Universidad Tecnológica del Norte de Guanajuato.

Arquitectura de Software.

Unidad I.

Fundamentos de Arquitectura de Software.

Alumnos:

Manzano Villafaña Gerardo

No. Control: **1222100474**

Mildred Mariana Banda López

No. Control: **1222100436**

Grupo: **GIDS6072**

Dolores Hidalgo C.I.N. Gto.

16 de septiembre, 2025

Índice

Manzano Villafaña Gerardo
Mildred

Introducción

En la actualidad, la educación enfrenta uno de los mayores retos de la era digital: garantizar que todos los estudiantes, independientemente de su contexto geográfico o socioeconómico, tengan acceso a recursos tecnológicos de calidad. La digitalización ha transformado de manera radical los métodos de enseñanza y aprendizaje, permitiendo la creación de entornos interactivos, colaborativos y globalizados. Sin embargo, esta transformación no se desarrolla de manera uniforme, lo que genera una profunda brecha digital que afecta directamente la igualdad de oportunidades educativas. En México y en gran parte de América Latina, esta brecha es especialmente visible en las comunidades rurales, donde las condiciones de conectividad y acceso a dispositivos siguen siendo limitadas y desiguales frente a los entornos urbanos.

La desigualdad en el acceso a la educación digital no solo refleja una carencia tecnológica, sino también una problemática estructural que impacta en el desarrollo social, cultural y económico de los pueblos marginados. De acuerdo con investigaciones recientes, más del 35% de los estudiantes en zonas rurales carecen de internet estable o de dispositivos adecuados para participar en actividades escolares. Esta cifra revela que la brecha digital no es simplemente un asunto de infraestructura, sino también un indicador de desigualdad social que condiciona el futuro de generaciones enteras. Mientras que en contextos urbanos los estudiantes pueden acceder a bibliotecas virtuales, clases en línea y plataformas interactivas, aquellos que viven en comunidades con baja conectividad dependen de métodos tradicionales, limitados y, en muchos casos, obsoletos.

La pandemia de COVID-19 puso en evidencia la urgencia de atender este problema. Durante el confinamiento, miles de estudiantes quedaron excluidos de la educación virtual, lo que incrementó el rezago académico y elevó las tasas de deserción escolar. Los efectos de esta exclusión fueron particularmente graves en los niveles medio superior y superior, donde la continuidad educativa depende en gran medida de las tecnologías digitales. Además, la carencia de capacitación en competencias digitales por parte de los docentes agravó el problema, ya que, aun con iniciativas gubernamentales para entregar infraestructura tecnológica, la falta de preparación en el uso de herramientas digitales limitó el impacto real de estas acciones.

La brecha digital también trasciende el ámbito educativo, influyendo en el desarrollo económico y social de las comunidades. Los jóvenes que no logran adquirir habilidades digitales enfrentan barreras para acceder a empleos de calidad, participar en programas de formación profesional o integrarse en entornos laborales que exigen competencias tecnológicas. De esta forma, la desigualdad tecnológica perpetúa un ciclo de pobreza y marginación, en el que las comunidades rurales se ven rezagadas frente a un mundo cada vez más interconectado y competitivo. En este contexto, la educación digital se convierte no solo en un derecho, sino en una condición indispensable para el desarrollo equitativo y sostenible.

Ante este panorama, surge la necesidad de diseñar soluciones innovadoras que respondan a los retos de la inclusión digital. No basta con proporcionar equipos o infraestructura; es necesario desarrollar plataformas tecnológicas accesibles, sostenibles y adaptadas a las condiciones de cada comunidad. Una solución viable es el diseño de una plataforma híbrida de educación digital, con capacidad para funcionar en línea y fuera de línea,

que integre contenidos pedagógicos, recursos interactivos y programas de capacitación docente. Esta propuesta se orienta a reducir la brecha digital al garantizar que estudiantes y profesores puedan continuar con sus actividades educativas aun en condiciones de baja o nula conectividad, ofreciendo además herramientas que fortalezcan la alfabetización digital en las comunidades.

Este documento tiene como propósito analizar la problemática de la desigualdad en el acceso a la educación digital en comunidades rurales, para luego plantear y justificar una propuesta tecnológica fundamentada en arquitecturas de software modernas. En particular, se evalúan dos alternativas: la primera, una arquitectura basada en SPA (Single Page Application) con API REST bajo un enfoque offline-first, orientada a la simplicidad, rapidez de implementación y compatibilidad con dispositivos de gama baja; y la segunda, una arquitectura híbrida que combina SPA con microservicios desplegados en contenedores Docker, lo cual ofrece escalabilidad, resiliencia y flexibilidad para enfrentar escenarios de expansión regional y demandas de alto volumen de usuarios.

Ambas propuestas se presentan como opciones complementarias, pensadas en fases evolutivas: una implementación inicial ligera y accesible para pilotos comunitarios, y una posterior expansión hacia un modelo más robusto y escalable a medida que la plataforma crezca en cobertura. La elección de estas arquitecturas responde a la necesidad de garantizar un sistema funcional en contextos de conectividad irregular, pero también preparado para sostener la demanda de múltiples comunidades rurales en México y Latinoamérica.

Finalmente, este trabajo no se limita a exponer soluciones técnicas, sino que plantea un compromiso social: utilizar la ingeniería de software como herramienta para cerrar brechas de desigualdad. La educación digital es un vehículo de transformación social, capaz de generar oportunidades, promover la equidad y abrir caminos hacia el desarrollo sostenible. Por ello, la propuesta aquí presentada no solo busca resolver un problema tecnológico, sino también contribuir al bienestar de las comunidades más vulnerables, asegurando que el acceso al conocimiento y a la tecnología no sea un privilegio de unos pocos, sino un derecho universal.

