

**Implementación de un sistema de notificaciones push web con firebase para el sitio
recursos-biblicos.com**

Diego Andrés Castilla Lázaro

Asesor

Rafael Perez Holguin

Universidad Nacional Abierta y a Distancia UNAD
Escuela de Ciencias Básicas, Tecnología e Ingeniería ECTBI
Tecnología en Desarrollo de Software

2025

Dedicatoria

A mi Creador, fuente de toda sabiduría y fortaleza, por la inspiración, la guía y la oportunidad de emprender y culminar este proyecto. A mi familia, por su inquebrantable apoyo, paciencia y por enseñarme el valor del conocimiento y la perseverancia, pilares fundamentales para alcanzar esta meta.

Agradecimientos

Ante todo, elevo mi más profundo agradecimiento a Dios, mi Creador. A mis docentes y a la UNAD, quienes guiaron mis pasos en el fascinante mundo del desarrollo de software, por la formación integral recibida y por proveer los espacios y recursos que hicieron posible mi desarrollo académico y profesional.

A mi familia por su comprensión, aliento y apoyo constante, especialmente en los momentos de mayor desafío. Al equipo detrás de recursos-biblicos.com, por la confianza depositada y por permitirme aplicar mis conocimientos en un proyecto con un propósito tan significativo.

Resumen

Este proyecto de grado describe el proceso de análisis, diseño, desarrollo e implementación de un sistema de notificaciones push web para el sitio recursos-biblicos.com. La solución utiliza Firebase Cloud Messaging (FCM) API HTTP v1 como motor de envío, con un backend desarrollado en PHP y una base de datos MySQL/MariaDB para la gestión de suscriptores y el seguimiento de interacciones. El sistema permite a los administradores del sitio, a través de un panel de control (dashboard) con autenticación básica, componer y enviar notificaciones personalizadas (título, mensaje, imagen opcional y URL de redirección) a los usuarios suscritos. Se incluyen funcionalidades para visualizar estadísticas básicas como el total de suscriptores y el conteo de clics por notificación, presentados en formato tabular y gráfico. Adicionalmente, el sistema realiza el seguimiento de clics en las notificaciones y la depuración automática de tokens FCM que Firebase reporta como inválidos o desregistrados. La configuración de parámetros sensibles se gestiona a través de variables de entorno para mayor seguridad y flexibilidad. Las tecnologías frontend incluyen HTML5, CSS3 con Bootstrap 4, y JavaScript para la interactividad del dashboard y la lógica del Service Worker.

Palabras Clave: Notificaciones Push, Firebase Cloud Messaging, PHP, MySQL, Dashboard, Web Push, JavaScript, cURL Multi, Cron Job, API HTTP v1, Control de Versiones.

Abstract

This degree project describes the process of analysis, design, development and implementation of a web push notification system for the site recursos-biblicos.com. The solution uses Firebase Cloud Messaging (FCM) API HTTP v1 as the sending engine, with a backend developed in PHP and a MySQL/MariaDB database for subscriber management and interaction tracking. The system allows site administrators, through a dashboard with basic authentication, to compose and send personalized notifications (title, message, optional image and redirection URL) to subscribed users. Functionality is included to display basic statistics such as total subscribers and click count per notification, presented in tabular and graphical format. Additionally, the system tracks clicks on notifications and automatically purges FCM tokens that Firebase reports as invalid or unregistered. Sensitive parameter settings are managed through environment variables for security and flexibility. Frontend technologies include HTML5, CSS3 with Bootstrap 4, and JavaScript for dashboard interactivity and Service Worker logic.

Keywords: Push Notifications, Firebase Cloud Messaging, PHP, MySQL, Dashboard, Web Push, JavaScript, cURL Multi, Cron Job, HTTP v1 API, Version Control.

Tabla de Contenido

Introducción	12
Justificación	14
Objetivos	16
Objetivo General	16
Objetivos Específicos	16
Planteamiento del Problema	17
Alcance	19
Funcionalidades Implementadas:	19
Suscripción de Usuarios:	19
Panel de Administración (Dashboard):	19
Envío de Notificaciones:	19
Seguimiento de Clics:	19
Estadísticas en el Dashboard:	20
Gestión de Tokens:	20
Configuración y Logs:	20
Funcionalidades Fuera del Alcance Actual (Posibles Mejoras Futuras):	20
Población Objetivo	22
Administradores del Sitio Web (recursos-biblicos.com):	22
Suscriptores y Visitantes del Sitio Web (recursos-biblicos.com):	22
Marco Teórico	23
Descripción de la Entidad: recursos-biblicos.com	23
Misión	23
Visión	23
Valores	23
Propuesta de Valor Única (PVU)	24
Organigrama	24
Procesos y Procedimientos que se Sistematizarán	26
Proceso de Suscripción a Notificaciones	26
Proceso de Envío de Notificaciones desde el Panel de Administración	27
Proceso de Seguimiento de Clics en Notificaciones	28

Proceso de Limpieza Automática de Tokens Inválidos	29
Marco Conceptual	30
Técnicas y Herramientas para el Desarrollo del Proyecto	30
Marco Legal	32
Licencia del Software.....	32
Privacidad de Datos de los Suscriptores	32
Metodología	34
Justificación de la Metodología.....	34
Fases de la Metodología.....	35
Fase 1: Análisis y Especificación de Requisitos.....	35
Fase 2: Diseño de la Arquitectura y la Interfaz	35
Fase 3: Desarrollo de los Componentes del Sistema.....	35
Fase 4: Validación y Pruebas Funcionales.....	36
Encuesta de Viabilidad del Proyecto.....	37
Diseño de la Encuesta	37
Resumen y Análisis de Resultados de la Encuesta.....	37
Frecuencia de Visita:.....	37
Interés en Recibir Notificaciones Push:.....	38
Contenido de Mayor Interés para Notificaciones:.....	39
Percepción sobre la Mejora del Compromiso:	40
Dispositivo Preferido para la Recepción:	41
Conclusión de la Encuesta de Viabilidad.....	42
Planificación (Cronograma de Trabajo).....	43
Explicación del Cronograma.....	43
Fase 1: Análisis y Especificación (Mes 1).....	43
Fase 2: Diseño (Mes 2).....	44
Fase 3: Desarrollo (Mes 2 y 3)	44
Fase 4: Validación (Mes 3)	44
Presupuesto	45
Costo Total Monetario Estimado del Proyecto.....	46
Desarrollo y Funcionalidad del Sistema	47
Base de Datos	47

Esquema de la Base de Datos.....	47
Descripción de Tablas y Campos	48
Subscribers_push:.....	48
Clicks_notificaciones_push:.....	48
Programación más Representativa	52
Obtención y Preparación de Datos:	52
Autenticación con Firebase (FCM)	53
Implementación de Envío por Lotes (cURL Multi)	54
Procesamiento de Respuestas y Gestión de Errores	57
Limpieza Automática de Tokens Inválidos.....	58
URL del Proyecto Web Funcional y Árbol de Directorios.....	59
Árbol de Directorios del Proyecto.....	59
Explicación del Árbol de Directorios.....	60
Wp_push/ (Directorio Raíz del Proyecto)	60
Public/	60
Src/.....	61
Storage/	61
Archivos en la Raíz (wp_push/)	62
Conclusiones.....	63
Referencias Bibliográficas	65

Lista de Figuras

Figura 1	<i>Organigrama de recursos-biblicos.com</i>	25
Figura 2	<i>Diagrama del Proceso de Suscripción a Notificaciones</i>	26
Figura 3	<i>Diagrama del Proceso de Envío de Notificaciones</i>	27
Figura 4	<i>Diagrama del Proceso de Seguimiento de Clics</i>	28
Figura 5	<i>Diagrama del Proceso de Limpieza Automática</i>	29
Figura 6	<i>Gráfico de Frecuencia de Visita de los Encuestados</i>	38
Figura 7	<i>Gráfico de Interés en Recibir Notificaciones</i>	39
Figura 8	<i>Gráfico de Preferencias de Contenido para Notificaciones</i>	40
Figura 9	<i>Gráfico de Percepción sobre la Mejora del Compromiso</i>	41
Figura 10	<i>Gráfico de Percepción sobre la Mejora del Compromiso</i>	42
Figura 11	<i>Diagrama de GANTT del proyecto</i>	43
Figura 12	<i>Diagrama MER de la Base de Datos del Sistema</i>	47
Figura 13	<i>Secuencia de Suscripción del Usuario a Notificaciones Push</i>	50
Figura 14	<i>Secuencia de Envío Directo de Notificaciones Push</i>	51
Figura 15	<i>Recuperación de Tokens de Suscriptores</i>	53
Figura 16	<i>Obtención del Token de Acceso a FCM</i>	54
Figura 17	<i>Inicialización del Proceso de Envío por Lotes</i>	55
Figura 18	<i>Preparación, Procesamiento y Ejecución del Lote</i>	56
Figura 19	<i>Procesamiento de Respuestas del Lote</i>	57
Figura 20	<i>Eliminación de Tokens Inválidos de la Base de Datos</i>	58

Lista de Tablas

Tabla 1 <i>Desglose de Costos y Recursos Estimados del Proyecto</i>	45
--	----

Lista de Apéndices

Apéndice A <i>Encuesta de Viabilidad del Proyecto</i>	67
--	----

Introducción

En la era digital actual, la capacidad de mantener a los usuarios informados y comprometidos con el contenido de un sitio web es crucial para su éxito y relevancia. Los sitios web con actualizaciones frecuentes, como recursos-biblicos.com, que ofrece material de estudio y recursos espirituales, pueden beneficiarse enormemente de herramientas que faciliten la comunicación directa y oportuna con su audiencia. Las notificaciones push web han surgido como un canal de comunicación efectivo, permitiendo a los sitios enviar mensajes directamente al navegador del usuario, incluso cuando este no tiene el sitio web abierto activamente.

El proyecto de grado “Implementación de un Sistema de Notificaciones Push Web con Firebase para el Sitio recursos-biblicos.com” se enfoca en el desarrollo de un sistema integral de notificaciones push web específicamente diseñado para las necesidades de recursos-biblicos.com. Utilizando tecnologías robustas y ampliamente adoptadas como PHP para el backend, MySQL/MariaDB para la persistencia de datos, y Firebase Cloud Messaging (FCM) como el motor de envío de notificaciones, se ha construido una solución que abarca desde la suscripción del usuario hasta el envío masivo y el seguimiento de la interacción.

El sistema cuenta con un panel de administración (dashboard) que simplifica la tarea de crear y enviar notificaciones personalizadas, incluyendo elementos multimedia y enlaces de redirección. Además, proporciona métricas básicas sobre el rendimiento de estas notificaciones y el crecimiento de la base de suscriptores.

La presente tesis documenta todo el ciclo de vida del proyecto, desde el planteamiento del problema y la justificación, pasando por el diseño de la arquitectura, la selección de tecnologías, el desarrollo de los componentes clave, hasta las pruebas y la implementación. Se detallan los marcos teórico, conceptual y legal que sustentan el proyecto, así como la metodología de

desarrollo empleada. Finalmente, se presentan las conclusiones obtenidas y se proponen líneas de trabajo futuras para la evolución del sistema.

