCM para hacer llegar los avisos a los dispositivos móviles.

Para que este flujo de ejecución se lleve a cabo, se completaron una serie de pasos (1, 2 y 3 en imagen 8) que implican la integración, puesta en marcha y configuración de las tecnologías mencionadas anteriormente. Estos pasos se dividen en tres etapas: Configuración e integración del servicio FCM, Puesta en marcha de AirNotifier y conexión con Moodle y Conexión de AirNotifier con FCM.

Configuración e integración del servicio FCM

Para implementar el módulo de notificaciones a partir del código fuente de la aplicación Moodle oficial, primero fue necesario descargar el código desde el repositorio oficial en GitHub²⁶.

²⁶ <https://github.com/>

El siguiente paso fue preparar el entorno de desarrollo, que consistió en instalar Node.js²⁷ y las dependencias nativas correspondientes al sistema operativo que se esté utilizando. Además, hay que tener en cuenta que la aplicación se encuentra desarrollada con Ionic 5 y se debe instalar como parte de las dependencias.

Para verificar la correcta configuración del entorno de desarrollo se inició la aplicación desde el navegador web (esto es posible gracias a Ionic). Posicionado dentro del proyecto, a través de una consola de comando, se corrió el comando: *ionic serve*.

Luego, para utilizar los servicios de Firebase²⁸, en particular el servicio de FCM, fue imprescindible crear una cuenta y un proyecto desde la consola de desarrollo en su página web. En el panel de control, dentro del proyecto, se genera una instancia de aplicación Android, que servirá de base para vincularse con nuestra nueva versión de la aplicación Moodle. Esta nueva versión fue denominada “AulasWebColegiosNotificaciones”.

De esta instancia de aplicación, se obtiene el archivo *google-services.json* el cual nos sirve para terminar de realizar la integración con la app. Para ello solo basta con cargar el archivo json en la raíz del proyecto. Las dependencias y llamadas a los servicios de Firebase ya se encuentran en el código oficial por defecto.

Para probar la integración se enviaron notificaciones push a la app desde la consola de *Messaging* en Firebase. Esto es posible porque el código de recepción de notificaciones ya se encuentra implementado por defecto en Moodle app a través del plugin *cordova-plugin-push*²⁹. La clase que implementa los métodos del plugin se encuentra en el directorio: *raízMoodleApp\src\core\features\pushnotifications\services\pushnotifications.ts*

Puesta en marcha de AirNotifier y conexión con Moodle

La puesta en marcha consistió en instalar el servicio de AirNotifier en un servidor Linux Debian 10. Para ello, previamente, se instalaron las dependencias del servicio:

²⁷ <https://nodejs.org/>

²⁸ <https://firebase.google.com/>

²⁹ <https://github.com/havesource/cordova-plugin-push>

- MongoDB 4.0
- Python 3.6
- Dependencias específicas para trabajar con Python (El detalle se puede encontrar en el repositorio oficial de AirNotifier en GitHub)

Una vez que se inicia el servicio de AirNotifier se puede acceder al panel de administración desde el navegador web apuntando al puerto 8801 (puerto por defecto). En el panel de administración se encuentra la opción *Create new app*, la cual nos permite generar la instancia de aplicación que encapsula la conexión con Moodle y FCM. Esto nos devuelve el “Access Key”, parámetro necesario para vincular con Moodle.

Para realizar la conexión de Moodle con AirNotifier fue preciso:

- Habilitar los servicios web para los dispositivos móviles dentro del entorno moodle AulasWebColegios, desde las opciones de administración del sitio, como se muestra en la imagen 9 (*Administración del sitio>App para dispositivos móviles>Ajustes Móviles*). Esta opción habilita el módulo general de servicios web y el protocolo REST que utiliza Moodle por defecto.

Página Principal > Administración del sitio > App para dispositivos móviles

Ajustes móviles

Habilitar servicio web para dispositivos móviles
enablemobilewebservice

Valor por defecto: Sí

Habilitar el servicio para móviles

Imagen 9. Servicios web para dispositivos móviles en AulasWebColegios.

- Agregar el “Access Key” obtenido en la configuración de AirNotifier dentro del entorno y completar el resto de parámetros de integración. Esto, desde la sección *Administración del sitio>Mensajería>Móvil*, como se muestra en la imagen 10.

<p>URL de Airnotifier airnotifierurl</p>	<input type="text" value="http://163.10.55.37"/>
	La URL del servidor al que conectarse
<p>Puerto de Airnotifier airnotifierport</p>	<input type="text" value="8801"/> Valor por defecto: 443
	El puerto a usar al conectarse al servi
<p>Nombre de aplicación móvil airnotifiermobileappname</p>	<input type="text" value="com.moodle.moodlemobile"/>
	El identificador único de la aplicaciór
<p>Nombre de aplicación Airnotifier airnotifierappname</p>	<input type="text" value="aulaswebcolegios"/>
	El identificador de nombre de la aplic
<p>Clave de acceso de Airnotifier airnotifieraccesskey</p>	<input type="text" value="9aa28ccd1c4792af07ea89790b5d"/>

Imagen 10. Parámetros de integración de AirNotifier en AulasWebColegios.

Conexión de AirNotifier con FCM

Como paso final y siguiendo con la imagen 8, se vinculó el servicio de AirNotifier con Firebase Cloud Messaging. Para ello se ingresó, nuevamente, al panel de administración dentro de la instancia de aplicación creada en AirNotifier y en las opciones de configuración de FCM, se ingresó el ID del proyecto y la clave JSON como se muestra en la imagen 11. Estos datos se consiguen en la consola del proyecto en Firebase.

Key	Description	Permission
9aa28ccd1c4792af07ea89790b5d502b	AulasWebColegios Key	<ul style="list-style-type: none">✓ Create token✓ Delete token✓ Send notification✓ Send broadcast✓ Create access key

Imagen 10. Configuración de FCM en el panel de AirNotifier.

3.2 Implementación de la funcionalidad “Borrar notificación”

Frontend

Para que cada una de las notificaciones listadas tengan la opción de borrar, se diseñó un botón rojo que aparece cuando se desliza hacia la izquierda sobre la notificación, como se muestra en la imagen 11.

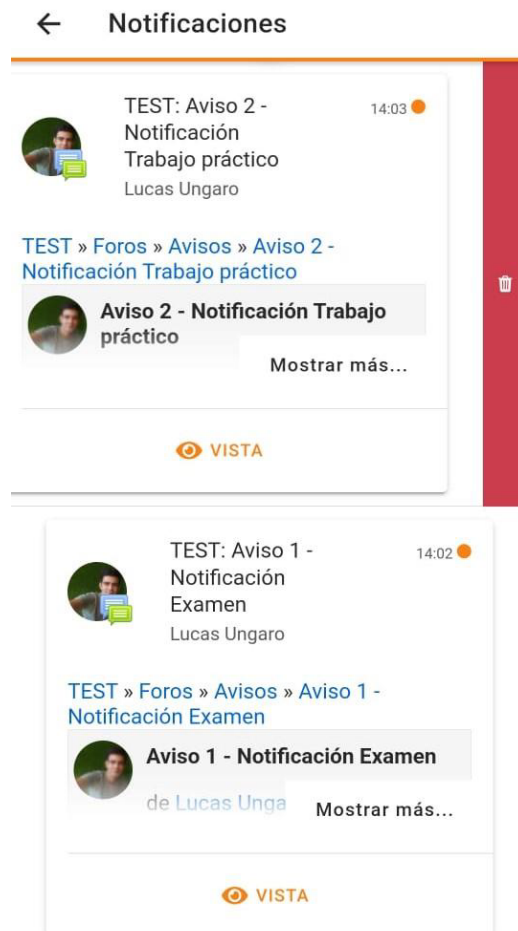


Imagen 11. Nuevo botón borrar en la sección de Notificaciones.