Problemática: Desigualdad en el Acceso a la Educación Digital.

En México y en varios países de Latinoamérica, la brecha digital representa un obstáculo critico para el desarrollo educativo y social de las comunidades rurales. Aunque la conectividad y el acceso a internet se han convertido en elementos esenciales para la educación, el trabajo y la vida cotidiana, millones de personas aún enfrentan dificultades para acceder a estas tecnologías de manera equitativa.

De acuerdo con estudios recientes, más del 35% de los estudiantes en zonas rurales carecen de acceso estable a internet y dispositivos adecuados para realizar actividades escolares. Esta situación se traduce en una desigualdad evidente: mientras que los alumnos en entornos urbanos tienen acceso a plataformas digitales, bibliotecas virtuales y clases en línea, los estudiantes en comunidades rurales dependen de métodos tradicionales, limitados y, en muchos casos, obsoletos.

La pandemia de COVID-19 acentuó esta problemática, al evidenciar la imposibilidad de miles de estudiantes de continuar su educación debido a la falta de recursos digitales. Esto no solo afectó su rendimiento académico, sino que también incrementó las tasas de deserción escolar, especialmente en niveles medio superior y superior. La falta de acceso a la educación digital donde existen indicios de pobreza y limita las oportunidades de desarrollo profesional y social de las nuevas generaciones. Otro aspecto crítico es la falta de capacitación digital entre los docentes de zonas rurales. Muchos maestros carecen de formación en el uso de herramientas tecnológicas, lo que agrava la brecha entre estudiantes urbanos y rurales. Aunque existan iniciativas gubernamentales para dotar de infraestructura tecnológica a las escuelas, estas resultan insuficientes sin la capacitación adecuada para su implementación.

Además, el acceso desigual a la tecnología no solo afecta al ámbito educativo, sino también al desarrollo económico de las comunidades. Jóvenes que no logran adquirir competencias digitales enfrentan dificultades para integrarse al mercado laboral actual, donde estas habilidades son cada vez más demandadas. Esto limita la competitividad de las comunidades rurales y las condena a permanecer rezagadas frente a un mundo globalizado y digitalizado.

La brecha digital también impacta en el ámbito social y cultural, ya que los estudiantes sin acceso a internet quedan excluidos de la información global, los movimientos sociales, la innovación y las oportunidades de participación ciudadana digital. Esto genera un círculo vicioso de aislamiento y marginación que afecta tanto al individuo como a la sociedad en su conjunto.

Ante esta situación, surge la necesidad de diseñar soluciones tecnológicas accesibles, sostenibles y adaptadas a las condiciones de las comunidades rurales. Una plataforma educativa híbrida —capaz de funcionar tanto en línea como fuera de línea— que integre contenidos pedagógicos de calidad, capacitación docente y recursos interactivos, podría ser una herramienta clave para reducir la brecha digital. Esta solución debe garantizar la inclusión de estudiantes y docentes, al tiempo que promueve la igualdad de oportunidades educativas y contribuye al desarrollo social y económico de las comunidades rurales.

La magnitud de la brecha digital en el ámbito educativo no puede comprenderse únicamente a partir de cifras generales o estudios nacionales; es necesario recoger evidencia directa en las comunidades afectadas. Por esta razón, se diseñaron y aplicaron encuestas exploratorias a estudiantes, docentes y padres de familia en contextos rurales. El objetivo principal fue identificar de manera puntual los obstáculos que enfrentan en relación con el acceso a internet, la disponibilidad de dispositivos y el nivel de capacitación tecnológica.