Justificación

El sitio web recursos-biblicos.com, dedicado a la difusión de material de estudio bíblico y recursos espirituales, genera contenido de forma regular que es de interés para una comunidad de usuarios específica. Actualmente, la comunicación de nuevo contenido o anuncios importantes puede depender de que los usuarios visiten activamente el sitio o se suscriban a listas de correo, métodos que pueden no ser siempre los más inmediatos o efectivos para captar la atención.

La implementación de un sistema de notificaciones push web ofrece una solución directa y proactiva para este desafío, el cual permite:

Mejorar el Compromiso del Usuario: Al enviar alertas instantáneas sobre nuevo contenido, estudios o eventos, se fomenta que los usuarios regresen al sitio con mayor frecuencia.

Comunicación Inmediata: Las notificaciones push llegan directamente al navegador del usuario, siendo una forma de comunicación más rápida que el correo electrónico para anuncios urgentes o contenido de tiempo limitado.

Incrementar el Tráfico Dirigido: Cada notificación puede incluir una URL específica, dirigiendo al usuario exactamente al contenido relevante, mejorando la experiencia y la efectividad de la comunicación.

Modernización de la Plataforma: Adoptar tecnologías como las notificaciones push web alinea a recursos-biblicos.com con las prácticas modernas de interacción digital, mejorando su percepción y alcance.

La necesidad de un sistema propio, en lugar de depender exclusivamente de plugins de terceros (comunes en plataformas como WordPress), surge del deseo de tener un control más granular sobre la funcionalidad, la personalización, la gestión de datos de suscriptores y la

integración con la lógica específica del sitio. Además, el desarrollo de este sistema como proyecto de grado permite aplicar y consolidar los conocimientos adquiridos en la carrera de Tecnología en Desarrollo de Software en un caso práctico y con potencial de utilidad real.

Objetivos

Objetivo General

Implementar un sistema funcional de notificaciones push web para el sitio recursos-biblicos.com, utilizando PHP, MySQL y Firebase Cloud Messaging, que permita a los administradores enviar mensajes a los suscriptores y visualizar estadísticas básicas de interacción.

Objetivos Específicos

Analizar los requisitos funcionales y no funcionales para un sistema de comunicación directa en el sitio web recursos-biblicos.com, con el fin de definir las especificaciones técnicas para la gestión de suscriptores, el envío de notificaciones y la visualización de estadísticas.

Diseñar la arquitectura del sistema, el esquema de la base de datos en MySQL para el almacenamiento de tokens y métricas de clics, y la interfaz de usuario del panel de administración.

Desarrollar los componentes de la plataforma utilizando PHP para el backend y HTML/CSS/JavaScript para el frontend, implementando la lógica para la suscripción de usuarios, el envío de notificaciones en lotes mediante cURL Multi, y la comunicación con la API de Firebase Cloud Messaging.

Validar el correcto funcionamiento del sistema de notificaciones push a través de pruebas funcionales, verificando que la recepción de mensajes, el seguimiento de clics y la presentación de estadísticas en el dashboard cumplan con los requisitos definidos.

Planteamiento del Problema

El sitio web recursos-biblicos.com se ha consolidado como una plataforma de alto valor para su comunidad, caracterizada por la publicación constante de contenido de estudio y recursos espirituales. La efectividad de su misión divulgativa depende fundamentalmente de la capacidad para que las nuevas publicaciones alcancen a su audiencia de manera oportuna.

Actualmente, el modelo de comunicación de la plataforma se basa principalmente en métodos pasivos, donde la iniciativa recae en el usuario. Estos métodos incluyen:

Visitas Directas del Usuario. Se depende de que los usuarios recuerden y decidan activamente visitar el sitio web para descubrir nuevo contenido

Comunicación por Correo Electrónico. Aunque útil, este canal enfrenta desafíos como la saturación de las bandejas de entrada y los filtros de spam. No garantiza una notificación inmediata ni una interacción en tiempo real.

La dependencia de estos canales genera varios problemas interconectados:

Pérdida de Inmediatez. El contenido de actualidad o los anuncios importantes pierden su carácter de urgencia al no ser comunicados de forma instantánea.

Dificultad para Medir el Interés. Al no tener un canal de notificación directo, es difícil medir el interés inmediato que genera una nueva publicación, más allá del análisis de tráfico general.

Ausencia de una Infraestructura Técnica de Notificación. A nivel técnico, la plataforma carece de un sistema integrado que permita gestionar una lista de suscriptores para un canal de comunicación directo.

Estos elementos configuran el problema central: la ausencia de un canal de comunicación directo, instantáneo y personalizable entre recursos-biblicos.com y su comunidad limita

significativamente la capacidad del sitio para maximizar el alcance de su contenido, fomentar un compromiso constante y mejorar la retención de su audiencia. Esto da origen a la necesidad de investigar, diseñar y desarrollar una solución tecnológica que resuelva esta deficiencia comunicacional.

Alcance

El presente proyecto comprende el análisis, diseño, desarrollo y validación de un sistema de notificaciones push web con las siguientes funcionalidades:

Funcionalidades Implementadas

Suscripción de Usuarios

Mecanismo para que los usuarios se suscriban a las notificaciones push a través de su navegador.

Almacenamiento de tokens FCM en la base de datos.

Panel de Administración (Dashboard)

Acceso protegido mediante autenticación básica HTTP.

Formulario para la creación de notificaciones con título, mensaje, URL de imagen y URL de redirección.

Botón para iniciar el proceso de envío de notificaciones.

Envío de Notificaciones

Lógica de backend (PHP) para comunicarse con la API FCM v1.

Obtención de token de acceso OAuth 2.0 para FCM.

Envío de notificaciones a todos los suscriptores registrados.

Implementación de envío en lotes mediante cURL Multi.

Seguimiento de Clics

Generación de una URL de seguimiento para cada notificación que pasa por registrar-clic.php.

Registro de cada clic (asociado a un notification_id y redirect_url) en la base de datos.

Redirección final del usuario a la URL de destino.

Estadísticas en el Dashboard

Visualización del número total de suscriptores activos.

Tabla paginada mostrando el historial de clics agrupados por notification_id y redirect_url, incluyendo total de clics y fecha del primer clic.

Gráfico de barras (con Chart.js) mostrando el total de clics por notification_id para la página actual de la tabla.

Gestión de Tokens

Detección de tokens inválidos (UNREGISTERED, INVALID_ARGUMENT) a partir de las respuestas de FCM.

Eliminación automática de estos tokens inválidos de la tabla subscribers_push.

Configuración y Logs

Uso de archivo .env para configuraciones sensibles.

Sistema de logging para errores PHP, proceso de envío y actividad del worker.

Funcionalidades Fuera del Alcance Actual (Posibles Mejoras Futuras)

Encolado de tareas de envío en una tabla de base de datos y procesamiento de la cola de envíos en segundo plano mediante un script worker ejecutado por un Cron Job. (esto para manejar volúmenes extremadamente grandes de suscriptores y evitar timeouts HTTP prolongados.)

Programación de notificaciones para envío en una fecha/hora futura desde el dashboard.

Segmentación de usuarios para enviar notificaciones a grupos específicos.

Estadísticas avanzadas (CTR por notificación, tasa de entrega, fallos detallados por campaña).

Interfaz de usuario más elaborada o integración con un framework frontend moderno.

Sistema de autenticación de usuarios más robusto para el dashboard que HTTP Basic Auth.

Limpieza o archivado de tareas antiguas en la tabla `notification_jobs`.

Reintentos automáticos para envíos fallidos a tokens específicos (que no sean inválidos).

Población Objetivo

El sistema de notificaciones push web está dirigido a dos grupos principales dentro del contexto de recursos-biblicos.com:

Administradores del Sitio Web (recursos-biblicos.com)

Son los usuarios primarios del panel de administración (dashboard).

Se beneficiarán de una herramienta para comunicarse de manera eficiente y directa con su audiencia, anunciar nuevo contenido, eventos o información relevante.

Podrán medir de forma básica la interacción con sus notificaciones.

Suscriptores y Visitantes del Sitio Web (recursos-biblicos.com)

Son los usuarios finales que optan por recibir las notificaciones.

Se beneficiarán al recibir actualizaciones oportunas sobre contenido de su interés directamente en sus navegadores, mejorando su acceso a la información y su compromiso con el sitio.

Marco Teórico

Descripción de la Entidad: recursos-biblicos.com

Recursos-biblicos.com es un sitio web dedicado a la provisión y difusión de materiales de estudio bíblico, recursos teológicos, sermones, comentarios, y contenido relacionado con la fe y la espiritualidad cristiana. Su objetivo principal es servir como una plataforma de apoyo para individuos, estudiantes de teología, pastores y cualquier persona interesada en profundizar su conocimiento y comprensión de las Escrituras y temas afines. El sitio se caracteriza por la publicación regular de nuevo contenido, incluyendo artículos, resúmenes de lecciones, presentaciones y otros materiales multimedia.

Misión

Ser una fuente confiable y accesible de recursos espirituales y de estudio bíblico, proveyendo herramientas de alta calidad a individuos, estudiantes, pastores y líderes laicos para facilitar un mayor conocimiento y comprensión de las Sagradas Escrituras, y fomentar el crecimiento espiritual de la comunidad de habla hispana a nivel global.

Visión

Convertirse en una de las plataformas de referencia en el mundo hispanohablante para el estudio bíblico y la distribución de recursos bíblicos digitales, reconocida por la profundidad de su contenido, la fidelidad a las Escrituras y el uso de la tecnología para alcanzar y servir a una audiencia cada vez más amplia y diversa.

Valores

Fidelidad bíblica: Nos basamos en la Palabra de Dios como única autoridad.

Accesibilidad: Todo creyente merece acceso libre y gratuito a material edificante.

Excelencia: Buscamos la excelencia en la presentación y relevancia de cada recurso.

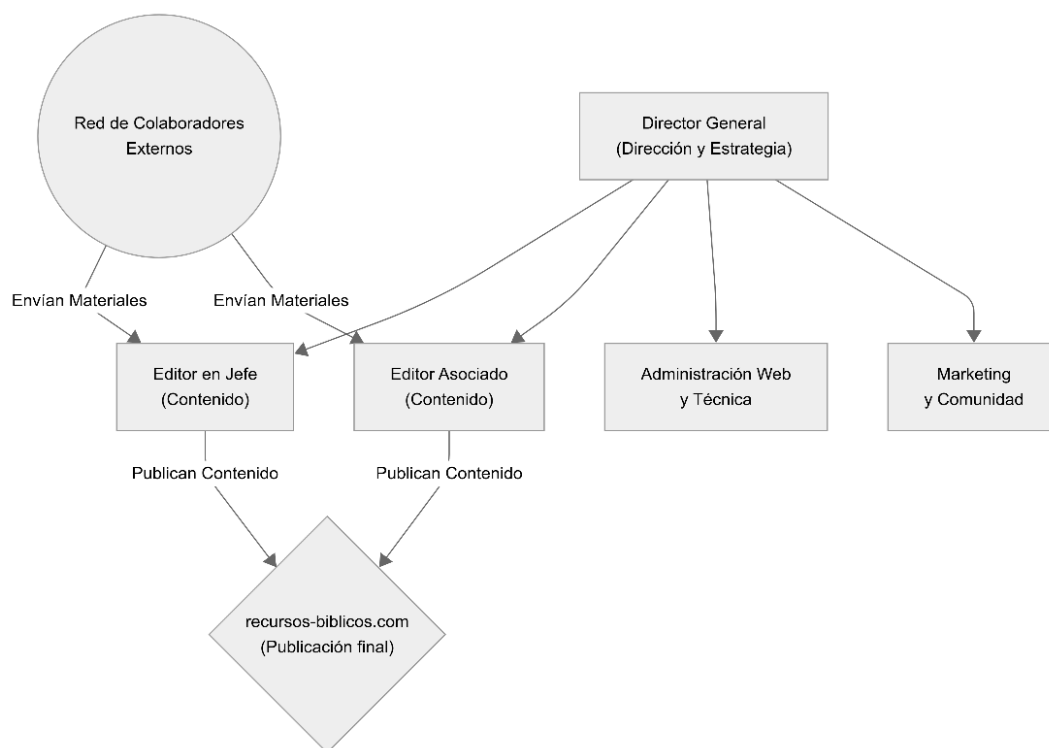
Servicio: Servimos a la iglesia global con humildad y entrega.