Para implementar la funcionalidad a nivel frontend fue necesario modificar tres archivos del código fuente:

- *raízMoodleApp\src\addons\notifications\pages\list\list.html*
- *raízMoodleApp\src\addons\notifications\pages\list\list.ts*
- *raízMoodleApp\src\addons\notifications\services\notifications.ts*

En list.html se agregó el botón de borrar rojo con el icono del cesto de basura. Este archivo html representa a la página Notificaciones dentro de la app, donde se listan las notificaciones que se reciben desde el entorno AulasWebColegios. En términos de código fuente, las notificaciones se listan con la directiva estructural, propia de Angular, *ngFor como se puede ver en la imagen 12, además, se destaca la utilización de la etiqueta <ion-item-sliding>, propia de Ionic, para generar la animación de deslizar sobre un elemento de la interfaz.

```

<ng-container *ngFor="let notification of notifications">
  <ion-item-sliding class="ion-text-wrap">
    <ion-item>
      <ion-card [ngClass]="{ 'fav-card': notification.belongsToFavouriteCourse
        <ion-item button class="ion-text-wrap" detail="false" *ngIf="notifica
        </ion-item>
        <ion-item class="ion-text-wrap" lines="none" [attr.aria-label]="...
        </ion-item>
        <ion-item class="ion-text-wrap">...
        </ion-item>
        <addon-notifications-actions [contextUrl]="notification.contexturl" [
          [data]="notification.customdata">
        </addon-notifications-actions>
      </ion-card>
    </ion-item>
    <ion-item-options side="end">
      <ion-item-option color="danger" (click)="deleteNotification(notification)
        <ion-icon name="trash"></ion-icon>
      </ion-item-option>
    </ion-item-options>
  </ion-item-sliding>
</ng-container>

```

Imagen 12. Código HTML de la función borrar notificación.

En la imagen 12, también, se puede ver como el evento click se maneja con la función *deleteNotification(notification)*. La implementación de dicha función se halla en *list.ts*, donde se encuentra el resto de la lógica del html. Dentro de esta última función se hace una llamada a una instancia (AddonNotification) de la clase *AddonNotificationsProvider* (imagen 13), quien a través de la función *deleteNotification(notification.id)* hace la invocación al servicio web Moodle (imagen 14) *core_message_delete_notification*. La clase *AddonNotificationsProvider* se encuentra en el archivo *notifications.ts*.

```

async deleteNotification(notification: AddonNotificationsNotificationToRender): Prom
  try {
    await AddonNotifications.deleteNotification(notification.id);
    CoreDomUtils.showToast('addon.notifications.successfullydeletedmessage', true,
      this.refreshNotifications());
  } catch (error) {
    CoreDomUtils.showErrorModalDefault(error, 'addon.notifications.errordeleting
  }
}

```

Imagen 13. Función *deleteNotification* en *list.ts*.


```

async deleteNotification(
  notificationId: number,
  siteId?: string,
): Promise<CoreMessageDeleteNotificationWSResponse> {
  const site = await CoreSites.getSite(siteId);

  const params: CoreMessageDeleteNotificationWSParams = {
    notificationId: notificationId,
    useridto: CoreSites.getCurrentSiteUserId(),
  };

  return site.write<CoreMessageDeleteNotificationWSResponse>('core_message_delete_notification'
}

```

Imagen 14. Función `deleteNotification` en `notifications.ts`

Backend

A nivel backend se desarrolló un nuevo servicio web Moodle, el servicio `core_message_delete_notification`. Para ello se modificaron tres archivos del código fuente Moodle, en este caso AulasWebColegios:

- `raízMoodleWeb/lib/db/services.php`
- `raízMoodleWeb/message/externallib.php`
- `raízMoodleWeb/message/classes/api.php`

En el archivo `services.php` se declaran las funciones y servicios externos a nivel de Core, es decir, características y funciones que forman parte del núcleo de Moodle. Con lo cual en este archivo se declaró el servicio `core_message_delete_notification` como se muestra en la imagen 15, agregando como parámetros principales:

- La ruta al archivo que contiene la clase/función que se encarga de resolver el servicio.
- Nombre de la clase
- Nombre de la función / método

```
'core_message_delete_notification' => array(  
    'classname' => 'core_message_external',  
    'methodname' => 'delete_notification',  
    'classpath' => 'message/externallib.php',  
    'description' => 'A certain notification is deleted',  
    'type' => 'write',  
    'capabilities' => 'moodle/site:deleteownmessage',  
    'services' => array(MOODLE_OFFICIAL_MOBILE_SERVICE),  
    'ajax' => true,  
)
```

Imagen 15. Declaración del nuevo servicio en services.php.

En `externallib.php` se implementaron tres funciones, las cuales forman parte del diseño estándar de un servicio web en Moodle. Estas funciones son:

- *function delete_notification_parameters()*
- *function delete_notification(\$notificationid, \$useridto)* (Función principal)
- *function delete_notification_returns()*

Todo servicio debe especificar los parámetros que recibirá y retornará la función principal (*delete_notification*) del servicio. En este caso, en la función *parameters* se declara el ID del usuario y el ID de la notificación. Y en *returns* se declaran los parámetros de retorno de la función principal. En este caso, *status* informa si el borrado fue exitoso y *warnings* se utiliza para que, en el caso de que exista un error, se especifique el mismo.

En la función principal del servicio web *delete_notification(\$notificationid, \$useridto)* se hace la llamada a la función *delete_notification(\$useridto, \$notificationid)*, como se muestra en la imagen 16, que se encuentra en el archivo `api.php`. Esta última es quien ejecuta las consultas SQL correspondientes a la baja de la notificación utilizando el ID que se paso por parámetro en la función.

```

public static function delete_notification($notificationid, $useridto){
    global $CFG;
    $warnings = array();

    $params = self::validate_parameters(self::delete_notification_parameters(), array(
        'notificationid' => $notificationid,
        'useridto' => $useridto,
    ));
    // Validate context.
    $context = context_system::instance();
    self::validate_context($context);
    $user = core_user::get_user($params['useridto'], '*', MUST_EXIST);
    core_user::require_active_user($user);

    $status = \core_message\api::delete_notification($user->id, $params['notificationid']);
    return array(

```

Imagen 16. Función deleteNotification en el archivo externallib.php.

Tanto el archivo externallib.php como api.php se encuentran dentro del módulo *message*. El cual es el encargado de los mensajes en general dentro del sistema web Moodle.

Luego de agregar el servicio a nivel código se lo agregó a nivel sistema, es decir, que se pueda visualizar en la sección de Administración del sitio dentro de AulasWebColegios.

Para ello se incrementó en uno el parámetro versión dentro del código Moodle para forzar la actualización del sistema y que el software se termine de cargar. Hecha la actualización se verificó en el apartado *Administración del sitio>Extensiones>Servicios web>Documentación de API*.

3.3 Implementación de la funcionalidad “Destacar Notificaciones”

Frontend

Como se mencionó anteriormente, se planeó que las notificaciones destacadas se visualicen al comienzo del listado con el icono de una estrella y el borde superior dorado, como se muestra en la imagen 17 con la notificación del curso PMI.