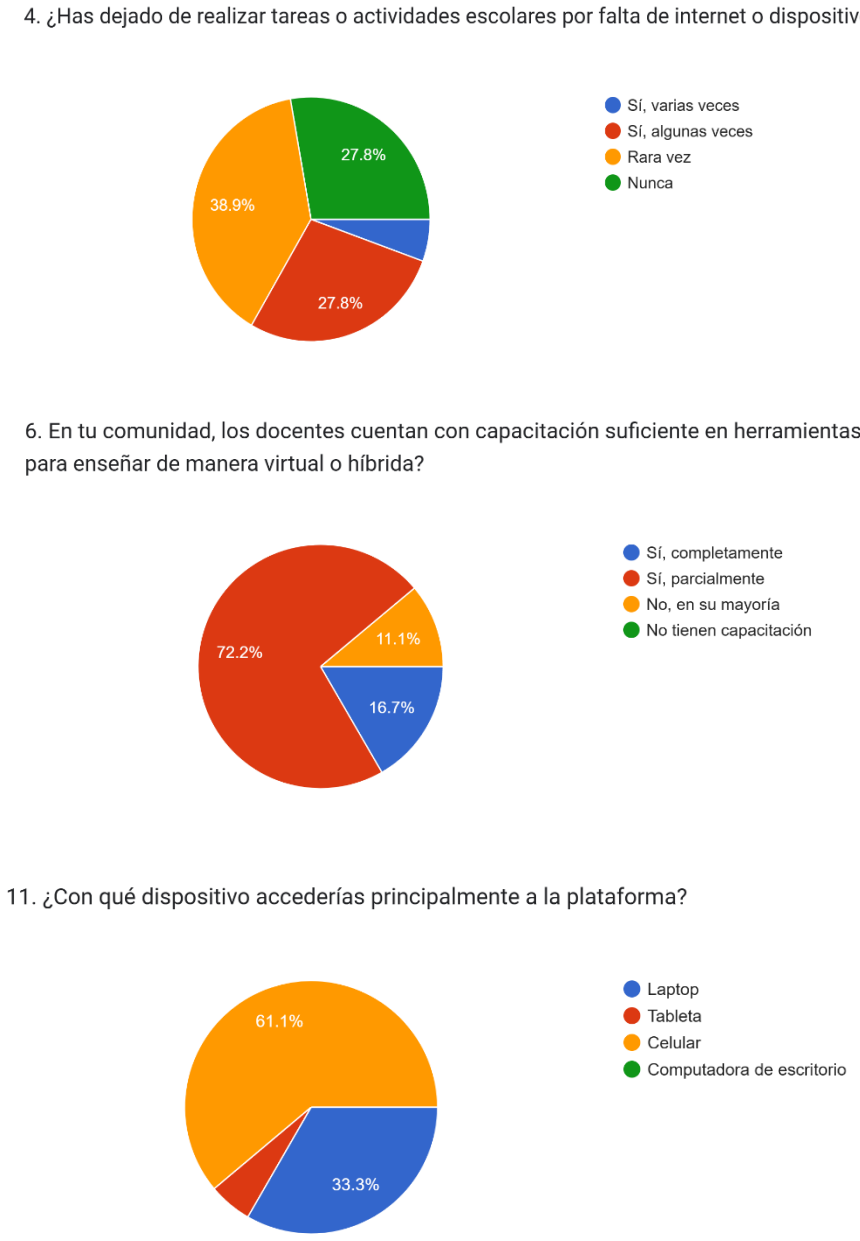


Ilustración 1: Graficas 4, 6 y 11. Resultados de encuestas Conectividad, Capacitación y Dispositivos de Acceso

Manzano Villafaña Gerardo
Mildred

Como podemos observar en la gráfica 4, una parte significativa de los estudiantes ha dejado de realizar tareas o actividades escolares debido a la falta de internet o dispositivos. Destaca que un 27.8% lo ha experimentado en varias ocasiones y otro 27.8% algunas veces, lo cual refleja cómo la carencia de recursos digitales afecta directamente el rendimiento académico.

En la gráfica 6, se aprecia la situación de los docentes respecto a la capacitación digital. La mayoría (72.2%) indicó que no cuentan con preparación suficiente para enseñar en modalidades virtuales o híbridas, mientras que solo un 16.7% afirmó estar completamente capacitado. Este dato revela una necesidad urgente de formación tecnológica para los profesores de comunidades rurales.

Finalmente, en la gráfica 11 se observa que el celular es el dispositivo más utilizado para acceder a la plataforma (61.1%), seguido de la laptop (33.3%). Este resultado confirma que el diseño de cualquier solución tecnológica debe priorizar la compatibilidad móvil, ya que es el medio más accesible para los estudiantes.

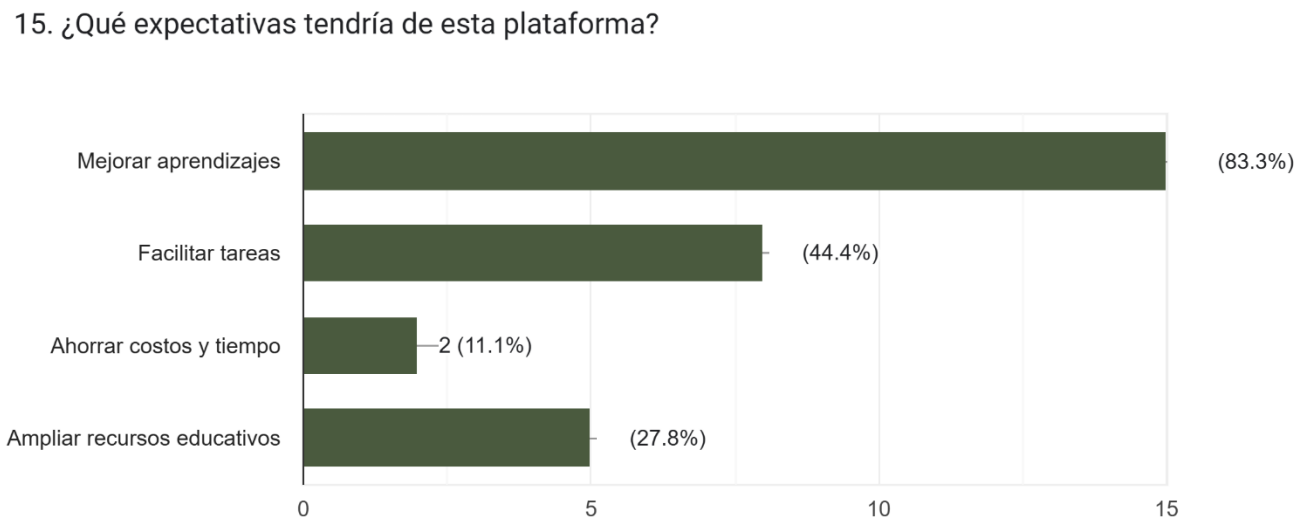


Ilustración 2: Grafica 15 Expectativas de la Plataforma.

En la gráfica 15 se muestran las expectativas que los encuestados tienen respecto a la plataforma propuesta. La mayoría, con un 83.3%, considera que la principal expectativa es mejorar los aprendizajes, lo que refleja la importancia de contar con una herramienta que fortalezca la calidad educativa. Un 44.4% señaló que la plataforma debe facilitar las tareas, destacando la necesidad de simplificar los procesos escolares. Asimismo, un 27.8% espera que contribuya a ampliar los recursos educativos disponibles, mientras que un 11.1% la percibe como una oportunidad para ahorrar costos y tiempo.