Propuesta de Valor Única (PVU)

Recursos Bíblicos es tu asistente ministerial digital. Te ofrecemos una curación de materiales listos para usar que te ahorran tiempo de preparación y elevan la calidad de tu enseñanza. Te damos la confianza para predicar y enseñar con autoridad, sabiendo que tu mensaje está construido sobre una base firme. Profundidad bíblica, sin complicaciones.

Organigrama

Siendo un ministerio en línea, la estructura de recursos-biblicos.com es compacta y funcional, basada en roles colaborativos más que en una jerarquía corporativa rígida.

Figura 1*Organigrama de recursos-biblicos.com*

Nota. El diagrama muestra la estructura operativa del sitio web Recursos-Biblicos.com, donde el director general coordina contenido, tecnología y comunidad, con apoyo editorial y colaboradores externos. Tomado de: Organigrama del ministerio (2025), recursos-Biblicos.com, <https://www.recursos-biblicos.com/acerca-de>

Dada la naturaleza de su contenido y su audiencia, mantener una comunicación fluida y constante con los usuarios es fundamental para el cumplimiento de su misión. La capacidad de informar oportunamente sobre nuevas publicaciones o recursos disponibles es clave para fomentar el engagement y asegurar que el material llegue a quienes lo buscan.

Procesos y Procedimientos que se Sistematizarán

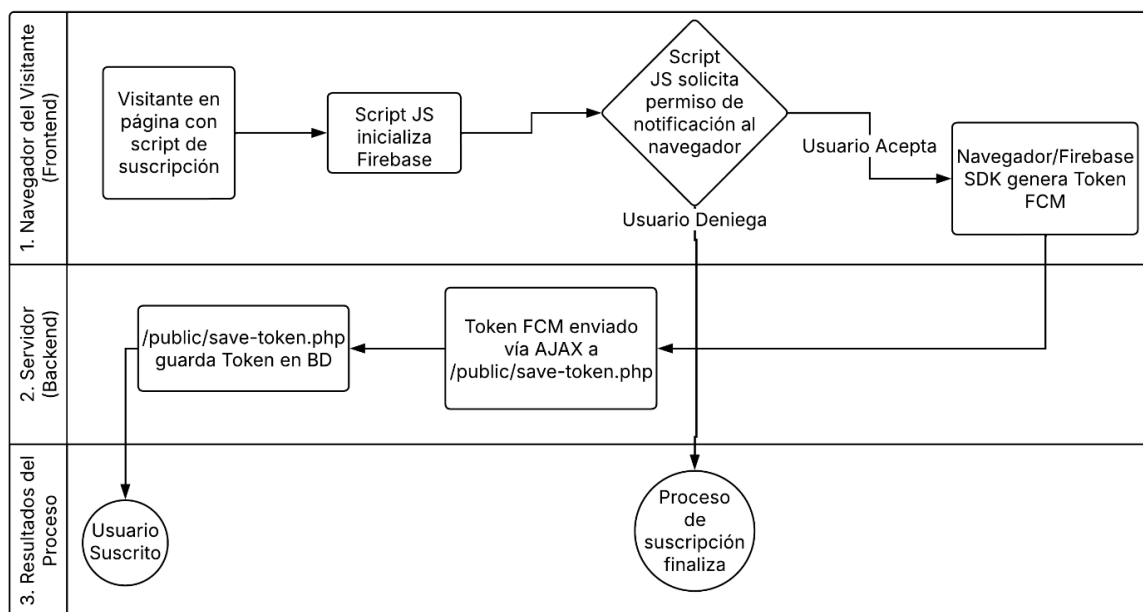
El sistema de notificaciones push web desarrollado busca sistematizar y mejorar varios procesos clave relacionados con la comunicación y la interacción con los usuarios de recursos-biblicos.com:

Proceso de Suscripción a Notificaciones

Este proceso permite a los visitantes del sitio web optar por recibir actualizaciones directas en sus navegadores. El sistema facilita la solicitud de permiso al usuario y, una vez concedido, registra un token de identificación único (token FCM) asociado a ese navegador/dispositivo en la base de datos del proyecto.

Figura 2

Diagrama de Flujo del Proceso de Suscripción a Notificaciones



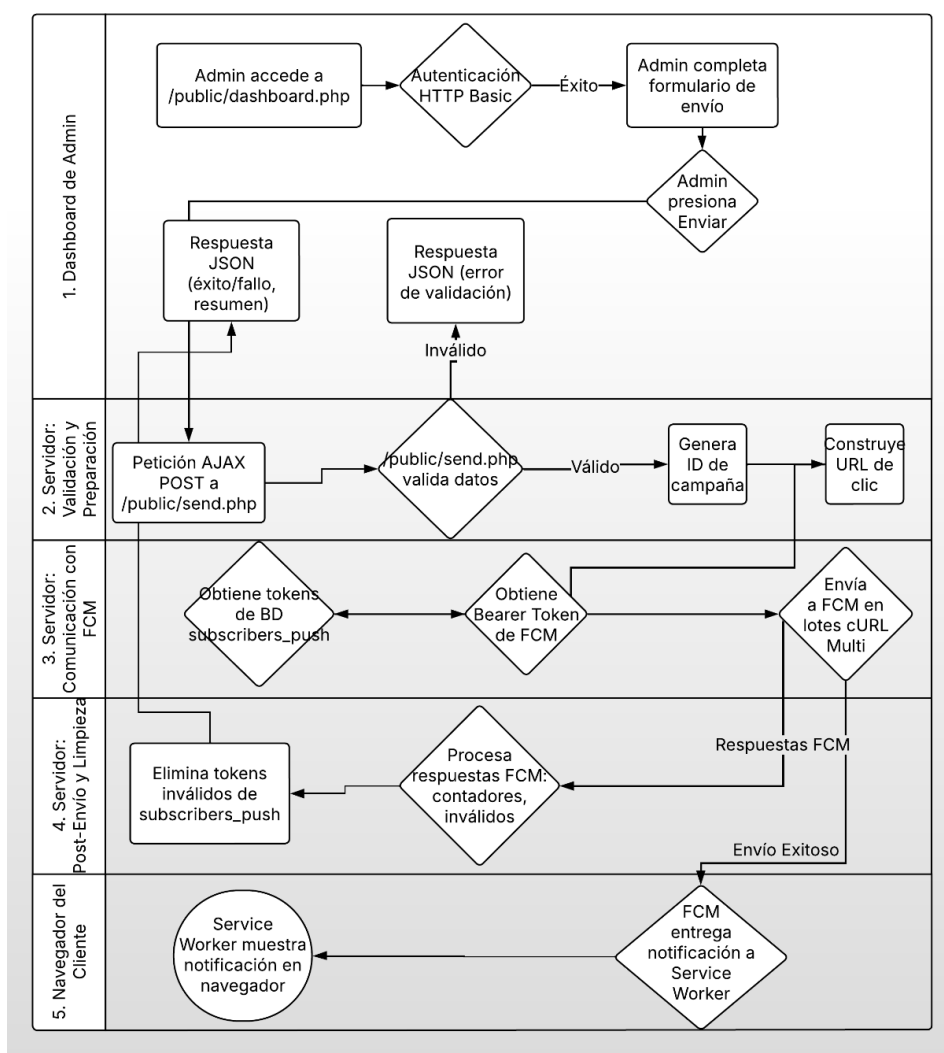
Nota. El diagrama ilustra los pasos que sigue un usuario, desde la visita a la página hasta la confirmación de su suscripción, incluyendo la interacción con el navegador y el servidor.

Proceso de Envío de Notificaciones desde el Panel de Administración

Este proceso permite a los administradores autenticados del sitio crear y enviar mensajes personalizados a todos los suscriptores. Para manejar la carga de envíos a un gran número de usuarios se procesa la solicitud de envíos mediante cURL Multi.

Figura 3

Diagrama de Flujo del Proceso de Envío de Notificaciones desde el Panel de Administración.



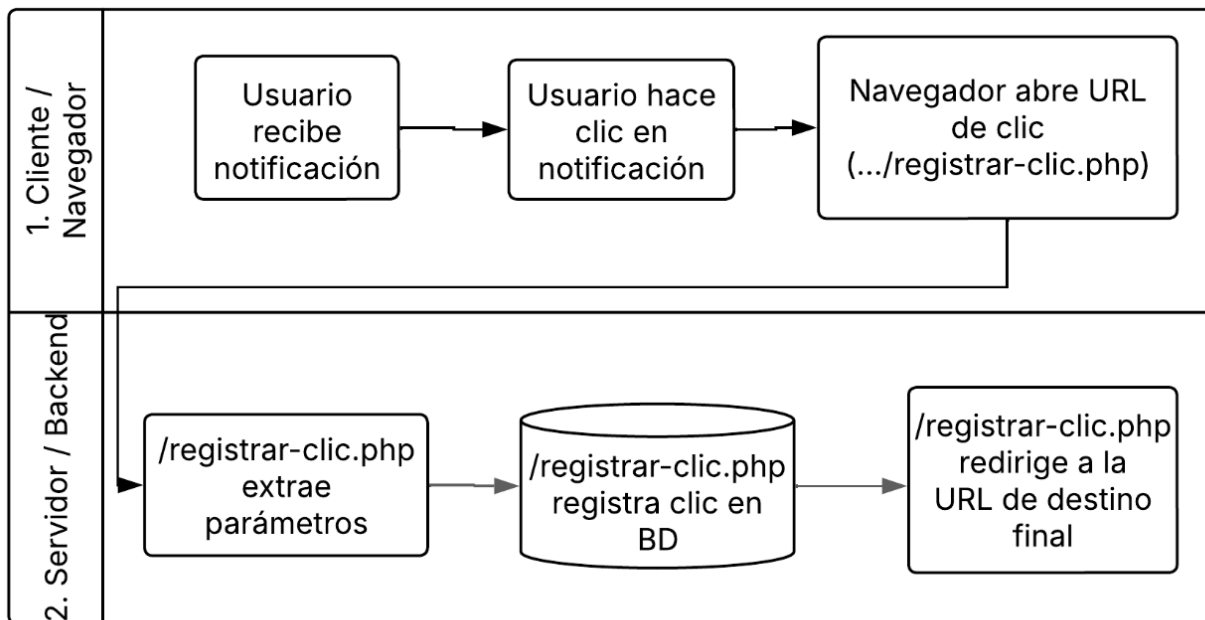
Nota. Representación visual del flujo de trabajo del administrador para enviar notificaciones masivas de forma directa.