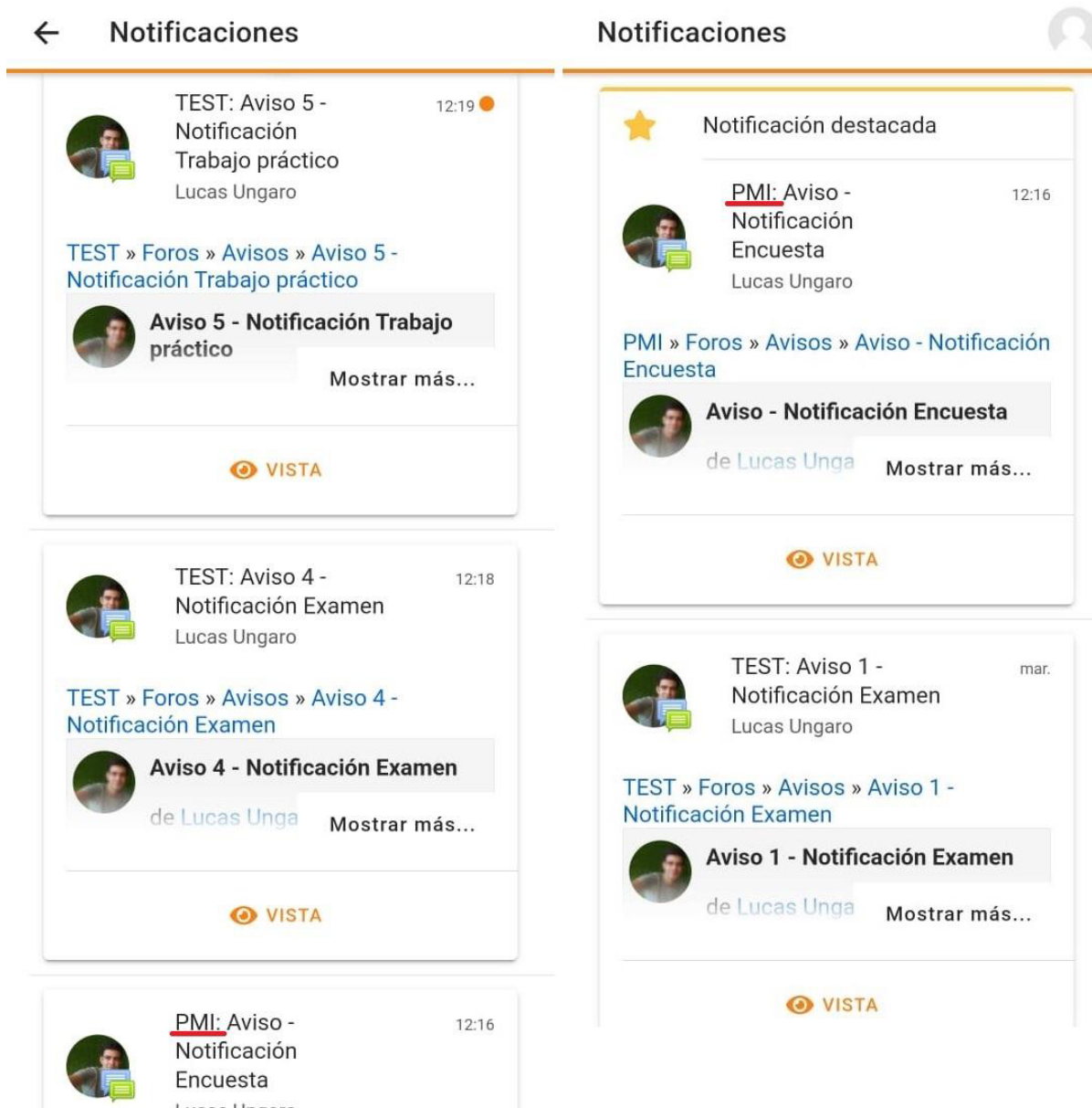


Imagen 17. Notificaciones destacadas en la sección Notificaciones.

Para implementar la funcionalidad a nivel frontend fue necesario modificar dos archivos del código fuente:

- `raízMoodleApp\src\addons\notifications\pages\list\list.html`
- `raízMoodleApp\src\addons\notifications\pages\list\list.ts`

En list.html se agregó la directiva condicional, propia de Angular, *NgIf para preguntar

por la nueva propiedad asignada a una notificación *belongsToFavouriteCourse*, variable de tipo Boolean. A las notificaciones que llegan a la app con esta propiedad se les agrega (imagen 18):

- El texto: Notificación destacada.
- El icono: de una estrella dorada.
- Nuevos estilos: para personalizar la tarjeta de ionic.

```
<ng-container *ngFor="let notification of notifications">
  <ion-item-sliding class="ion-text-wrap">
    <ion-item>
      <ion-card [ngClass]="{ 'fav-card': notification.belongsToFavouriteCourse }">
        <ion-item button class="ion-text-wrap" detail="false" *ngIf="notification.belongsToFavouriteCourse">
          <ion-icon name="fas-star" class="star" slot="start" aria-hidden="true"></ion-icon>
          <ion-label>
            <p class="item-heading">Notificación destacada</p>
          </ion-label>
        </ion-item>
      </ion-card>
    </ion-item>
  </ion-item-sliding>
</ng-container>
```

Imagen 18. Código HTML de la función *destacar notificaciones*.

En “list.ts” se agregó la función `ionViewWillEnter()`, propia de Ionic, para volver a ejecutar el método que trae a las notificaciones antes de cargar la página, cuando se ingresa a la sección “Notificaciones”. Esto tiene que ver con los ciclos de vida de las páginas y componentes Ionic. Es decir, antes de que se ejecute la animación de carga, dentro de la página, se vuelven a buscar las notificaciones del usuario.

Backend

Para implementar la funcionalidad a nivel backend se modificaron dos archivos del código fuente del sistema Moodle de AulasWebColegios:

- *raízMoodleWeb/message/externallib.php*
- *raízMoodleWeb/message/lib.php*

En este caso no se creó un nuevo servicio web Moodle, sino que se agregó código a la función principal del servicio ya implementado, *core_message_get_messages*. Este

servicio trae a la aplicación móvil las notificaciones pendientes provenientes del sistema Moodle, para un usuario determinado (el usuario que inicio sesión en la app).

Para este caso, al no crear un servicio nuevo no es necesario actualizar el sistema Moodle.

En `externallib.php` se encuentra definido el servicio `core_message_get_messages`, es decir, en este archivo se encuentran las 3 funciones propias de la estructura estándar de un servicio web Moodle:

- `function get_messages_parameters()`
- `function get_messages($useridto, $useridfrom = 0, $type = 'both', $read = true, $newestfirst = true, $limitfrom = 0, $limitnum = 0)` (FUNCIÓN

PRINCIPAL)

- `function get_messages_returns()`

Se trabajó en la función principal y en `returns`. En `get_messages` se insertó código para verificar si el usuario tiene cursos destacados y que en caso afirmativo se modifique el orden que luego utilizará el método `message_get_messages` (Imagen 19), implementado en `lib.php`, que busca las notificaciones en la base de datos respetando el orden que se pasa por parámetro. El orden se altera ya que las notificaciones destacadas deben aparecer al principio del listado de notificaciones en la app.

```
if($userHasFavouriteCourses){
    $sort = "fav DESC";
}else{
    $orderdirection = $newestfirst ? 'DESC' : 'ASC';
    $sort = "mr.timecreated $orderdirection";
}

if ($messages = message_get_messages($useridto, $useridfrom, $notifications, $read, $sort, $limitnum, $limitfrom, $canviewfullname = has_capability('moodle/site:viewfullnames', $context));

// In some cases, we don't need to get the to/from user objects from the sql query.
$userfromfullname = '';
$userstofullname = '';
```

Imagen 19. Función principal `get_messages` en `externallib.php`.

Luego, en base al retorno del método `message_get_messages` se agrega el

parámetro *belongsToFavouriteCourse* a cada notificación según corresponda.

Por esto último, es que se declara este nuevo parámetro en la función *get_messages_returns*, ya que se está agregando un atributo nuevo a las notificaciones que retorna el servicio web.

Detalles a considerar

- El payload que se emite desde AirNotifier hacia la aplicación móvil y el manejo del evento click sobre una notificación emergente: se buscó que al click sobre una notificación el usuario sea dirigido a la sección de *Notificaciones*, dentro de la app.

Como se mencionó anteriormente, el manejo de las notificaciones que llegan a la aplicación se realiza a través del plugin cordova-plugin-push. En la documentación de dicho plugin, se indica que para manejar el evento click sobre una notificación emergente no solo hay que agregar el código correspondiente al controlador del evento sino que también hay que activarlo por medio del payload que llega a la aplicación con la notificación. El payload debe contener en el parámetro *data* el atributo *content-available* con el valor *1*.

Para este desarrollo fue necesario modificar el payload que se emite desde AirNotifier, ya que por defecto el mismo no contiene el parámetro *content-available*.

Por lo tanto, se agregó código en la función *build_request* de la clase *FCMClient* (imagen 20), que se encuentra en el archivo (dentro del código fuente de AirNotifier):

- *raízAirNotifier/pushservices/fcm.py*

```

def build_request(self, token, alert, **kwargs):
    if alert is not None and not isinstance(alert, dict):

        fcm_param = kwargs.get("fcm", {})
        android = fcm_param.get("android", {})
        apns = fcm_param.get("apns", {})
        webpush = fcm_param.get("webpush", {})
        data = fcm_param.get("data", {})
        payload = {"message": {"token": token}}
        if alert: ...

        if data: ...

        if android: ...

        if webpush: ...

        if apns: ...

        payload["message"]["data"]["content-available"] = "1"

```

Imagen 20. Modificación del payload que se emite desde AirNotifier.