Estos resultados dejan en claro que los usuarios valoran el impacto pedagógico de la plataforma demostrando así su utilidad, continuando con la practicidad y la eficiencia en el desarrollo de aprendizaje que se puede obtener de esta.

¿Por Qué Desarrollar esta Plataforma – Solución?

El desarrollo de una solución tecnológica que atienda la brecha digital es esencial en un mundo cada vez más dependiente de la información y la conectividad. La exclusión digital no solo limita el acceso a plataformas educativas o a oportunidades laborales, sino que además perpetúa desigualdades históricas que afectan principalmente a las comunidades más vulnerables. En este sentido, la propuesta de un sistema accesible, seguro y adaptable representa una estrategia clave para promover la inclusión social y el desarrollo sostenible.

La necesidad de esta solución en varios aspectos:

- Educativo: La pandemia por COVID-19 demostró que la educación en línea es una herramienta indispensable. Sin embargo, la falta de conectividad dejó a millones de estudiantes fuera de sus clases. Una plataforma con soporte offline permitiría a los alumnos descargar contenidos y acceder a ellos en cualquier momento, evitando el rezago escolar.

- Social: La exclusión digital incrementa la marginación social. Quienes carecen de acceso a tecnología tienen menos posibilidades de participar en procesos comunitarios, de expresar sus opiniones en espacios digitales y de beneficiarse de programas gubernamentales en línea.

- Económico: En un mercado laboral cada vez más digitalizado, quienes no tienen competencias tecnológicas enfrentan barreras de acceso a empleos de calidad.

- Cultural: La falta de acceso a la tecnología no solo priva a las personas de herramientas prácticas, sino también de experiencias culturales y recreativas.

La solución propuesta, basada en arquitecturas de software modernas como SPA (Single Page Application) y microservicios, permitirá que el sistema sea ligero, escalable y multiplataforma, facilitando su uso en dispositivos de gama baja y en contextos con conectividad limitada. Además, al incluir un modo offline con sincronización diferida, se asegura que los usuarios puedan acceder a la información en todo momento, independientemente de la disponibilidad de internet.

Desarrollar esta solución no es solo un reto tecnológico, sino una responsabilidad social. Se trata de ofrecer a las comunidades más vulnerables un espacio digital donde puedan aprender, comunicarse y desarrollarse, cerrando así la brecha que limita sus oportunidades de crecimiento.

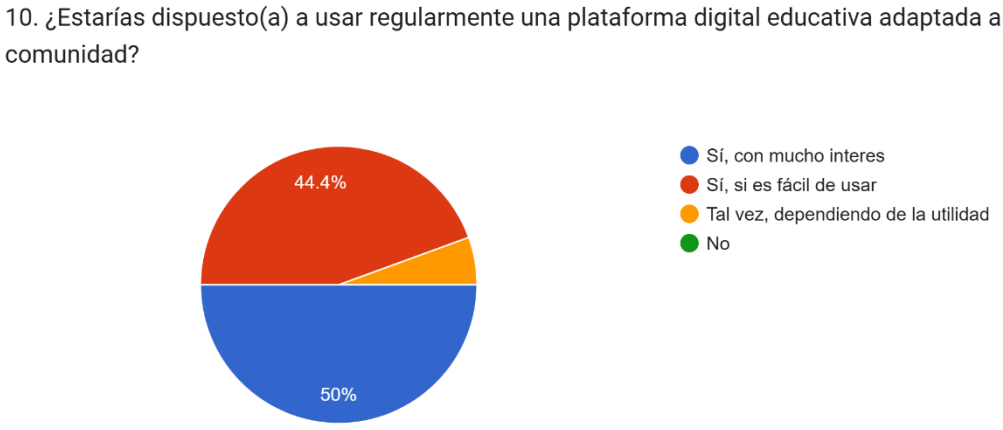


Ilustración 3: Grafica Disposición de los encuestados para utilizar la plataforma en su comunidad

En la gráfica 10 se muestran las respuestas a la disposición de los encuestados para utilizar de manera regular una plataforma digital educativa adaptada a su comunidad. El 50% manifestó que lo haría con mucho interés, mientras que un 44.4% señaló que la usaría si resulta fácil de utilizar. Solo un 5.6% respondió que tal vez, dependiendo de la utilidad, y ningún participante eligió la opción negativa. Estos resultados reflejan una alta aceptación potencial de la plataforma, siempre que se priorice la facilidad de uso y se asegure una experiencia intuitiva y accesible para los usuarios.

Objetivo General: Diseñar e implementar una plataforma híbrida de educación digital que brinde a estudiantes y docentes de comunidades rurales la posibilidad de acceder a contenidos educativos de calidad, programas de capacitación docente y herramientas interactivas de aprendizaje, tanto en modalidad en línea como fuera de línea. La solución busca garantizar la continuidad académica aun en condiciones de baja o nula conectividad, promoviendo la inclusión digital y la igualdad de oportunidades educativas. Asimismo, se pretende que la plataforma contribuya al fortalecimiento de competencias digitales, al aprovechamiento de los recursos tecnológicos disponibles y al desarrollo social y profesional de las comunidades, reduciendo así la brecha digital educativa que limita el acceso equitativo al conocimiento y al progreso en entornos rurales.