Proceso de Seguimiento de Clics en Notificaciones

Permite medir la interacción básica de los usuarios con las notificaciones. Cada notificación contiene una URL de acción que primero dirige al usuario a un script de seguimiento del sistema (`registrar-clic.php`) antes de redirigirlo al contenido final.

Figura 4

Diagrama de Flujo del Proceso de Seguimiento de Clics en Notificaciones



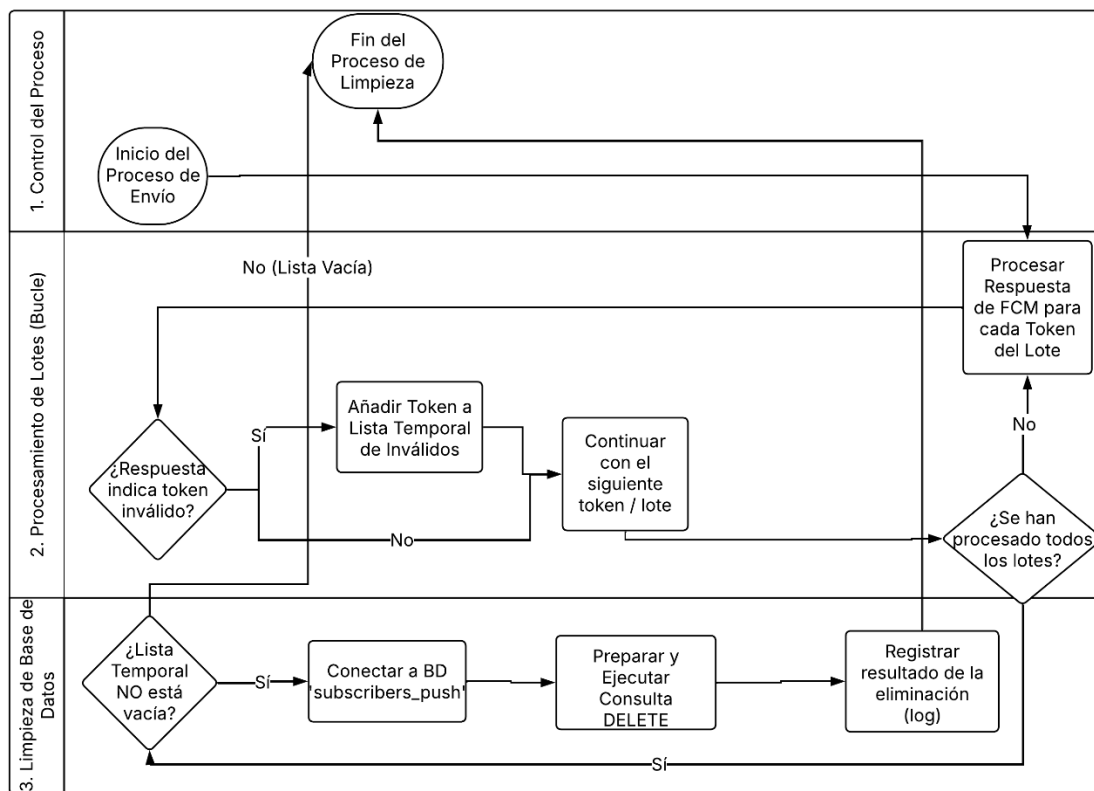
Nota. El diagrama detalla cómo el sistema intercepta el clic de un usuario en una notificación para registrar el evento en la base de datos antes de redirigirlo a la URL de destino final.

Proceso de Limpieza Automática de Tokens Inválidos

Durante el envío de notificaciones (realizado por `send.php`), FCM puede indicar que algunos tokens ya no son válidos. El sistema identifica estos tokens y los elimina de la base de datos `subscribers_push`.

Figura 5

Diagrama de Flujo del Proceso de Limpieza Automática de Tokens Inválidos



Nota. Diagrama que describe cómo el sistema identifica y elimina de la base de datos los tokens que son reportados como inválidos por Firebase Cloud Messaging durante el proceso de envío.

Marco Conceptual

Técnicas y Herramientas para el Desarrollo del Proyecto

El desarrollo del sistema se apoyó en un conjunto de tecnologías y herramientas estándar en el desarrollo web moderno, seleccionadas por su adecuación a los requisitos del proyecto, su robustez y la disponibilidad de documentación y comunidad.

PHP (Hypertext Preprocessor). Lenguaje de scripting del lado del servidor utilizado para toda la lógica de backend, incluyendo la interacción con la base de datos, el manejo de peticiones AJAX desde el dashboard, la comunicación con la API de Firebase, y la ejecución de tareas en segundo plano (worker). Versión 7.4 o superior.

MySQL / MariaDB. Sistema de gestión de bases de datos relacional para almacenar información persistente como los tokens de los suscriptores, los registros de clics y las tareas de envío encoladas.

Firebase Cloud Messaging (FCM). Servicio de Google utilizado como la plataforma principal para el envío de notificaciones push. El proyecto interactúa con su API HTTP v1.

HTML5, CSS3, JavaScript. Tecnologías estándar del lado del cliente para la estructura, estilo e interactividad del dashboard y la página de suscripción. El JavaScript se utiliza para las llamadas AJAX, la manipulación del DOM, la lógica del Service Worker y la renderización de gráficos.

Bootstrap 4. Framework frontend CSS para el diseño responsivo y la maquetación de la interfaz de usuario del dashboard.

Chart.js (v3.7.0). Librería JavaScript para la creación de gráficos, utilizada en el dashboard para visualizar las estadísticas de clics.

Composer: Gestor de dependencias para PHP, utilizado para instalar y gestionar la librería vlucas/phpdotenv.

vlucas/phpdotenv. Librería PHP para cargar variables de entorno desde un archivo .env, permitiendo una gestión segura de la configuración.

HTTP Basic Auth. Mecanismo de autenticación simple utilizado para proteger el acceso al dashboard y a los endpoints de API.

cURL (PHP Extension). Utilizada para realizar las peticiones HTTP a la API de FCM, con curl_multi para el envío eficiente de notificaciones en lotes de forma paralela.

JSON (JavaScript Object Notation). Formato estándar utilizado para el intercambio de datos entre el frontend (dashboard AJAX) y el backend (scripts PHP como get_stats.php, send.php, get_job_status.php).

Marco Legal

Licencia del Software

El software desarrollado como parte de este proyecto de grado se rige bajo los siguientes términos:

No se concede permiso para redistribuir, modificar o utilizar este software con fines comerciales o públicos sin el consentimiento explícito y por escrito del autor. Para fines académicos y de evaluación por parte de la institución designada, se otorga un permiso limitado de uso y revisión del código fuente y la aplicación funcional.

Privacidad de Datos de los Suscriptores

El sistema maneja tokens de registro de Firebase Cloud Messaging (FCM), los cuales son identificadores únicos asociados a una instancia de navegador o dispositivo de un usuario. Por lo tanto, se han considerado los siguientes aspectos de privacidad:

Consentimiento Explícito. El usuario debe aceptar recibir notificaciones para que su token sea generado y almacenado.

Finalidad del Tratamiento. Los tokens FCM recolectados se utilizan con la única finalidad de enviar las notificaciones solicitadas por el usuario, relacionadas con el contenido y anuncios del sitio recursos-biblicos.com. No se utilizarán para otros fines ni se compartirán con terceros sin consentimiento adicional.

Seguridad de los Datos. Los tokens se almacenan en la base de datos del proyecto.

Transparencia y Derechos del Usuario. Se recomienda que el sitio web recursos-biblicos.com disponga de una Política de Privacidad clara y accesible que informe a los usuarios sobre la recolección y uso de tokens para notificaciones.

Cumplimiento Normativo (Colombia). Para el contexto colombiano, el tratamiento de estos datos debe considerar los principios y obligaciones establecidos en la Ley 1581 de 2012 (Régimen General de Protección de Datos Personales) y sus decretos reglamentarios, en lo que respecta a la obtención del consentimiento, la finalidad, la seguridad y los derechos de los titulares.

Metodología

Justificación de la Metodología

Para el desarrollo del Sistema de Notificaciones Push Web para recursos-biblicos.com, se seleccionó una Metodología Iterativa e Incremental adaptada. Esta elección se fundamenta en varias consideraciones clave:

Gestión de la Complejidad. El proyecto, aunque con un objetivo claro, involucra múltiples componentes (frontend, backend, base de datos, API externa, Service Worker). Un enfoque iterativo permite abordar estos componentes en bloques manejables, reduciendo la complejidad inicial.

Flexibilidad. Una metodología iterativa permite realizar ajustes y adaptaciones en cada ciclo de desarrollo.

Aprendizaje Continuo. El desarrollo en iteraciones permite aplicar conocimientos, recibir retroalimentación y refinar el producto de forma progresiva.

Entrega de Valor Temprana y Continua. Cada iteración produce un incremento funcional del software. Permite tener versiones funcionales del sistema en etapas tempranas para pruebas y validación.

Mitigación de Riesgos. Al dividir el proyecto en partes más pequeñas, los riesgos asociados a cada funcionalidad pueden identificarse y mitigarse de manera más oportuna.

Aunque no se implementó un marco ágil formal como Scrum (que está más orientado a equipos), se adoptan principios como el desarrollo iterativo, la entrega incremental y la adaptabilidad.

Fases de la Metodología

El desarrollo del proyecto se estructuró en cuatro fases principales, cada una contribuyendo al cumplimiento de los objetivos específicos definidos y alineada con el cronograma de trabajo.

Fase 1: Análisis y Especificación de Requisitos

Esta fase inicial se centró en la comprensión de los requisitos del proyecto y las necesidades del sitio recursos-biblicos.com. Se definieron los objetivos, se delimitó el alcance de las funcionalidades y se seleccionaron las tecnologías clave. Finalmente, se elaboró el cronograma de trabajo para guiar el desarrollo.

Esta fase sentó las bases teóricas y de planificación para todos los objetivos específicos.

Fase 2: Diseño de la Arquitectura y la Interfaz

Se diseñó la arquitectura del sistema, el esquema de la base de datos en MySQL. Se configuró el proyecto en la Consola de Firebase, obteniendo las credenciales de cliente y servidor. Se preparó el entorno de desarrollo local (servidor PHP, MySQL) y se implementó la base de datos. Se desarrolló la lógica inicial de backend para la suscripción de usuarios y la comunicación con Firebase.

Fase 3: Desarrollo de los Componentes del Sistema

Fue la fase principal de codificación. Se construyó el panel de administración (dashboard) con su interfaz y sistema de autenticación. Se implementó la función principal de envío de notificaciones, optimizándola para manejar lotes mediante cURL Multi y para detectar y gestionar tokens inválidos. Se crearon los endpoints AJAX (send.php, get_stats.php) para la comunicación entre el frontend y el backend.

Fase 4: Validación y Pruebas Funcionales

Se realizaron pruebas para asegurar el correcto funcionamiento de todos los componentes de manera integrada. Se verificó el flujo completo desde la suscripción del usuario en el frontend hasta el envío y la recepción de la notificación, incluyendo el seguimiento de clics y la visualización de estadísticas. Se corrigieron los errores detectados.