- Forzar la actualización de la aplicación web Moodle:

Se mencionó que para cargar el nuevo software, que borra una notificación en particular, a nivel sistema hay que incrementar la versión del Moodle. En específico, se incrementó en 1 la parte decimal del número version, como se muestra en la imagen 21, que se encuentra en el archivos version.php dentro de la raíz del código fuente Moodle. La parte decimal representa los cambios mínimos a nivel sistema.


```
defined('MOODLE_INTERNAL') || die();

// $version = 2020061502.07;
$version = 2020061502.08;

$release = '3.9.2+ (Build: 20201009)';
$branch = '39';
$maturity = MATURITY_STABLE;
```

Imagen 21. Incremento de la versión en Moodle.

- Es poco probable que existan problemas de memoria del lado del frontend, en el listado de notificaciones ya que se realiza un paginado a nivel backend. Y con respecto a la base de datos que administra Moodle, se sabe que si bien es de tipo relacional, existe un manejo de índices en las tablas que involucran a las notificaciones.

Publicación en Google Play

Con las nuevas funcionalidades ya integradas se procedió a publicar la app en la tienda de Google Play. Para ello fue necesario:

- Registrarse como desarrollador Google, que implica realizar un único pago y estar de acuerdo con los términos y condiciones.
- Preparar la aplicación, teniendo en cuenta las políticas de desarrollo de Google Play y los errores que surgen al intentar subir el APK (formato de archivo que contiene todos los elementos para instalar y ejecutar la app en un dispositivo con sistema operativo Android) utilizando la consola de Google Play.
- Crear un “lanzamiento”, una vez que el APK es aceptado en la consola se deben

configurar los detalles de lanzamiento, como idioma, categoría en la tienda, clasificación de contenido, regiones en las que estará disponible, entre otros.

- Enviar la app para revisión, el equipo de Google realiza un proceso de evaluación para verificar que se cumpla con todas las políticas de desarrollo. Este es el paso más complicado y que puede llevar más tiempo. En este caso particular, hubo que eliminar el permiso "REQUEST_INSTALL_PACKAGES", ya que no cumplía con las nuevas políticas y hubo que realizar modificaciones en el plugin cordova-plugin-zip por generar vulnerabilidades a nivel de seguridad.

Una vez completados los pasos, la aplicación móvil AulasWebColegiosNotificaciones pasó a estar disponible en la tienda de aplicaciones de Google como se muestra en la imagen 22.

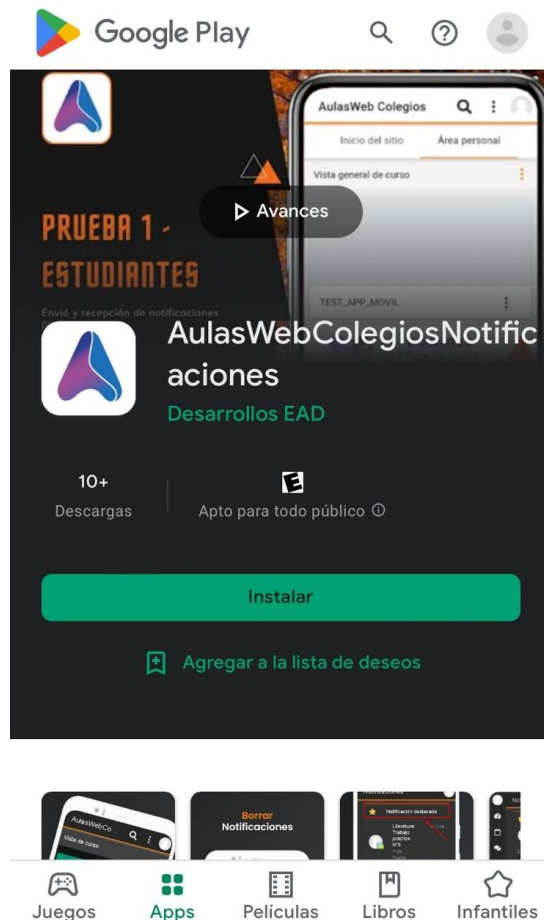


Imagen 22. Aplicación publicada en Google Play.

Como se puede visualizar en la imagen 22, también se publicaron imágenes descriptivas de las nuevas funcionalidades y videos tutoriales sobre cómo probarlas. Estos aspectos se abordarán en detalle en el siguiente capítulo.

CAPÍTULO 4: Pruebas con los usuarios

En este capítulo se aborda la presentación del proyecto a los usuarios finales y se expone la prueba piloto que se llevó a cabo en esta primera instancia.

4.1 Introducción

Una vez publicada la aplicación, se informó a los colegios de la UNLP para que los estudiantes y docentes la descarguen y comprueben las nuevas funcionalidades. El comunicado se realizó por correo electrónico informando:

- Disponibilidad de la app en tienda de Google Play.
- Disponibilidad exclusiva para sistemas operativos Android.
- Se recordó las tres funcionalidades nuevas implementadas.
- Se mencionó la creación de videos tutoriales en Youtube (canal Desarrollos EAD³⁰) explicando el paso a paso para probar las funcionalidades.
- Se presentó un Formulario Google para recabar datos respecto a la validación de los usuarios.

Si bien se informó a todos los colegios, se decidió trabajar en forma directa con dos grupos del Bachillerato de Bellas Artes “Francisco Américo de Santo” que se hallaban disponibles para realizar una prueba piloto, la cual se realizó con el objetivo de identificar posibles problemas y obtener el feedback de los usuarios. Sin embargo, se destaca que esta prueba piloto es el inicio de un proceso más amplio de pruebas, el desarrollo seguirá abierto a futuras evaluaciones y correcciones.

³⁰ <https://www.youtube.com/@desarrollosead1375>

El primer grupo estuvo conformado por un docente y cuatro estudiantes, mientras que el segundo grupo estuvo compuesto por un docente y ocho estudiantes. A continuación se describen las dos etapas en las que se dividió la prueba piloto.

4.2 Pruebas

En esta etapa se verificaron las tres funcionalidades desarrolladas a partir de una serie de pruebas que tuvieron que realizar los usuarios con ayuda de los videos tutoriales de youtube, que se pueden visualizar en la imagen 23.