Objetivos Específicos

1. **Facilitar el acceso a recursos educativos en entornos con baja conectividad.**
    - Implementar mecanismos que permitan la descarga y consulta de contenidos de manera offline, asegurando que los usuarios puedan continuar su aprendizaje sin depender de una conexión continua.
2. **Diseñar un sistema multiplataforma accesible en dispositivos de gama baja.**
    - Asegurar que la solución sea funcional en computadoras, tablets y teléfonos móviles, priorizando el rendimiento y la compatibilidad con dispositivos de recursos limitados.
3. **Integrar mecanismos de sincronización diferida de datos.**
    - Permitir que los usuarios trabajen sin conexión y que la información se sincronice automáticamente con la base de datos central cuando exista acceso a internet.
4. **Promover la alfabetización digital en las comunidades marginadas.**
    - Incluir contenidos y herramientas que permitan a los usuarios adquirir competencias básicas en el uso de la tecnología y las plataformas digitales.
5. **Garantizar la seguridad y privacidad de la información.**
    - Implementar estándares de seguridad que protejan los datos personales y la información sensible de los usuarios, especialmente en contextos de educación y gestión comunitaria.
6. **Establecer un entorno colaborativo y comunitario.**
    - Permitir la interacción entre usuarios a través de foros o espacios digitales, promoviendo la creación de comunidades de aprendizaje y de apoyo mutuo.
7. **Incorporar un diseño centrado en el usuario (UX).**
    - Diseñar una interfaz intuitiva y fácil de usar, adaptada a personas con diferentes niveles de alfabetización digital, para reducir barreras tecnológicas y fomentar el uso continuo de la plataforma.
8. **Fomentar la escalabilidad del sistema.**
    - Diseñar la solución con la capacidad de crecer en número de usuarios, funcionalidades y recursos sin afectar su rendimiento ni comprometer la experiencia del usuario.

Requerimientos Funcionales

Los requerimientos funcionales (RF) definen las características y acciones que el sistema debe ser capaz de realizar para cumplir con los objetivos planteados. A continuación, se presentan los más relevantes, cada uno con su descripción, prioridad y funcionalidad.

1. **Registro y autenticación de usuarios (docentes, estudiantes, administradores).**
    - **Descripción:** El sistema debe permitir que los usuarios se registren y autentiquen de forma segura, utilizando correo electrónico, número de teléfono o autenticación mediante terceros (Google, Facebook). Además, debe existir la opción de acceso anónimo para fomentar la participación en comunidades sin comprometer la identidad.
    - **Prioridad:** Alta.
    - **Funcionalidad:** El usuario podrá crear una cuenta, recuperar contraseñas y establecer un perfil. En el caso del acceso anónimo, se permitirá la navegación y uso básico de recursos, garantizando al mismo tiempo la privacidad.

2. **Acceso a contenidos educativos en línea y posibilidad de descarga para consulta offline.**
    - **Descripción:** Los usuarios deben poder descargar recursos educativos (artículos, videos, ejercicios, guías) para consultarlos sin conexión.
    - **Prioridad:** Alta.
    - **Funcionalidad:** El sistema guardará temporalmente los archivos en la memoria del dispositivo y los sincronizará para actualizaciones cuando se restablezca la conexión a internet.

3. **Sincronización de Datos.**
    - **Descripción:** El sistema debe sincronizar automáticamente los cambios realizados por los usuarios mientras estaban sin conexión.
    - **Prioridad:** Alta.
    - **Funcionalidad:** Una vez que se detecte conexión, los datos se actualizarán en la base central (por ejemplo: progreso en cursos, aportes en foros, descargas nuevas).

4. **Biblioteca digital organizada por materias y niveles educativos.**
    - **Descripción:** Los usuarios deben tener acceso a una biblioteca organizada con recursos categorizados por temas.
    - **Prioridad:** Media.
    - **Funcionalidad:** El sistema permitirá filtrar, buscar y marcar recursos. Además, ofrecerá recomendaciones basadas en intereses del usuario.

5. **Creación y participación en comunidades de apoyo.**
    - **Descripción:** Los usuarios podrán unirse a comunidades temáticas para compartir experiencias, plantear dudas o colaborar en proyectos.
    - **Prioridad:** Media.
    - **Funcionalidad:** Foros moderados, secciones de preguntas frecuentes y espacios de intercambio con posibilidad de interacción anónima.

6. **Comunicación entre usuarios y moderadores.**
    - **Descripción:** El sistema incluirá un módulo de mensajería básica para aclarar dudas o recibir orientación en temas de conectividad o uso de la plataforma.
    - **Prioridad:** Media.
    - **Funcionalidad:** Chat seguro y limitado, con filtros automáticos para prevenir abuso o mal uso del sistema.

7. **Gestión de perfiles de usuario.**
    - **Descripción:** Los usuarios podrán personalizar su perfil con información básica y preferencias de uso.
    - **Prioridad:** Baja.
    - **Funcionalidad:** Configuración de idioma, accesibilidad (fuentes grandes, modo contraste), intereses temáticos, etc.

8. **Panel de Administración para Moderadores.**
    - **Descripción:** Los administradores del sistema tendrán acceso a un panel de control para gestionar usuarios, comunidades y contenidos**.**
    - **Prioridad:** Alta.
    - **Funcionalidad:** Permite suspender cuentas, aprobar recursos educativos y supervisar interacciones para mantener un entorno seguro.

Requerimientos no Funcionales

Los requerimientos no funcionales establecen las cualidades y condiciones bajo las cuales el sistema debe operar para ser eficiente, seguro y accesible. No describen lo que el software hace, sino cómo lo hace, asegurando la confiabilidad, usabilidad y sostenibilidad de la solución tecnológica. Entre ellos se consideran aspectos como el rendimiento, la escalabilidad, la seguridad de los datos, la accesibilidad para todos los usuarios y la capacidad de funcionar en dispositivos de bajo costo y con conectividad limitada.