Encuesta de Viabilidad del Proyecto

Diseño de la Encuesta

Para evaluar la viabilidad y aceptación del sistema de notificaciones push propuesto para recursos-biblicos.com, se diseñó una encuesta dirigida a los usuarios potenciales del sitio. El objetivo de esta encuesta sería medir el interés en recibir notificaciones, el tipo de contenido preferido, y las percepciones sobre la utilidad de dicho sistema.

El cuestionario completo de la encuesta se presenta en el Apéndice A.

Resumen y Análisis de Resultados de la Encuesta

Para determinar la viabilidad y la aceptación del sistema de notificaciones push, se aplicó una encuesta a una muestra de 28 usuarios de la comunidad de recursos-biblicos.com. Los resultados obtenidos, que se analizan a continuación, demuestran una clara pertinencia y un interés mayoritario en la implementación de esta nueva herramienta de comunicación.

Frecuencia de Visita

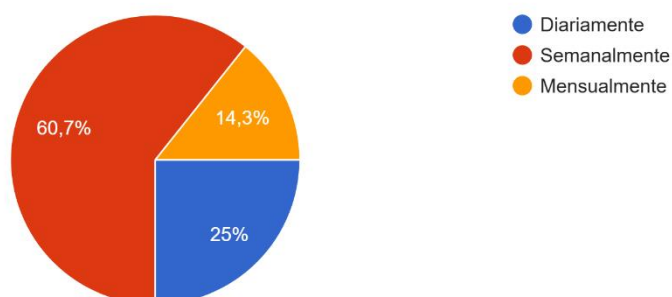
El análisis de la frecuencia con la que los encuestados visitan el sitio web revela una audiencia comprometida. Un 57.1% (16 de 28) visita el sitio semanalmente, mientras que un 28.6% (8) lo hace diariamente. Un 14.3% (4) indica una frecuencia mensual. Esta alta recurrencia de visitas sugiere la existencia de una base de usuarios leal que podría beneficiarse de un canal de comunicación más directo para las novedades.

Figura 6

Gráfico de Frecuencia de Visita de los Encuestados

1. ¿Con qué frecuencia visita usted el sitio web recursos-biblicos.com?

28 respuestas



Nota. El gráfico de barras muestra la distribución porcentual de la frecuencia con la que la población encuestada declara visitar el sitio web recursos-biblicos.com.

Interés en Recibir Notificaciones Push

Se observa una clara mayoría a favor de recibir notificaciones. El 67.9% de los encuestados manifestó una actitud positiva, distribuyéndose entre un 39.3% (11) que lo considera "definitivamente" útil y un 28.6% (8) que opina que "podría ser útil". En contraparte, un 28.6% (8) prefiere visitar el sitio directamente, y solo un participante (3.6%) indicó no tener interés.

Figura 7

Gráfico de Interés en Recibir Notificaciones

2. ¿Le gustaría ser informado/a directamente en su navegador sobre nuevo contenido (artículos, estudios, lecciones) o anuncios importantes del sitio, incluso si no tiene la página abierta?

28 respuestas



Nota. Distribución de las respuestas de los usuarios sobre su nivel de interés para recibir notificaciones push directamente en sus navegadores.

Contenido de Mayor Interés para Notificaciones

Al preguntar sobre el tipo de contenido que les resultaría más valioso recibir, las

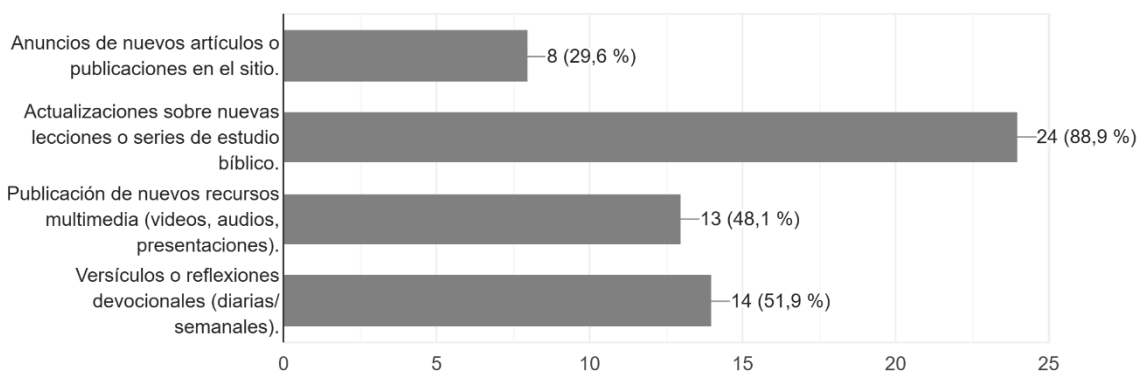
"Actualizaciones sobre nuevas lecciones o series de estudio bíblico" fueron la opción más popular, seleccionada por 20 de los 28 encuestados. Esta información es crucial para definir una estrategia de contenido efectiva para las notificaciones.

Figura 8

Gráfico de Preferencias de Contenido para Notificaciones.

3. ¿Qué tipo de información le resultaría más valiosa recibir a través de notificaciones push de recursos-biblicos.com? (Puede marcar varias opciones)

27 respuestas



Nota. Visualización de los tipos de contenido que los usuarios encuestados consideran más valiosos para ser notificados, permitiendo la selección de múltiples opciones.

Percepción sobre la Mejora del Compromiso

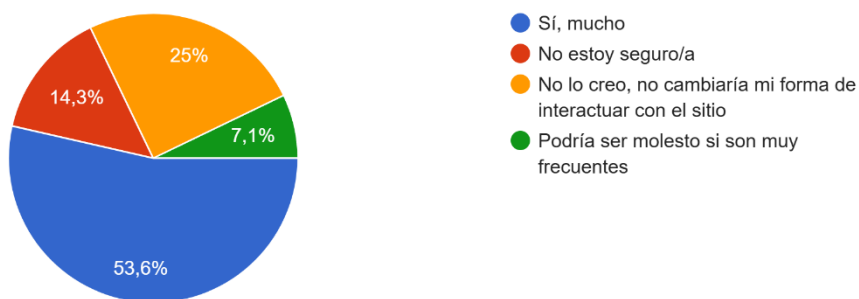
El 64.3% (18 de 28) de los participantes considera que recibir notificaciones push mejoraría "mucho" su conexión y compromiso con el sitio. Un 14.3% (4) no está seguro, mientras que el 14.3% (4) no cree que cambiaría su forma de interactuar. Un 7.1% (2) expresó preocupación de que podría ser molesto. Este resultado positivo refuerza la justificación del proyecto como una herramienta para fortalecer la comunidad y el engagement.

Figura 9

Gráfico de Percepción sobre la Mejora del Compromiso.

4. ¿Considera que recibir notificaciones push mejoraría su conexión y compromiso con el contenido y la comunidad de recursos-biblicos.com?

28 respuestas



Nota. El gráfico representa la percepción de los usuarios sobre si las notificaciones push mejorarían su conexión y compromiso con el contenido del sitio web.

Dispositivo Preferido para la Recepción

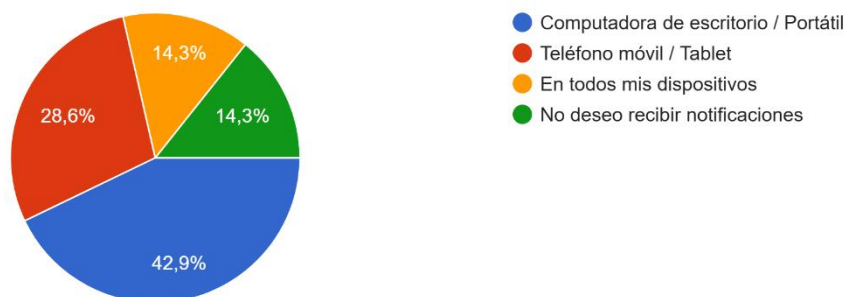
Se observa una clara preferencia por los dispositivos de escritorio, con un **46.4%** (13) eligiendo "Computadora de escritorio / Portátil". Le siguen los dispositivos móviles con un **25%** (7), y un significativo **14.3%** (4) que desearía recibir las notificaciones en "todos sus dispositivos".

Figura 10

Gráfico de Percepción sobre la Mejora del Compromiso

5. ¿En qué tipo de dispositivo preferiría recibir estas notificaciones?

28 respuestas



Nota. Muestra la distribución de los dispositivos (escritorio, móvil, etc.) en los que los usuarios encuestados preferirían recibir las notificaciones push.

Conclusión de la Encuesta de Viabilidad

Los resultados de la encuesta validan de manera contundente la viabilidad y necesidad del proyecto. Existe una audiencia activa y comprometida que, en su mayoría, está dispuesta y considera útil recibir notificaciones push para mantenerse informada sobre el contenido de recursos-biblicos.com. Los datos sobre el tipo de contenido preferido y la percepción positiva sobre la mejora del engagement proporcionan una base sólida para justificar la implementación y el futuro uso de este sistema.

Planificación (Cronograma de Trabajo)

El desarrollo del Sistema de Notificaciones Push Web se planificó para ser ejecutado a lo largo de un periodo total de cinco meses. A continuación, se describe la distribución de tareas principales en un formato conceptual de Diagrama de GANTT.

Figura 11

Diagrama de GANTT del Proyecto

	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5
Fase 1: Análisis y Especificación	Definición de Requisitos y Planificación Inicial				
Fase 2: Diseño		Diseño de Arquitectura y Entorno de Desarrollo			
Fase 3: Desarrollo		Desarrollo Plataforma PHP y Frontend			
Fase 4: Validación				Evaluación y Pruebas de Funcionalidad	

Nota. Representación visual del cronograma de trabajo, mostrando las fases principales del proyecto y su duración estimada a lo largo de cinco meses.

Explicación del Cronograma

El cronograma de trabajo se estructuró en cinco fases principales, distribuidas a lo largo de los cinco meses asignados para el proyecto:

Fase 1: Análisis y Especificación (Mes 1)

Se definieron los requisitos, objetivos y alcance del proyecto. Se investigaron y seleccionaron las tecnologías (PHP, MySQL, FCM, etc.) y se creó un plan de trabajo inicial.

Fase 2: Diseño (Mes 2)

Se centró en el diseño de la arquitectura, la base de datos y la interfaz de usuario. Se configuró el proyecto en Firebase, obteniendo las credenciales necesarias. Se preparó el entorno de desarrollo local (servidor PHP, MySQL) y el archivo .env para la configuración.

Fase 3: Desarrollo (Mes 2 y 3)

Se codificaron los componentes principales: lógica de suscripción, obtención de token de acceso FCM, función de envío de notificaciones (con cURL Multi), y el dashboard (HTML, autenticación, formulario de envío, endpoints AJAX y visualización de estadísticas con Chart.js).

Fase 4: Validación (Mes 3)

Se realizaron pruebas para verificar el correcto funcionamiento de todos los flujos del sistema (suscripción, envío, recepción, clics, estadísticas) y se corrigieron los errores detectados.

Este cronograma facilitó un desarrollo progresivo y organizado, asegurando la entrega de un producto funcional y documentado.