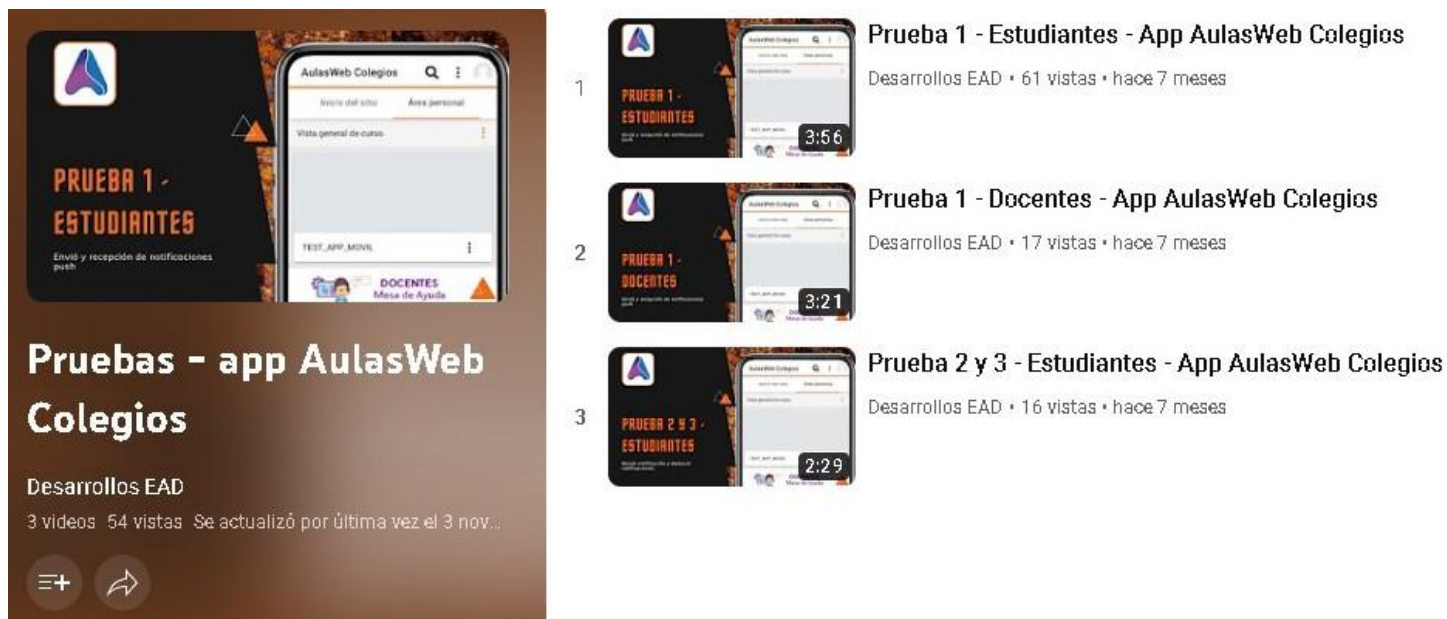


Imagen 23. Videos tutoriales para completar las pruebas.

Como se puede ver en la imagen 23 los estudiantes hicieron participaron en tres y los docentes en una. El objetivo de la primera prueba era verificar el envío y recepción de notificaciones push. En esta instancia participaron tanto los docentes como los estudiantes. El docente era el encargado de generar la notificación desde la plataforma AulasWebColegios. Para ello, se le pidió que accediera al aula virtual donde se encontraban todos los participantes matriculados y genere un evento de notificación, en este caso se solicitó que realicen una publicación en un foro de avisos.

Por su parte, los estudiantes debían descargar la aplicación móvil, iniciar sesión y esperar la recepción de las notificaciones push. Una vez que llegaban, debían hacer clic

sobre una de ellas y verificar si eran dirigidos a la página de notificaciones dentro de la aplicación. Para la prueba dos y tres se les pidió a los estudiantes que experimenten la acción de eliminar y destacar para terminar de comprobar todo el desarrollo.

4.3 Encuesta de evaluación

Finalizada la prueba piloto, se pidió a los usuarios que completen un formulario Google donde se preguntó directamente por el desempeño de cada una de las nuevas funcionalidades (**ver Anexo 3**).

El gráfico 6 refleja el rendimiento del envío y recepción, desde la llegada del aviso emergente al teléfono hasta la correcta redirección dentro de la aplicación:

1 - Funcionalidad "envío y recepción de notificaciones push":
14 respuestas



Gráfico 6. Prueba 1 - envío y recepción de notificaciones push.

Se destaca que el 28.6% de los usuarios (equivalente a 4 personas) pudieron realizar la prueba pero con un comportamiento incorrecto. En estos cuatro casos se logró la recepción de las notificaciones pero no pudieron visualizarlas en la página de Notificaciones. Esto último ocurre la primera vez que se envían avisos a la aplicación y la causa se relaciona a los tiempos de sincronización con el entorno Moodle. Para que aparezcan las notificaciones en el listado hicieron "pull down to refresh", es decir, deslizaron hacia abajo la pantalla para actualizar el contenido de la página. De esta manera pudieron completar las otras dos funcionalidades.

Solo en dos casos las notificaciones no llegaron al dispositivo móvil, ambos estudiantes tenían bloqueada la recepción de notificaciones push a nivel sistema operativo.

Los gráficos 7 y 8 reflejan el rendimiento de las funciones eliminar y destacar respectivamente:

2 - Funcionalidad "borrar notificación":

14 respuestas

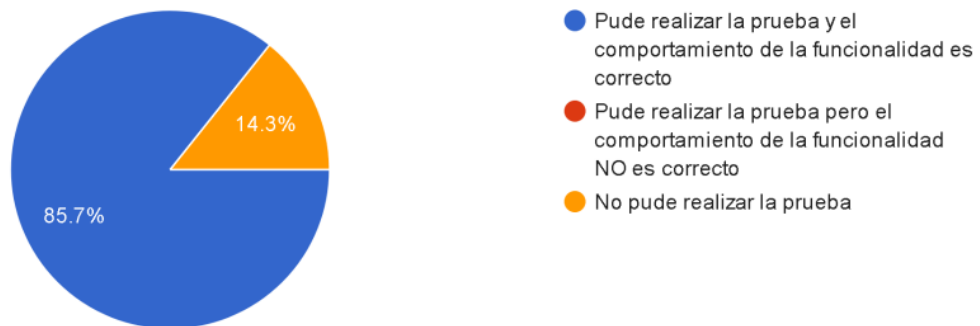


Gráfico 7. Prueba 2 - borrar notificación.

3 - Funcionalidad "Destacar Notificaciones":

14 respuestas

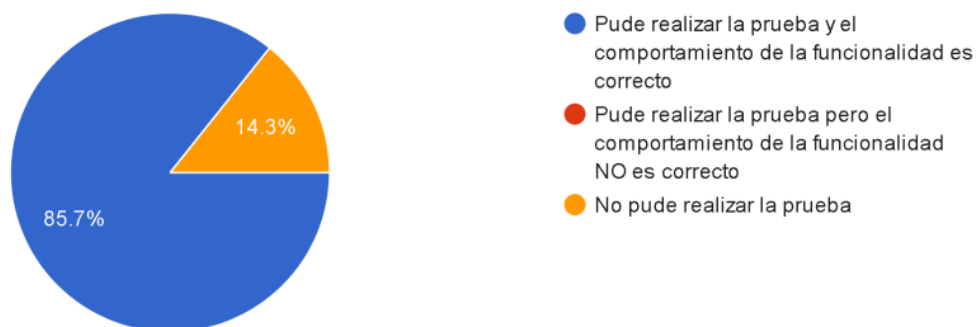


Gráfico 8. Prueba 3 - destacar notificaciones.

En ambos casos el 14.3% corresponde a los dos estudiantes que no recibieron las notificaciones push en su teléfono.

CAPÍTULO 5: Conclusiones y trabajos futuros

Como primera conclusión, se logró alcanzar los objetivos propuestos de incorporar notificaciones push a una aplicación móvil basada en Moodle y mejorar el módulo de notificaciones a partir del desarrollo de dos nuevas funcionalidades, agilizando la comunicación docente-estudiante en el entorno AulasWebColegios de la Universidad Nacional de La Plata. A continuación, se detalla cómo los logros alcanzados contribuyeron al objetivo:

- La implementación de las notificaciones push en la aplicación móvil personalizada permitió establecer un canal de comunicación directo entre el entorno virtual y los celulares, brindando un acceso flexible y conveniente para los estudiantes.
- El desarrollo de la funcionalidad de borrado aporta a la gestión de la información, buscando el orden en el listado de notificaciones. Ahora los usuarios pueden eliminar las notificaciones que consideren irrelevantes o que ya hayan sido leídas.
- El desarrollo de la funcionalidad para destacar notificaciones permitió que ahora los usuarios puedan priorizar y resaltar las novedades/avisos más relevantes dentro de las diferentes aulas virtuales.
- El análisis tecnológico permitió tomar decisiones fundamentadas y estratégicas, permitiendo enfocar al desarrollo propio y la mejora continua. El análisis específico de Moodle a nivel comunicacional fue clave para comprender el funcionamiento de las notificaciones dentro del sistema.
- El análisis de viabilidad permitió identificar los posibles aspectos a mejorar sobre la comunicación desde las aulas virtuales y el grado de receptividad a las notificaciones push.

A través del análisis de viabilidad, se observó la receptividad positiva de los estudiantes hacia la implementación de una aplicación móvil personalizada que permita el acceso al EVEA vía dispositivo móvil y que facilite la comunicación de las novedades del aula virtual a través de notificaciones push. Estos aspectos por parte de los estudiantes se relacionan con otros dos factores provenientes de los datos recabados.