1. **Seguridad y Privacidad de los datos.**
    - **Descripción:** Toda la información personal y sensible debe estar protegida mediante encriptación y protocolos de seguridad.
    - **Prioridad:** Alta.
    - **Explicación**: Garantizar la confianza de los usuarios es fundamental, especialmente en comunidades vulnerables donde el mal uso de la información podría tener consecuencias graves.

2. **Escalabilidad del sistema.**
    - **Descripción:** El sistema debe poder crecer en número de usuarios y funcionalidades sin comprometer el rendimiento.
    - **Prioridad:** Media.
    - **Explicación:** A medida que más comunidades adopten la plataforma, será necesario aumentar servidores y capacidad de almacenamiento.

3. **Desempeño y tiempo de respuesta.**
    - **Descripción:** El sistema debe responder en menos de 3 segundos incluso en dispositivos de gama baja.
    - **Prioridad:** Alta.
    - **Explicación:** Una experiencia lenta puede generar frustración y abandono del sistema, especialmente en usuarios con poca experiencia digital.

4. **Disponibilidad y fiabilidad del servicio.**
    - **Descripción:** El sistema debe estar disponible al menos el 95% del tiempo, con mecanismos de respaldo para evitar caídas prolongadas.
    - **Prioridad:** Alta.
    - **Explicación:** En comunidades con pocas ventanas de conectividad, es esencial que el sistema esté operativo cuando el usuario logre conectarse.

5. **Usabilidad y experiencia de usuario (UX).**
    - **Descripción:** La interfaz debe ser clara, intuitiva y accesible, diseñada con principios de diseño universal.
    - **Prioridad:** Alta.
    - **Explicación:** Muchos usuarios no cuentan con altos niveles de alfabetización digital, por lo que la facilidad de uso es determinante.

6. **Soporte multiplataforma.**
    - **Descripción:** La plataforma debe funcionar en cualquier dispositivo, sin importar el sistema operativo.
    - **Prioridad:** Media.
    - **Explicación:** Asegura la inclusión de comunidades con dispositivos variados y en muchos casos obsoletos.

7. **Consumo reducido de recursos.**
    - **Descripción:** La aplicación debe ser ligera, ocupando poco espacio en memoria y con bajo consumo de datos móviles.
    - **Prioridad:** Alta.
    - **Explicación:** En contextos con limitaciones económicas, un sistema que consuma muchos datos o recursos será rápidamente descartado.

8. **Mantenimiento y actualización continua.**
    - **Descripción:** El sistema debe permitir actualizaciones frecuentes sin afectar la disponibilidad.
    - **Prioridad:** Baja.
    - **Explicación:** Facilita la incorporación de mejoras y garantiza seguridad a largo plazo.

9. **Accesibilidad Universal.**
    - **Descripción:** El sistema debe incluir opciones de accesibilidad (lectura de texto, contraste alto).
    - **Prioridad:** Media.
    - **Explicación:** Esto garantiza la inclusión de personas con discapacidades visuales o limitaciones motrices.
10. **Procesamiento seguro de pagos**
    - **Descripción:** En caso de incluir servicios premium o pagos por certificaciones, estos deben procesarse de manera segura.
    - **Prioridad:** Baja.
    - **Explicación:** Aunque no es una necesidad inicial, es importante contemplar la opción futura de transacciones seguras para la sostenibilidad de la plataforma.

Justificación de los Requerimientos no Funcionales

En el caso de una plataforma educativa dirigida a comunidades rurales, la justificación de estos requerimientos cobra especial relevancia, ya que las condiciones de baja conectividad, dispositivos de gama baja y usuarios con poca experiencia tecnológica hacen indispensable que el sistema opere bajo estándares de simplicidad, confiabilidad y accesibilidad.

1. **Seguridad y privacidad de los datos:** La información de los usuarios pertenece a los datos más sensibles del sistema. Muchas comunidades no tienen experiencia en medidas de seguridad digital, lo que las hace vulnerables a ataques y mal uso de sus datos. Garantizar la seguridad y privacidad no es opcional, sino una condición indispensable para la confianza en la plataforma.

   Se requiere implementar encriptación de extremo a extremo, almacenamiento seguro de credenciales y protocolos como HTTPS y JWT para la autenticación. Proteger la confidencialidad genera un entorno de confianza, esencial para que los usuarios no teman en compartir su información ni se genere alguna inseguridad por usar el servicio.

2. **Escalabilidad del sistema:** La escalabilidad asegura que el sistema pueda crecer en número de usuarios y funcionalidades sin afectar el rendimiento. Al inicio, la plataforma puede usarse en una sola comunidad, pero con el tiempo se prevé su expansión a más regiones. Sin una arquitectura escalable, cada aumento en la demanda afectaría el tiempo de respuesta y la estabilidad del sistema.

   Adoptar arquitecturas modulares, con separación entre frontend y backend, así como servicios distribuidos, permitirá que la plataforma se expanda de manera progresiva. Esto garantiza la sostenibilidad a largo plazo y la posibilidad de convertirse en un modelo replicable en diferentes contextos sociales.

3. **Desempeño y tiempo de respuesta:** Un sistema lento es, en la práctica, un sistema inútil. Los usuarios en zonas rurales suelen depender de dispositivos de gama baja y conexiones inestables, por lo que el rendimiento debe estar optimizado. El objetivo es que las páginas y funciones respondan en menos de 3 segundos.

   Para lograrlo, se deben emplear técnicas como almacenamiento en caché, optimización de consultas a bases de datos y compresión de recursos. Un buen desempeño mejora la experiencia del usuario y evita la frustración, especialmente en comunidades con poca alfabetización digital, donde la paciencia frente a fallos tecnológicos es menor.