Presupuesto

Tabla 1

Desglose de Costos y Recursos Estimados del Proyecto

Categoría	Detalle	Costo Estimado (COP)
Recursos de Software		
	PHP (Lenguaje de Programación)	\$0.00
	MySQL / MariaDB (Sistema Gestor de Base de Datos)	\$0.00
	Servidor Web (Apache, Nginx, u OpenLiteSpeed)	\$0.00
	Firebase Cloud Messaging (FCM)	\$0.00
	Composer (Gestor de Dependencias PHP)	\$0.00
	Librería vlucas/phpdotenv	\$0.00
	Bootstrap 4 (Framework CSS)	\$0.00
	Chart.js (Librería de Gráficos)	\$0.00
	Sistema Operativo (para desarrollo y servidor)	\$0.00
	Entorno de Desarrollo Integrado (IDE) (Visual Studio Code)	\$0.00
Recursos de Hardware		
	Computadora de Desarrollo	\$0.00
	Desarrollo Local: Utilización de XAMPP, WAMP, MAMP o similar	\$0.00
	Hosting	\$0.00
Dominio Web		
	Dominio recursos-biblicos.com	\$0.00
Recursos Humanos		
	Mi tiempo invertido	\$25.000.000.00

Costo Total Monetario Estimado del Proyecto	\$25.000.000.00
--	-----------------

Nota. La tabla destaca la optimización de recursos, con un costo monetario directo de \$0.00. El total de \$25.000.000.00 COP corresponde a un valor de oportunidad estimado por desarrollo y no a un desembolso financiero.

Costo Total Monetario Estimado del Proyecto

Considerando la utilización predominante de software de código abierto, servicios con planes gratuitos y la infraestructura personal o preexistente, el costo monetario directo para la realización y desarrollo de este proyecto es de \$25.000.000.00 COP (valor de oportunidad).

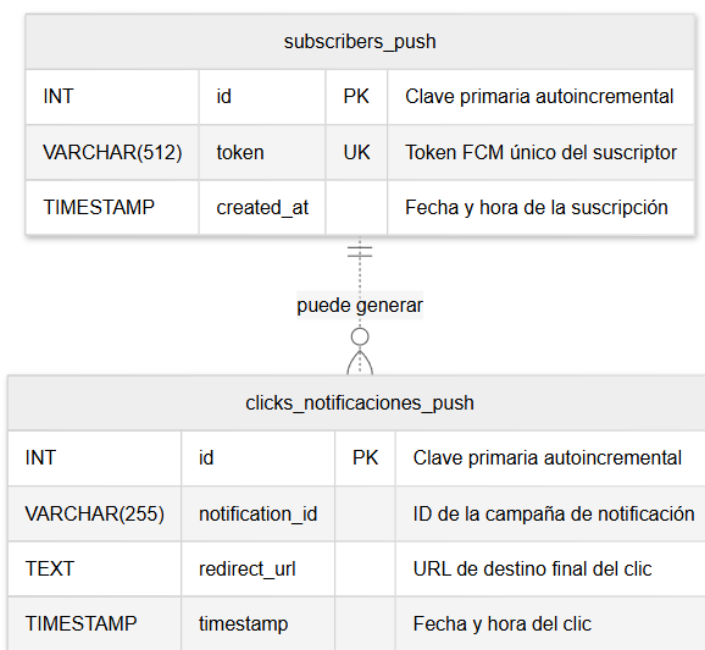
Desarrollo y Funcionalidad del Sistema

Base de Datos

El sistema de notificaciones push web se apoya en una base de datos relacional MySQL/MariaDB para almacenar la información crítica relacionada con los suscriptores, los clics en las notificaciones y la gestión de las tareas de envío. A continuación, se detalla el esquema y la descripción de las tablas implementadas.

Figura 12

Diagrama MER de la Base de Datos del Sistema



Nota. El diagrama ilustra las entidades de datos principales, subscribers_push y clicks_notificaciones_push, con sus respectivos atributos y claves.

Esquema de la Base de Datos

Se utilizan dos tablas principales: `subscribers_push` y `clicks_notificaciones_push`:

subscribers_push: Almacena los tokens de los dispositivos suscritos.

SQL

```
CREATE TABLE `subscribers_push` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `token` VARCHAR(512) COLLATE utf8mb4_unicode_ci NOT NULL UNIQUE,
  `created_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

clicks_notificaciones_push: Registra los clics realizados en las notificaciones.

SQL

```
CREATE TABLE `clicks_notificaciones_push` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `notification_id` VARCHAR(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `redirect_url` TEXT COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `timestamp` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

ALTER TABLE `clicks_notificaciones_push` ADD INDEX `idx_notif_id_ts` (`notification_id`,
`timestamp`);
```

Descripción de Tablas y Campos

Subscribers_push

id: Clave primaria autoincremental.

token: Almacena el token de registro FCM único para cada dispositivo/navegador suscrito. Es la pieza clave para dirigir la notificación.

created_at: Fecha y hora de la suscripción.

Clicks_notificaciones_push

id: Clave primaria autoincremental.

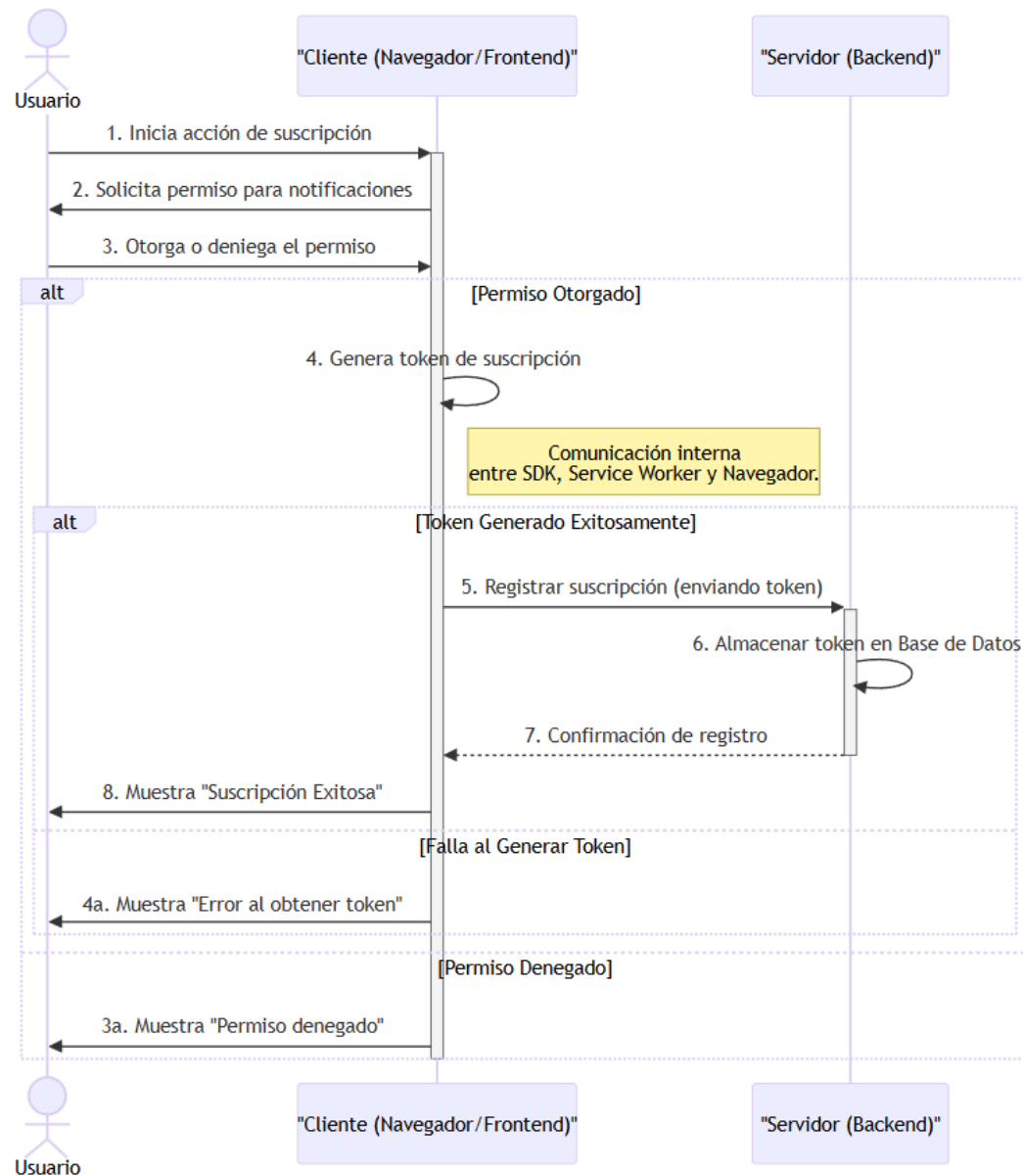
notification_id: Identificador de la campaña de notificación (corresponde a generated_notification_id de notification_jobs) a la que pertenece el clic. Permite agrupar clics.

`redirect_url`: La URL de destino final a la que el usuario fue redirigido.

`timestamp`: Fecha y hora en que se registró el clic.

Figura 13

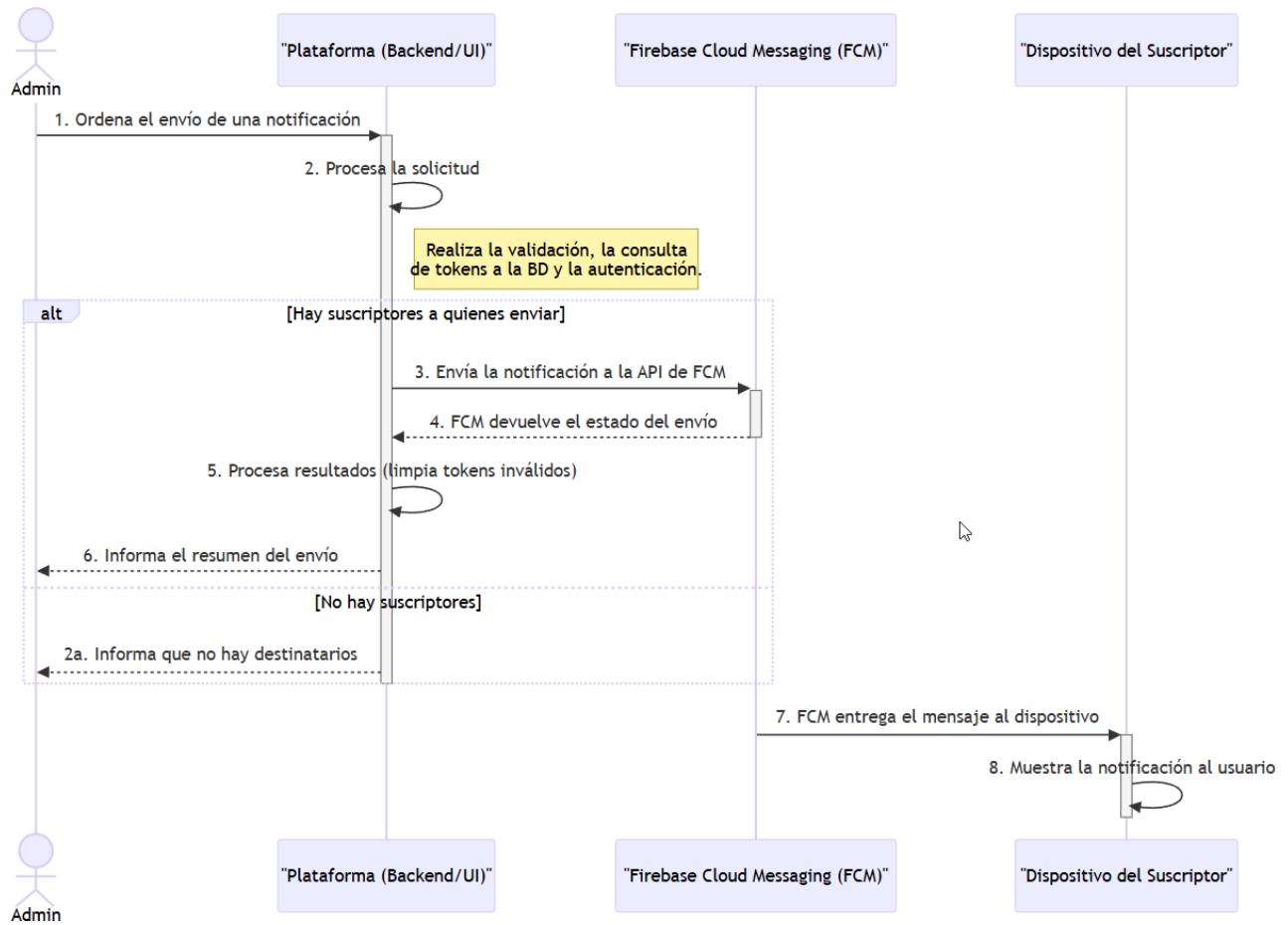
Diagrama de Secuencia del Proceso de Suscripción de Usuario a Notificaciones Push



Nota. El diagrama ilustra el proceso de suscripción a notificaciones para el registro de un usuario a notificaciones web. El participante "Cliente" agrupa los componentes del navegador y la lógica del frontend, mientras que "Servidor" representa el backend y la base de datos.

Figura 14

Diagrama de Secuencia del Proceso de Envío Directo de Notificaciones Push



Nota. El diagrama modela el proceso de envío de notificaciones iniciado por el administrador.

Ilustra la interacción entre la plataforma interna, FCM y el dispositivo final del usuario.

Programación más Representativa

A continuación, se describe la lógica y funcionalidad más importantes del sistema, que en consideración propia es el script “send.php”, y en particular la función “send_fcm_notification” que contiene, la cual representa el núcleo funcional y la lógica más compleja del sistema. Este componente es definitivamente el corazón de todo el proceso de notificaciones.

El propósito de la función “send_fcm_notification” es orquestar todo el proceso de envío masivo de notificaciones push, desde la obtención de los destinatarios hasta la comunicación con Firebase y el manejo de los resultados. A continuación, su funcionamiento y explicación de sus secciones principales.

Obtención y Preparación de Datos

Primero, lee las credenciales de la base de datos desde las variables de entorno (\$_ENV). Luego, se conecta a la base de datos MySQL para obtener la lista completa de tokens de suscriptores activos. Luego carga los token en un array en memoria (\$registration_ids) y cierra la conexión para liberar la memoria.

Figura 15

Recuperación de Tokens de Suscriptores