En primer lugar, se observó que los estudiantes priorizan el uso del celular como medio para acceder al entorno virtual de enseñanza y aprendizaje. Esto sirvió de base a la propuesta de implementar una aplicación móvil oficial para AulasWebColegios que les permita la recepción de las novedades del aula virtual y les ofrezca un acceso rápido a la información relevante.

En segundo lugar, se observó que los estudiantes ya están haciendo un uso considerable de WhatsApp como herramienta para recibir notificaciones y estar al tanto de las novedades. Es decir, ya existe una predisposición hacia la comunicación a través de notificaciones push.

Si bien los estudiantes hacen uso de WhatsApp y otras aplicaciones de mensajería instantánea (“Otros” en Gráfico 4), se espera que, al contar con una aplicación móvil específica en el ámbito académico, se facilite la comunicación enfocada al aprendizaje, evitando las distracciones y la sobrecarga de información. Una herramienta de comunicación focalizada en el contexto educativo tiene la capacidad de transmitir información relevante como fechas de exámenes, tareas pendientes, y demás información puntual dentro del aula.

En cuanto a los docentes, se constató una preferencia por el uso de la notebook como dispositivo para el acceso al entorno y del correo electrónico como tecnología para transmitir los avisos e información importante. Y, tanto docentes como estudiantes comparten la iniciativa de unificar los canales de comunicación existentes en el EVEA. En

futuras investigaciones se espera hacer foco a los requerimientos de los docentes y buscar la unificación de los canales de comunicación.

Además de la recepción exitosa de las notificaciones a la aplicación AulasWebColegiosNotificaciones, se demostró que es factible trabajar con los servicios web que expone el sistema Moodle (AulasWebColegios). Las dos nuevas funcionalidades desarrolladas, que contribuyeron a la mejora del módulo de notificaciones interactuaron de manera efectiva con los servicios web. En particular para la funcionalidad de eliminar notificaciones se creó un servicio web desde cero en el backend del EVEA y fue posible siguiendo la documentación oficial de Moodle.

También, se logró una interacción exitosa con AirNotifier, siendo que inicialmente se presentó como un desafío. Aunque existían otras tecnologías para la gestión de notificaciones, se decidió priorizar la compatibilidad con Moodle ya que la empresa lo mencionaba como una opción recomendada. A diferencia de otras empresas que ofrecen el servicio de notificaciones push, AirNotifier no presentaba foros de ayuda, una comunidad de usuarios o un soporte oficial por parte de Moodle. Esto significó acceder al código fuente de la herramienta y comenzar con el reto de comprender su funcionamiento y estructura. Además, fue necesario ir realizando pruebas para verificar su comportamiento antes de su posterior puesta en producción. Como recomendación, se plantea investigar otros enfoques, como la incorporación de notificaciones a través del desarrollo de un plugin en Moodle que se encargue de gestionarlas.

Como reflexión final se plantean tres cuestiones sobre el proceso de realización del proyecto:

Primero, el desarrollo móvil en general representó un desafío en este proyecto de tesina, dado que fue mi primera incursión fuera del ámbito del desarrollo web. Si bien gracias al enfoque híbrido y al uso de tecnologías web como Ionic Framework facilitó el proceso de desarrollo, existieron obstáculos que son propios del aprendizaje inicial en el desarrollo móvil. En particular significó un desafío afrontar los diferentes errores de programación que

surgían dependiendo del entorno de ejecución. Al ejecutar la aplicación en un dispositivo físico surgían errores que no aparecen al ejecutar en un navegador web. Otro de los desafíos surgió al publicar la aplicación en Google Play. Las políticas y requisitos de la plataforma presentan varias restricciones y las mismas pueden variar en lapsos cortos, con lo cual pueden surgir errores nuevos a la hora de publicar una actualización.

Segundo, este proyecto al requerir de un enfoque estructurado y académico, significó una experiencia completa a nivel profesional. Antes de utilizar las tecnologías fue necesario un proceso de abstracción que permita comprender los conceptos teóricos subyacentes. Hacer un análisis inicial de la arquitectura del software a desarrollar ayuda a gestionar de manera eficiente los obstáculos que surgen durante el proceso de desarrollo. Este enfoque es una disciplina que adquirí durante mi paso por la facultad y este trabajo me permitió ponerlo en práctica.

Para finalizar, cabe destacar que si bien este proyecto se realizó como respuesta a una problemática surgida durante la pandemia y ante una propuesta específica, de acceso a través de dispositivos móviles al entorno virtual y de comunicación mediante notificaciones push, es posible seguir mejorando la aplicación móvil teniendo en consideración nuevos requerimientos sin tener que partir de cero. Esta investigación sirve de base para futuros desarrolladores que quieran realizar una personalización sobre la aplicación móvil de Moodle.

Trabajos futuros

Existe un amplio abanico de oportunidades que podrían ser exploradas para mejorar la experiencia de los estudiantes y docentes del entorno AulasWebColegios.

Se espera seguir perfeccionando las características de la nueva aplicación móvil con los siguientes trabajos futuros:

- Compilar la aplicación móvil para sistemas operativos iOS: Gracias a Ionic Framework es sencillo compilar para otros sistemas operativos, el desafío está en buscar la

correcta configuración de APNs con AirNotifier y modificar el payload que se emite desde el servidor push para que la recepción de la notificación en la aplicación sea adecuada.

- Durante el análisis de viabilidad se observó una diferencia en la receptividad hacia el uso de las notificaciones push entre los docentes y estudiantes. En este sentido, se plantea como desafío a futuro investigar sobre el tipo adecuado de notificación para el docente, considerando contenido, estructura y frecuencia de envío. Enfatizando en ejemplos concretos y prácticos de funcionalidades específicas que puedan ser beneficiosas. Un ejemplo concreto sería un aviso de emergencia ante una tarea mal configurada. Como se mencionó anteriormente en este escrito, la tarea es una herramienta de tipo actividad dentro del aula virtual que puede ser configurada. Ante una configuración errónea que impide el funcionamiento esperado de la actividad podría ser de utilidad una notificación de emergencia.
- En relación con el ítem anterior, se propone explorar el registro y desarrollo de nuevos eventos de notificación dentro del sistema Moodle.
- Se plantea una futura investigación sobre la posibilidad de brindar una personalización más completa sobre la función de silenciar notificaciones. Actualmente, en el sistema Moodle, los usuarios tienen la opción de silenciar eventos de notificación referidos a un contexto ya determinado en el sistema (imagen 2). Sería interesante desarrollar una funcionalidad para personalizar el contexto y silenciar las notificaciones de manera más específica. Un ejemplo concreto sería pasar de silenciar la notificación de tarea pendiente de todos las aulas virtuales a silenciar específicamente la notificación de tarea pendiente de un aula en particular.
- Como se mencionó en los detalles a considerar del capítulo desarrollo, fue necesario explorar y modificar el controlador que maneja el evento click sobre una notificación. Al ya tener esta área analizada, se plantea el desarrollo de un nuevo controlador que a la hora de manejar el evento tenga en consideración el tipo de notificación entrante. De esta manera el click sobre una notificación no siempre redirigirá a la página de Notificaciones. Un ejemplo concreto sería que el usuario que cliquee sobre una notificación de tarea pendiente sea redirigido a la tarea en cuestión para que pueda completarla.
- Diseño de un plugin para Moodle que le permita al administrador del sistema realizar

la configuración general de las notificaciones, en cuanto a estructura, diseño y componentes (sonidos, imágenes). Teniendo en cuenta los recursos disponibles a nivel de hardware y software.

Exceptuando la propuesta de compatibilidad con el sistema operativo iOS, los trabajos futuros se dirigen hacia la mejora continua del módulo de notificaciones. Sin embargo, se plantea la idea de mejorar y ampliar la aplicación móvil en general, abordando las demás restricciones y limitaciones identificadas en el subcapítulo de análisis tecnológico.

En definitiva, se propone abarcar un horizonte más amplio que permita expandir y fortalecer la nueva aplicación móvil personalizada.