4. **Disponibilidad y fiabilidad del servicio:** La plataforma debe estar disponible al menos el 95% del tiempo, ya que los usuarios en comunidades desconectadas suelen tener ventanas muy limitadas de conexión. Si el sistema está inactivo en esos momentos, la experiencia del usuario se verá seriamente afectada.
   Por ello, es necesario implementar sistemas de respaldo y recuperación ante fallos, además de servidores redundantes que permitan mantener el servicio activo. La fiabilidad del sistema refuerza la confianza en la tecnología y asegura que los usuarios consideren la plataforma como una herramienta confiable

5. **Usabilidad y experiencia de usuario (UX):** La usabilidad es uno de los factores más determinantes en este proyecto. Muchos de los usuarios potenciales tienen una alfabetización digital limitada, por lo que el diseño debe ser intuitivo, sencillo y accesible. Una interfaz compleja puede convertirse en una barrera tan grande como la falta de internet.
   Se recomienda aplicar principios de diseño universal: menús simples, iconos representativos, textos claros y opciones de accesibilidad como contraste alto y tamaño de fuente configurable. Además, pruebas de usabilidad con usuarios reales ayudarán a ajustar la interfaz a las necesidades del público meta.

6. **Soporte multiplataforma:** En comunidades marginadas, los usuarios emplean una gran diversidad de dispositivos: desde teléfonos básicos con Android antiguo, hasta laptops de segunda mano y equipos escolares. Por ello, el sistema debe ser multiplataforma y adaptarse a distintos entornos sin perder funcionalidad.
   Diseñar una aplicación basada en SPA (Single Page Application) con un backend robusto facilita esta compatibilidad. Asegura que los recursos estén disponibles sin importar el dispositivo o sistema operativo, contribuyendo así a la inclusión digital.

7. **Consumo reducido de recursos:** Uno de los RNF más críticos es el bajo consumo de datos y recursos. Para muchas familias, los planes de datos móviles son limitados y costosos. Una aplicación que consuma demasiado ancho de banda o requiera dispositivos de alta gama se vuelve inaccesible.
   Por esta razón, la plataforma debe optimizar el uso de recursos: comprimir archivos multimedia, evitar procesos innecesarios en segundo plano y limitar el peso de las descargas. El objetivo es que los usuarios puedan usar el sistema sin afectar significativamente sus gastos en conectividad.

8. **Mantenimiento y actualización continua:** La tecnología avanza rápidamente, y un sistema que no se actualiza se vuelve obsoleto en poco tiempo. El mantenimiento continuo asegura que la plataforma permanezca segura, confiable y alineada con las necesidades de los usuarios.
   Este RNF se justifica porque las comunidades en desventaja no pueden permitirse sistemas inestables. Las actualizaciones deben realizarse de manera transparente y sin interrupciones, de modo que los usuarios no perciban fallos ni pérdidas de servicio.

9. **Accesibilidad Universal:** La inclusión digital no solo depende del acceso a internet, sino también de que la plataforma pueda ser utilizada por personas con discapacidades físicas o sensoriales. Por ello, la accesibilidad universal es un requisito indispensable.
   Se deben aplicar estándares como WCAG 2.1, incorporando lectores de pantalla, navegación por teclado, subtítulos en videos y opciones de personalización visual. Esto no solo amplía el alcance del sistema, sino que cumple con un compromiso ético y social de garantizar igualdad de oportunidades para todos.

10. **Procesamiento seguro de Pagos (futuro):** Aunque la plataforma será gratuita en su fase inicial, es posible que en el futuro se incluyan servicios adicionales, como certificaciones o cursos especializados. Para entonces, será fundamental contar con un sistema de pagos seguros que cumpla con los estándares internacionales (PCI-DSS).
    Esto asegura que, si se incorporan modelos de sostenibilidad económica, los usuarios puedan realizar transacciones sin temor a fraudes o pérdidas financieras. Aunque no es un requerimiento inmediato, incluirlo desde el diseño inicial prepara a la plataforma para evolucionar con seguridad.

Propuesta 1: Arquitectura SPA con API REST (Offline-first)