```
// --- Recuperación de Tokens de Suscriptores ---
// Consulta para obtener todos los tokens de la tabla de suscriptores.
$sql = "SELECT token FROM subscribers_push";
$result = $conn->query($sql);
$registration_ids = [];
if ($result === FALSE) {
    error_log($log_prefix . "DB Fetch Error: " . $conn->error, 3,
    $php_error_log);
    $conn->close();
    return ['success' => false, 'message' => 'Error Interno (DB Fetch).',
    'sent' => 0, 'failed' => 0, 'removed' => 0];
} else {
    // Almacenar todos los tokens recuperados en un array.
    while ($row = $result->fetch_assoc()) {
        $registration_ids[] = $row['token'];
    }
    $result->free(); // Liberar memoria del resultado de la consulta.
}
$conn->close();
```

Nota. Consulta SQL para obtener la lista de tokens almacenados en la base de datos.

Autenticación con Firebase (FCM)

Para comunicarse de forma segura con la API HTTP v1 de Firebase, el servidor debe autenticarse usando el protocolo OAuth 2.0, mediante una función auxiliar, “get_bearer_token()”, que utiliza la clave de servicio JSON del proyecto para generar un token de acceso de corta duración. Este token se incluirá en la cabecera de todas las peticiones a FCM.

Figura 16

Obtención del Token de Acceso a FCM

```
// --- Obtención del Token de Acceso a FCM ---
// Llama a la función (definida en get_bearer_token.php) para obtener el
token OAuth 2.0.
try {
    $accessToken = get_bearer_token();
    // Validar que el token no esté vacío.
    if (empty($accessToken)) {
        throw new Exception("No se pudo obtener el Bearer Token (vacío).");
    }
} catch (\Throwable $e) {
    // Error al obtener el token de autenticación, no se puede continuar.
    error_log($log_prefix . "Bearer Token Error: " . $e->getMessage(), 3,
    $php_error_log);
    // Retorna fallo, asumiendo que todas las notificaciones fallarían.
    return ['success' => false, 'message' => 'Error Interno (Auth Token).',
    'sent' => 0, 'failed' => $total_tokens_found, 'removed' => 0];
}
```

Nota. Llamada a la función `get_bearer_token` que obtiene el token OAuth 2.0 para autenticar el acceso a Firebase Cloud Manager.

Implementación de Envío por Lotes (cURL Multi)

Esta es la técnica más importante para la eficiencia y escalabilidad del sistema. Enviar miles de notificaciones de forma secuencial (una tras otra) sería muy lento y excedería los límites de tiempo de ejecución del servidor. Para resolver esto, se utiliza la extensión cURL Multi de PHP, que permite procesar múltiples peticiones HTTP de forma concurrente.

La lógica es la siguiente:

Se divide el gran array de tokens de suscriptores en lotes más pequeños (200 tokens por lote).

Para cada lote, se prepara una petición cURL individual por cada token con su payload correspondiente.

En lugar de ejecutar cada petición, se añaden todas a un "manejador múltiple" (`curl_multi_add_handle`).

Finalmente, la función `curl_multi_exec` ejecuta todas las peticiones del lote en paralelo, reduciendo drásticamente el tiempo total de espera de la red.

Figura 17

Inicialización del Proceso de Envío por Lotes

```
// --- Inicialización del Proceso de Envío por Lotes ---
$invalid_tokens =
$total_messages_sent = 0;
$total_failures = 0
$max_messages_per_batch = 200; // Número de notificaciones a enviar en cada
lote concurrente.
$delay_between_batches = 100000;
$mh = curl_multi_init(); // Inicializar manejador cURL Multi para peticiones
paralelas.
curl_multi_setopt($mh, CURLOPTPIPELINING, CURLOPTPIPE_MULTIPLEX);
$processed_tokens = [];

// ...

// --- Bucle Principal de Envío por Lotes ---
for ($i = 0; $i < $total_tokens_found; $i += $max_messages_per_batch) {
    // Obtener el lote actual de tokens.
    $batch_tokens = array_slice($registration_ids, $i,
$max_messages_per_batch);
    $current_batch_handles = [];
```

Nota. Lógica que inicializa el bucle del proceso de envío por lotes mediante CURL Multi y configura cada lote a 200 tokens.

Figura 18

Preparación, Procesamiento y Ejecución del Lote

```

// --- Preparación de Peticiones cURL para el Lote ---
foreach ($batch_tokens as $token) {
    // Construcción del Payload JSON para la API FCM v1. (es muy importante
    la estructura)
    $payload = [
        'message' => [
            'token' => $token, // Token del destinatario actual.
            'data' => [ // Payload personalizado para el cliente.
                'title' => $current_title,
                'body' => $current_message,
                'image' => $image_url,
                'click_action' => $click_action_url
            ]
        ]
    ];
    // Convertir el payload a formato JSON.
    $json_data = json_encode($payload);
    // --- Configuración de la Petición cURL Individual ---
    $ch = curl_init(); // Inicializar un handle cURL.
    // ...
    curl_multi_add_handle($mh, $ch);
    $current_batch_handles[$token] = $ch;
}
// --- Ejecución y Procesamiento del Lote Actual ---
if (!empty($current_batch_handles)) {
    $running = null;
    do {
        curl_multi_exec($mh, $running); // Ejecuta/continúa las
transferencias.
        curl_multi_select($mh); // Espera actividad en alguno de los
handles.
    } while ($running > 0);
    // --- Procesamiento de Respuestas del Lote ---
    foreach ($current_batch_handles as $token => $ch) {
        // ... aquí se procesarían las respuestas para este lote ...
        // ...
    }
} // --- Fin Bucle Principal de Lotes ---
// Cerrar el manejador cURL Multi.
curl_multi_close($mh);

```

Nota. Construcción de la estructura del Payload JSON para la API FCM v1, configuración de la petición cURL individual y procesamiento del lote actual de tokens. Finaliza el bucle de envío.

Procesamiento de Respuestas y Gestión de Errores

Una vez que un lote de notificaciones ha sido enviado, el sistema procesa las respuestas de FCM para cada token individualmente

Se verifica el código de estado HTTP de la respuesta. Un código en el rango 2xx indica que FCM aceptó el mensaje para su entrega.

Si el código de estado indica un error (ej. 4xx o 5xx), se decodifica el cuerpo de la respuesta JSON de FCM para identificar el código de error específico.

Se contabilizan los envíos exitosos (\$total_messages_sent) y los fallidos (\$total_failures) para generar un resumen final.

Figura 19

Procesamiento de Respuestas del Lote

```
// --- Procesamiento de Respuestas del Lote ---
// Iterar sobre los handles que se ejecutaron en este lote.
foreach ($current_batch_handles as $token => $ch) {
// Obtener el contenido de la respuesta y el código de estado HTTP.
$response_content = curl_multi_getcontent($ch);
$http_status = curl_getinfo($ch, CURLINFO_HTTP_CODE);

// Verificar si hubo un error en la transferencia CURL misma.
if ($response_content === false) {
    // ... Lógica para manejar el error ...
}
}
}

curl_multi_remove_handle($mh, $ch);
curl_close($ch);
}
```

Nota. Se procesan las respuestas del lote enviado obteniendo el contenido y el código de la respuesta http.

Limpieza Automática de Tokens Inválidos

Una parte crucial del mantenimiento del sistema es la depuración de la lista de suscriptores.

Cuando FCM devuelve un código de error como UNREGISTERED o INVALID_ARGUMENT, significa que el token ya no es válido (el usuario desinstaló la app, borró los datos del navegador, etc.). El script identifica estos tokens durante el procesamiento de respuestas y los almacena en un array temporal (\$invalid_tokens).

Al finalizar todo el proceso de envío, se ejecuta una única consulta DELETE a la base de datos para eliminar todos los tokens inválidos recolectados.

Figura 20

Eliminación de Tokens Inválidos de la Base de Datos

```
// --- Eliminación de Tokens Inválidos de la Base de Datos ---
$removed_count = 0; // Contador de tokens eliminados.
if (!empty($invalid_tokens)) {
    // Asegurar que solo se intente eliminar cada token inválido una vez.
    $unique_invalid_tokens = array_unique($invalid_tokens);

    // ...

    $placeholders = implode(',', array_fill(0,
count($unique_invalid_tokens), '?'));
    // ...