BIBLIOGRAFÍA

Aguín, J. (2011). Arquitectura: Cómo funcionan las notificaciones en los teléfonos móviles. Jasoft.org. Recuperado de <https://www.jasoft.org/Blog/post/Arquitectura-Como-funcionan-las-notificaciones-en-los-telefonos-moviles-iPhone-Windows-Phone>

Albertengo, G., Debele, F., Hassan, W., Stramandino D. (2020). On the performance of web services, google cloud messaging and firebase cloud messaging. <https://www.sciencedirect.com/science/article/pii/S235286481830035X>

Almendra, F. y Martínez de Treviño, J. (2020). E-TUTOR: Herramienta virtual para el Programa de Tutorías. <http://sedici.unlp.edu.ar/handle/10915/118530>

Andreoli, S. (2021). “Modelos híbridos en escenarios educativos en transición”. https://www.exactas.unlp.edu.ar/articulo/2021/6/17/modelos_hibridos_en_escenarios_educativos_en_transicion

Bertone, R., Filippi, J., Lafuente, J., Ballesteros, C., Lafuente, G., Aguirre, S., Mansilla, A.

(2020). Incorporación de la Tecnología Móvil en el Proceso Educativo.
(<http://sedici.unlp.edu.ar/handle/10915/103861>)

Cabrera Medina, J., Sánchez Medina, I., y Rojas, F. (2016). Uso de objetos virtuales de aprendizaje OVAs como estrategia de enseñanza – Aprendizaje Inclusivo y Complementario para los cursos teórico-prácticos.
<https://educacioneningenieria.org/index.php/edi/article/view/602>

Coggiola, L. (2017). Mi Universidad: Una aplicación móvil para mejorar la experiencia de usuario de los estudiantes de la Universidad Nacional de La Plata.
<http://sedici.unlp.edu.ar/handle/10915/72022>

Da Porta, E. (2015). El acceso a las tecnologías de la comunicación. Debates y perspectivas en América Latina. Las significaciones de las TIC en educación.
https://perio.unlp.edu.ar/catedras/comyeduc2/wp-content/uploads/sites/197/2021/05/da_porta_-_el_acceso_a_las_tecnologias_debates_y_perspectivas_en_al.pdf

Developer Resource Centre Moodle. (2023). External Services.
<https://moodledev.io/docs/apis/subsystems/external>

Documentación oficial de Android. (s. f.). Descripción general de notificaciones.
<https://developer.android.com/>

Documentación oficial de Android. (s. f.). Guía de diseño de notificaciones.
<https://m2.material.io/design/platform-guidance/android-notifications.html#types-of-notifications>

Documentación oficial de Apple. (2018). Local and Remote Notifications Overview.
<https://developer.apple.com/library/archive/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/>

Documentación oficial de Firebase Cloud Messaging. (2022). Introducción al servicio.
<https://firebase.google.com/docs/cloud-messaging?hl=es-419>

Fagina, M. (2018): El tecnólogo educativo. <https://roldeltecnologoeducativo.blogspot.com/>

Fernández, L. (2006). Metodologías ágiles. Software Guru. Recuperado de
https://ozarate.net/articulos/arquitectura_sw_sg_2006.pdf

González, A H. Martin, M.M.,. Barletta, C.M.,. Esnaola, F. (2022). Análisis de las acciones en educación digital desarrolladas en la Universidad Nacional de la Plata: programa de apoyo a la educación a distancia ante la pandemia COVID19.
<http://sedici.unlp.edu.ar/handle/10915/139045>

Gonzalez, A. H., & Martín, M. M. (2017). Educación superior a distancia en Argentina: tensiones y oportunidades. Trayectorias universitarias.

González A. H.(2019) Tendencias en el desarrollo de plataformas educativas en el ambiente universitario. Presentación de casos. LAS PLATAFORMAS VIRTUALES EN LA EDUCACIÓN SUPERIOR. Conferencias y comunicaciones de la Jornada de Plataformas Educativas en el Nivel Superior (JoPIEd). San Justo: Editorial UNLaM. pág. 15 - 34. ISBN 978-987-4417-35-0

González, Alejandro Héctor & García Chicote, Gonzalo.(2016). Integración de entornos virtuales de enseñanza y aprendizaje en la educación universitaria.
<http://sedici.unlp.edu.ar/handle/10915/133732>

Ibáñez, J. (2018). Blended learning, más allá de la clase presencial.
<https://dialnet.unirioja.es/servlet/articulo?codigo=6296112>

Ionic Framework. (2023). Overview. <https://ionicframework.com/docs>

Jaime, C, J. (2020). Informe técnico. Dirección General de Educación a Distancia y Tecnologías. Edición extraordinaria en el marco del ASPO. UNLP
<http://sedici.unlp.edu.ar/handle/10915/120345>

Litwin, E. (2012): Prácticas con tecnologías. Praxis educativa. Facultad de Ciencias Humanas. UNLPam.

Malek, N. (2020). AirNotifier: A How-To Guide for Beginners. Tyne Solutions.
https://tyne-solutions.com/2020/12/31/airnotifier_a_how-to_guide_for_beginners/

Marquez, C. (2021). Las TICs en la educación: impacto y situación actual en las escuelas latinoamericanas. <https://aulicum.com/blog/tics-en-la-educacion/>

Mosquera Gende, I. (2019): Las cifras no mienten: la digitalización en las aulas es una realidad a nivel mundial. La universidad en Internet. Recuperado de:
<https://www.unir.net/educacion/revista/las-cifras-no-mienten-la-digitalizacion-en-las-aulas-es-una-realidad-a-nivel-mundial/>

Mujica Sequera, R. (2020). Diferencias entre LMS, LCMS y VLE.
<https://blog.docentes20.com/2021/07/%E2%9C%8D-diferencias-entre-lms-lcms-y-vle-docentes-2-0/>

Nicols, A. (2022). Descripción general de la aplicación Moodle. Developer Resource Centre Moodle. <https://moodledev.io/general/app/overview>

Picherili, M. y Tolosa, M. (2020). Educación en pandemia y desigualdad. Observatorio socioeconómico UCALP.
<https://www.ucalp.edu.ar/wp-content/uploads/2020/09/Educaci%C3%B3n-en-pandemia-y-desigualdad>

Príncipe, A. G., Russo, C., Zangara, A., Esnaola, F., & Salvioli, A. P. S. (2009). La

internacionalización de la educación a distancia: estrategias de abordaje. Presentación del Proyecto Aula Cavila UNLP. RIED. Revista iberoamericana de educación a distancia.

Quizhpe, L. (2019). Revisión sistemática sobre arquitectura push en el análisis de aplicabilidad de ambientes móviles.

<https://dspace.unl.edu.ec/jspui/handle/123456789/22303>

Romanut, L. M. (2016). Desarrollo de asistencia en el selector de actividades de Moodle para facilitar la incorporación de herramientas de trabajo colaborativo. <http://sedici.unlp.edu.ar/handle/10915/59092>

Sanz, C. V., Zangara, M. A., Gonzalez, A. H., De Giusti, A. E., & Ibañez, E. (2004). Design of distance courses in a web learning environment (WebLIDI). Journal of Computer Science & Technology,


Swal, M. (2021). La pandemia y el año que enseñamos por whatsapp: el recurso tecnológico más utilizado en las escuelas secundarias pobres de Bahía Blanca durante el 2020.

Vila Ruiz, T. (2015). Centro de gestión de notificaciones Push para dispositivos móviles basados en IOS y Android. <https://addi.ehu.es/handle/10810/15912>

Zuazo, N. (2019). Los dueños de internet.

<http://catedratos.com.ar/media/Los-duenos-de-Internet.pdf>

Anexo 1: Formulario de sondeo para estudiantes



DIRECCIÓN GENERAL
de educación a distancia y tecnologías
UNIVERSIDAD NACIONAL DE LA PLATA

Estudiantes de AulasWebColegios

Desde la Dirección General de Educación a Distancia y Tecnologías de la UNLP les acercamos la siguiente encuesta de carácter anónimo. La misma tiene por objetivo poder reunir información sobre el uso del entorno virtual AulasWebColegios por parte de los estudiantes, y a corto plazo utilizar la información brindada para mejorar las formas de comunicación y poder incorporar nuevas funcionalidades.