    $delete_sql = "DELETE FROM subscribers_push WHERE token IN
($placeholders)";

    $stmt = $conn_delete->prepare($delete_sql);
    if ($stmt) {
        // Vincular los tokens inválidos a los placeholders.
        $stmt->bind_param($types, ...$unique_invalid_tokens);
        // Ejecutar la consulta DELETE.
        if ($stmt->execute()) {
            $removed_count = $stmt->affected_rows;
            // ...
        }
    }
}
```

Nota. Todos los tokens de los que se ha obtenido una respuesta invalida de FCM son identificados, contados y eliminados de la base de datos.

URL del Proyecto Web Funcional y Árbol de Directorios

El panel de administración (dashboard) del sistema de notificaciones push implementado esta accesible, a través de la siguiente URL: https://www.recursos-biblicos.com/wp_push3/public/dashboard.php

Árbol de Directorios del Proyecto

La estructura de archivos y carpetas del proyecto se ha organizado para promover la separación de responsabilidades, la seguridad y la mantenibilidad:

```

wp_push/
├── public/          # Directorio Raíz Web (DocumentRoot) - Archivos públicos
|   ├── dashboard.php # Interfaz principal del dashboard (UI)
|   ├── get_stats.php # Endpoint AJAX para obtener estadísticas (JSON)
|   ├── send.php      # Endpoint AJAX para procesar envío de notificaciones (JSON)
|   ├── save-token.php # Endpoint para guardar tokens de suscriptores
|   ├── registrar-clic.php # Endpoint para registrar clics en notificaciones
|   |
|   └── firebase-messaging-sw.js # Service Worker OBLIGATORIO para recibir push (debe
estar instalado en la raíz del sitio web)
|
├── src/            # Código PHP interno
|   ├── get_bearer_token.php # Script/Función para obtener token OAuth para FCM API v1
|   └── firebase_config.php # Configuración de Firebase desde el lado del servidor
|
└── storage/        # Almacenamiento no público.

```

```

|   └─ keys/
|   |   └─ *.json      # Archivo JSON de Clave de Servicio de Firebase Admin
|   └─ logs/
|       └─ php_error.log # Log de errores generales de PHP
|       └─ send_log.log  # Log específico del proceso de envío
|       └─ logfile_fecha*.log # Log detallado de cada proceso de envío
|
|
|
└─ .env          # Variables de entorno (Credenciales DB, Firebase, Auth, etc.) -
└─ composer.json # Definición de dependencias PHP
└─ composer.lock # Versiones fijas de dependencias
└─ vendor/      # Carpeta de dependencias instaladas

```

Explicación del Árbol de Directorios

Wp_push/ (Directorio Raíz del Proyecto)

Es la carpeta principal que contiene todos los archivos y subdirectorios del proyecto.

Public/

Este es el directorio raíz web. Es la carpeta accesible directamente a través de un navegador web.

`dashboard.php`: Interfaz principal del panel de administración donde el usuario gestiona las notificaciones y ve estadísticas.

`get_stats.php`: Endpoint AJAX que el dashboard utiliza para solicitar y recibir datos de estadísticas (total de suscriptores, clics) en formato JSON.

send.php: Endpoint AJAX que recibe los datos del formulario del dashboard para procesar el envío de notificaciones. Devuelve una respuesta JSON.

save-token.php: Endpoint que recibe y guarda los tokens de registro FCM de los nuevos suscriptores en la base de datos.

registrar-clic.php: Endpoint que registra los clics realizados en las notificaciones y luego redirige al usuario a la URL de destino final.

firebase-messaging-sw.js: El Service Worker. Es un script JavaScript crucial que se ejecuta en segundo plano en el navegador del usuario y es responsable de recibir las notificaciones push de Firebase, incluso cuando el sitio web no está abierto. También maneja la visualización de la notificación.

Src/

Contiene el código PHP interno de la aplicación.

get_bearer_token.php: Script o función que se encarga de generar el token de acceso OAuth 2.0 necesario para autenticar las peticiones del servidor a la API HTTP v1 de Firebase Cloud Messaging.

firebase_config.php: Este archivo se usa para servir la configuración del *cliente* de Firebase al JavaScript (leyéndola desde .env).

Storage/

Directorio para almacenamiento no público.

storage/keys/: Carpeta para guardar archivos de claves sensibles.

republicos-push-firebase-clave-servicio.json: Aquí se guarda el archivo JSON de la Clave de Servicio de Firebase Admin.

storage/logs/: Carpeta para almacenar los archivos de log generados por la aplicación.

Vital para la depuración y el seguimiento de errores.

Archivos en la Raíz (wp_push/)

.env: Archivo de variables de entorno.

composer.json: Define las dependencias PHP del proyecto que son gestionadas por Composer (vlucas/phpdotenv).

vendor/: Carpeta creada por Composer.

Conclusiones

Al culminar el desarrollo e implementación del Sistema de Notificaciones Push Web para recursos-biblicos.com, se han alcanzado los objetivos propuestos y se derivan las siguientes conclusiones:

Sistema Funcional Implementado: Se ha logrado desarrollar un sistema completo y funcional que permite a los administradores de recursos-biblicos.com enviar notificaciones push web a sus suscriptores. El panel de administración facilita la creación de mensajes, y el sistema gestiona el envío, el seguimiento básico de clics y la visualización de estadísticas.

Cumplimiento de Objetivos: Se implementó el dashboard con autenticación, suscripción y almacenamiento de tokens, lógica de backend para FCM con envío en lotes (cURL Multi), seguimiento de clics, visualización de estadísticas (limitadas a N resultados), limpieza de tokens inválidos y configuración vía ``.env``.

Aplicación Práctica de Conocimientos: El proyecto ha servido como un excelente ejercicio para aplicar y consolidar los conocimientos adquiridos en la carrera de Tecnología en Desarrollo de Software, abarcando desarrollo backend con PHP, gestión de bases de datos MySQL, interacción con APIs externas (FCM), desarrollo frontend básico con HTML/CSS/JavaScript, y la implementación de tareas asíncronas.

Desafíos y Aprendizajes: Entre los desafíos afrontados se encontraron la correcta configuración del Service Worker para diferentes escenarios, el manejo eficiente de errores y respuestas de la API de FCM, y la implementación robusta del procesamiento en segundo plano. Estos desafíos proporcionaron valiosas oportunidades de aprendizaje.

Potencial de Crecimiento: El sistema actual, aunque funcional, posee un considerable potencial de mejora y expansión, como se detalla en las "Posibles Mejoras Futuras"

(programación de notificaciones, segmentación, estadísticas avanzadas, autenticación más robusta), lo que demuestra la escalabilidad de la solución base.

Se reconoce que el envío directo dentro de una petición HTTP, aunque optimizado con lotes, puede enfrentar timeouts del servidor web si el número de suscriptores crece a cientos de miles. La implementación de un sistema de encolado y procesamiento en segundo plano (como se describe en "Mejoras Futuras") sería necesaria para una escalabilidad mayor.

En definitiva, el proyecto ha culminado en una herramienta útil y funcional, demostrando la capacidad de integrar diversas tecnologías para resolver una necesidad comunicacional específica, al tiempo que ha constituido una valiosa experiencia formativa.

Referencias Bibliográficas

- Bootstrap. (s.f.). Bootstrap Documentation (v4.x). Recuperado el 31 de enero de 2025, de <https://getbootstrap.com/docs/4.5/getting-started/introduction/>
- Chart.js. (s.f.). Chart.js Documentation. Recuperado el 31 de enero de 2025, de <https://www.chartjs.org/docs/latest/>
- Firebase. (s.f.-a). Firebase Cloud Messaging. Recuperado el 31 de enero de 2025, de <https://firebase.google.com/docs/cloud-messaging>
- Firebase. (s.f.-b). Send messages with the HTTP v1 API. Recuperado el 31 de enero de 2025, de <https://firebase.google.com/docs/cloud-messaging/migrate-v1>
- Firebase. (s.f.-c). Set up a Firebase Cloud Messaging client app on JavaScript. Recuperado el 31 de enero de 2025, de <https://firebase.google.com/docs/cloud-messaging/js/client>
- Graham, V. (s.f.). PHP DotEnv (vlucas/phpdotenv). GitHub. Recuperado el 31 de enero de 2025, de <https://github.com/vlucas/phpdotenv>
- MariaDB. (s.f.). MariaDB Server Documentation. Recuperado el 31 de enero de 2025, de <https://mariadb.com/kb/en/documentation/>
- Mozilla Developer Network. (s.f.-a). Notifications API. Recuperado el 31 de enero de 2025, de https://developer.mozilla.org/en-US/docs/Web/API/Notifications_API
- Mozilla Developer Network. (s.f.-b). Push API. Recuperado el 31 de enero de 2025, de https://developer.mozilla.org/en-US/docs/Web/API/Push_API
- Mozilla Developer Network. (s.f.-c). Service Worker API. Recuperado el 31 de enero de 2025, de https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API
- MySQL. (s.f.). MySQL Documentation. Recuperado el 31 de enero de 2025, de <https://dev.mysql.com/doc/>

PHP. (s.f.-a). cURL. Recuperado el 31 de enero de 2025, de

<https://www.php.net/manual/en/book.curl.php>

PHP. (s.f.-b). MySQLi. Recuperado el 31 de enero de 2025, de

<https://www.php.net/manual/en/book.mysqli.php>

PHP. (s.f.-c). PHP Manual. Recuperado el 31 de enero de 2025, de

<https://www.php.net/manual/es/>

Recursos Bíblicos. (2025). [Organigrama del ministerio] [Gráfico]. Recursos Bíblicos.

Recuperado el 2 de julio de 2025, de <https://www.recursos-biblicos.com/acerca-de>

Apéndices

Apéndice A

Encuesta de Viabilidad del Proyecto (Cuestionario Completo)

Cuestionario de Encuesta: Viabilidad del Sistema de Notificaciones Push para recursos-biblicos.com

Introducción: Apreciado usuario de recursos-biblicos.com, esta encuesta tiene como objetivo recoger su opinión sobre la posible implementación de un sistema de notificaciones push para el sitio web. Su participación es voluntaria y sus respuestas serán de gran valor.

¿Con qué frecuencia visita usted el sitio web recursos-biblicos.com?

Diariamente

Semanalmente

Mensualmente

¿Le gustaría ser informado/a directamente en su navegador sobre nuevo contenido (artículos, estudios, lecciones) o anuncios importantes del sitio, incluso si no tiene la página abierta?

Sí, definitivamente

Sí, podría ser útil

No, prefiero visitar el sitio directamente para ver novedades

No, no me interesan las notificaciones de ningún tipo

¿Qué tipo de información le resultaría más valiosa recibir a través de notificaciones push de recursos-biblicos.com? (Puede marcar varias opciones)

Anuncios de nuevos artículos o publicaciones en el sitio.

Actualizaciones sobre nuevas lecciones o series de estudio bíblico.

Publicación de nuevos recursos multimedia (videos, audios, presentaciones).

Versículos o reflexiones devocionales (diarias/semanales).

Otro (por favor especifique): _____

¿Considera que recibir notificaciones push mejoraría su conexión y compromiso con el contenido y la comunidad de recursos-biblicos.com?

Sí, mucho

No estoy seguro/a

No lo creo, no cambiaría mi forma de interactuar con el sitio

Podría ser molesto si son muy frecuentes

¿En qué tipo de dispositivo preferiría recibir estas notificaciones?

Computadora de escritorio / Portátil

Teléfono móvil

En todos mis dispositivos

No deseo recibir notificaciones

¡Gracias por su tiempo y colaboración!