[Iniciar sesión en Google](#) para guardar lo que llevas hecho. [Más información](#)

* Indica que la pregunta es obligatoria

1 - ¿A qué establecimiento perteneces? *

Elige ▼

2 - ¿Alguna vez te olvidaste o no te enteraste de alguna tarea por no haber ingresado * a tiempo a la plataforma AulasWebColegios?

Sí

No

3 - ¿Consideras difícil estar al tanto de las novedades y tareas nuevas dentro del * curso?

Sí

No

4 - Actualmente, ¿Cómo te enteras de las novedades dentro del aula virtual? (Puedes * elegir más de una opción)

Otros en caso de comunicación presencial, a través de blogs, campus, etc.

- Whatsapp
- Ingreso a la plataforma web AulasWebColegios
- Correo electrónico
- Ingresando a la aplicación para celular de Moodle
- Otro: _____

5 - ¿En donde preferirías recibir las notificaciones sobre las novedades del curso? *

- Al correo electrónico
- Al celular
- Ambos
- Ninguna

6 - ¿Consideras que para lograr una mejor experiencia de cursada son necesarias las * notificaciones al celular (avisos al celular)?

Notificar, a través de los foros, cuando: el docente crea una tarea nueva (para entregar), cuando se crea un cuestionario (examen con preguntas), cuando el docente corrige el cuestionario, participación en un foro, eventos del calendario (calendario del curso).

- No
- Si

7 - ¿Conoces la aplicación para celular de Moodle? *

Para ingresar a la plataforma AulasWebColegios o alguna otra plataforma Moodle.

- Si
- No

8 - Si la conoces, Indica las frases que mas te identifican (puedes elegir más de una opción) *

- No la conozco (en la anterior pregunta respondí 'No')
- No me parece útil
- Utilizo otra tecnología para comunicarme y/o estar al tanto de las novedades del curso
- No me llegan las notificaciones al celular, solo me sirve para ver los cursos
- La utilizo pero tiene poca información
- Me permite estar atento a cada curso
- La utilizo pero tiene mucha información

9 - Además de la aplicación para celular de Moodle, ¿utilizas otra tecnología para comunicarte y/o estar al tanto de las novedades del curso? (puedes elegir más de una opción) *

- No la conozco (En la anterior pregunta respondí 'No la conozco')
- Si, whatsapp
- Si, correo electrónico
- Si, otros

10 - ¿Qué dispositivo usas normalmente para ingresar al aula virtual? *

- Celular
- Computadora de escritorio
- Notebook
- Tablet
- Otro: _____

11 - En caso de usar el celular, ¿Qué sistema operativo tienes? *

- Android
- IOS (uso Iphone)
- Ninguno (En la anterior no seleccione "Celular")

12 - Si te enteras por diferentes canales de comunicación de las novedades del aula virtual, ¿crees que sería de ayuda unificarlos en una única aplicación? *

Diferentes canales de comunicación como whatsapp, telegram, redes sociales, correo electrónico, blog, campus, etc.

- Si
- No

Enviar

Borrar formulario

Nunca envíes contraseñas a través de Formularios de Google.

Este contenido no ha sido creado ni aprobado por Google. [Notificar uso inadecuado](#) - [Términos del Servicio](#) - [Política de Privacidad](#)

Google Formularios

Anexo 2: Formulario de sondeo para docentes



DIRECCIÓN GENERAL
de educación a distancia y tecnologías
UNIVERSIDAD NACIONAL DE LA PLATA

Docentes de AulasWebColegios

Desde la Dirección General de Educación a Distancia y Tecnologías de la UNLP les acercamos la siguiente encuesta de carácter anónimo. La misma tiene por objetivo poder reunir información sobre el uso del entorno virtual AulasWebColegios por parte de los docentes y a corto plazo utilizar la información brindada para mejorar las formas de comunicación y poder incorporar nuevas funcionalidades.

[Iniciar sesión en Google](#) para guardar lo que llevas hecho. [Más información](#)

* Indica que la pregunta es obligatoria

1 - ¿A qué establecimiento perteneces? *

Elige

2 - ¿Qué asignatura dictas en el aula virtual? *

Tu respuesta

3 - ¿Considera necesario agregar más vías de comunicación para notificar las novedades dentro de un curso en AulasWebColegios? *

Novedades como: creación de tareas (recurso Moodle para agregar tareas), creación de un cuestionario, calificación, participación en un foro, mensajes y eventos del calendario.

Sí

No

4 - Indique el porque de su respuesta en la pregunta anterior *

Tu respuesta

5 - Actualmente, ¿Cómo realizas el envío de novedades del aula virtual? (puede seleccionar más de una opción) *

'Otros' en caso de notificar de manera presencial, a traves de blogs, campus, etc.

- Whatsapp
- Ingreso a la plataforma web AulasWebColegios
- Correo electrónico
- Ingresando a la aplicación para celular de Moodle
- Otro: _____

6 - ¿En donde prefiere recibir las notificaciones sobre las novedades de los cursos a su cargo? (puede elegir más de una opción) *

- Correo
- Celular
- Ambos
- Ninguna

7 - ¿Conoce la aplicación para celular de Moodle? *

- Si
- No

8 - En caso de conocerla, Indique la frase que mas la/lo identifica (puede seleccionar * más de una opción)

- No la conozco (en la anterior pregunta repondí 'No')
- No me parece útil
- Utilizo otra tecnología para comunicarme y/o estar al tanto de las novedades del curso
- Me permite estar atenta/o al curso
- La utilizo pero tiene poca información
- No me llegan las notificaciones al celular, solo me sirve para ver los cursos
- La utilizo pero tiene mucha información
- Otro: _____

9 - ¿Considera que para lograr una mejor experiencia de cursada son necesarias las * notificaciones al celular (avisos al celular)?

- Si
- No

10 - Indique el porque de su respuesta en la pregunta anterior *

Tu respuesta _____

11 - ¿Qué dispositivo usas normalmente para ingresar al aula virtual? (Puede elegir * más de una respuesta)

- Celular
- Computadora de escritorio
- Notebook
- Tablet
- Otro: _____

12 - En caso de usar el celular, ¿Qué sistema operativo tienes? *

- Android
- IOS (uso Iphone)
- Ninguno (En la anterior no seleccione "Celular")

13 - En el caso de utilizar múltiples canales de comunicación para informar las novedades del aula virtual, ¿crees que sería de ayuda unificarlos en una única aplicación? *

- Si
- No

Enviar


Borrar formulario

Nunca envíes contraseñas a través de Formularios de Google.

Este contenido no ha sido creado ni aprobado por Google. [Notificar uso inadecuado](#) - [Términos del Servicio](#) - [Política de Privacidad](#)

Google Formularios

Anexo 3: Formulario para validación de funcionalidad con los usuarios



DIRECCIÓN GENERAL
de educación a distancia y tecnologías
UNIVERSIDAD NACIONAL DE LA PLATA

Pruebas - App AulasWeb Colegios

Desde la Dirección General de Educación a Distancia y Tecnologías de la UNLP les acercamos la siguiente encuesta de carácter anónimo. La misma tiene por objetivo poder reunir información sobre las pruebas realizadas sobre la aplicación Moodle de 'AulasWeb Colegios', y a corto plazo utilizar la información brindada para mejorar el modulo de notificaciones.

[Iniciar sesión en Google](#) para guardar lo que llevas hecho. [Más información](#)

* Indica que la pregunta es obligatoria

1 - Funcionalidad "envío y recepción de notificaciones push": *

Pude realizar la prueba y el comportamiento de la funcionalidad es correcto

Pude realizar la prueba pero el comportamiento de la funcionalidad NO es correcto

No pude realizar la prueba

2 - Funcionalidad "borrar notificación": *

Pude realizar la prueba y el comportamiento de la funcionalidad es correcto

Pude realizar la prueba pero el comportamiento de la funcionalidad NO es correcto

No pude realizar la prueba

3 - Funcionalidad "Destacar Notificaciones": *

Pude realizar la prueba y el comportamiento de la funcionalidad es correcto

Pude realizar la prueba pero el comportamiento de la funcionalidad NO es correcto

No pude realizar la prueba