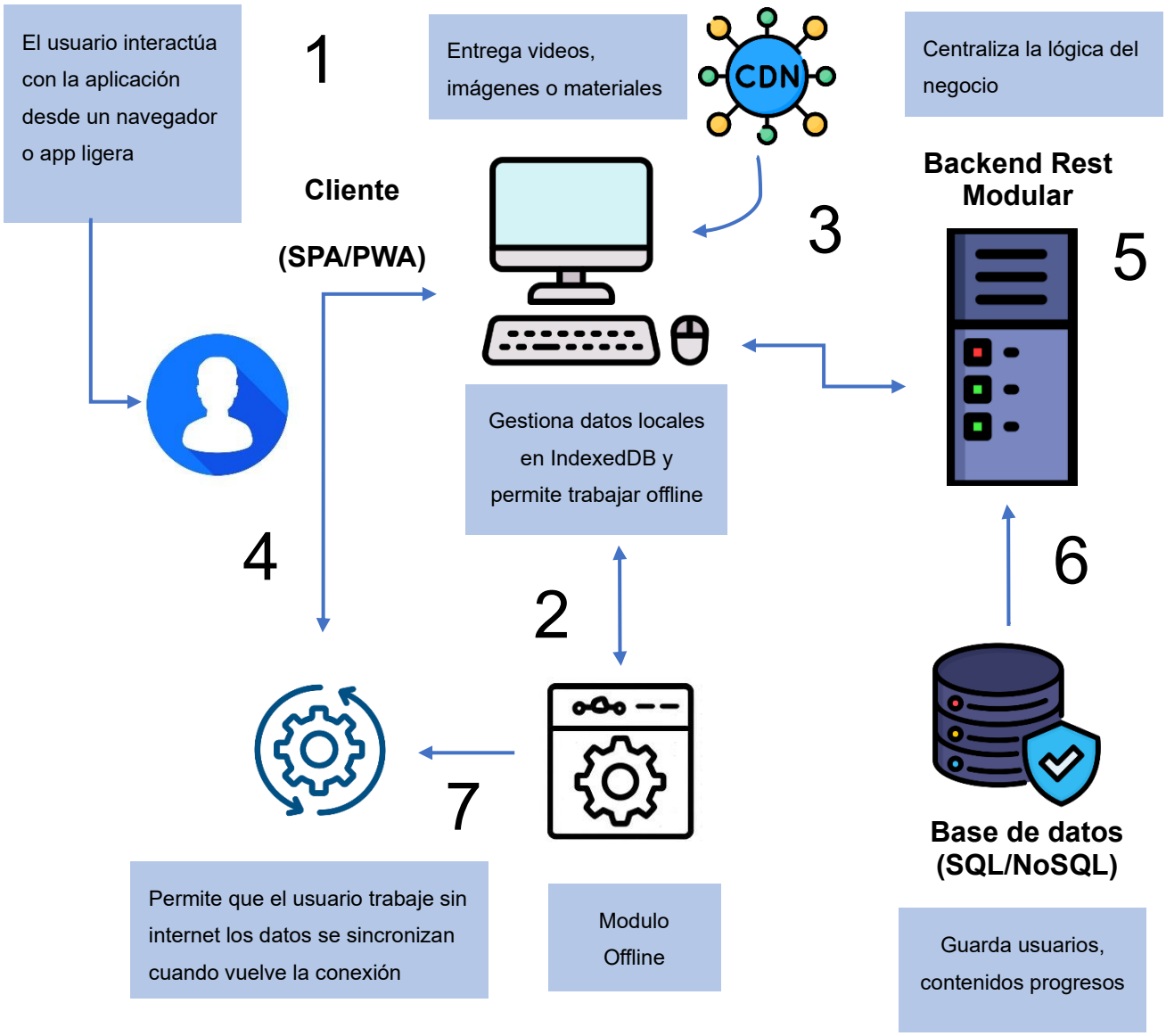
La primera propuesta arquitectónica consiste en el uso de una Aplicación de Página Única (SPA) apoyada en APIs REST para la comunicación con el servidor. En este modelo, el frontend se encarga de gestionar la experiencia del usuario y las funcionalidades de modo offline, mientras que el backend centraliza la lógica de negocio y el acceso a la base de datos. Este enfoque resulta altamente adecuado para comunidades rurales que enfrentan problemas de conectividad, pues la SPA tiene la capacidad de almacenar datos localmente y sincronizarlos de manera diferida una vez que el dispositivo recupere conexión. De esta manera, se asegura que estudiantes y docentes puedan continuar con sus actividades educativas aun en condiciones de red inestable o limitada.

Justificación

La elección de esta arquitectura responde a la necesidad de ofrecer una experiencia de usuario fluida y accesible, incluso en dispositivos de gama baja que predominan en las comunidades rurales. Al reducir los tiempos de carga, empaquetar de manera ligera los recursos y eliminar recargas completas en cada interacción, la SPA garantiza un mejor aprovechamiento de los limitados recursos de hardware y datos móviles. Además, al utilizar APIs REST con formato JSON, se logra compatibilidad con una amplia variedad de dispositivos y sistemas operativos, lo cual es crucial en contextos donde los usuarios utilizan celulares económicos, tabletas escolares o equipos de segunda mano. La simplicidad de REST facilita también el mantenimiento y la extensión del sistema en fases posteriores.

Por otro lado, el uso de tecnologías como IndexedDB, LocalStorage y Cache Storage, junto con Service Workers, permite implementar un modelo offline-first, en el cual la aplicación puede ejecutarse aun cuando no exista conexión a internet. Este aspecto es clave para reducir la brecha digital, ya que asegura que los alumnos no dependan de una conexión constante para acceder a materiales educativos.

Modelo Arquitectura SPA con API REST (Offline)



Manzano Villafaña Gerardo
Mildred

La arquitectura se compone de tres capas principales:

- **Cliente (SPA):** Desarrollado en frameworks como Angular o React, con diseño accesible y responsivo, gestor de estado, servicios de sincronización y módulos de accesibilidad (fuentes grandes, contraste, navegación por teclado).

- **API Gateway ligero:** Encargado de terminar conexiones TLS/HTTPS, aplicar controles de seguridad como rate limiting y enrutar solicitudes hacia el backend.

- **Backend REST:** Estructurado en módulos (usuarios, autenticación, contenidos, comunidades), con repositorios y colas de sincronización para procesar cambios diferidos.

Adicionalmente, la arquitectura incluye una base de datos híbrida (relacional o NoSQL según el tipo de información), un CDN para optimizar la entrega de recursos estáticos y un sistema de observabilidad (logs, métricas y trazas).

Flujo de operaciones clave

El flujo de información en esta propuesta puede describirse de la siguiente forma:

- Cuando el usuario accede en línea, la SPA se comunica con el API Gateway, el cual envía la petición al backend, procesa la lógica correspondiente y devuelve una respuesta en formato JSON.

- En modo offline, la SPA consulta los datos almacenados en IndexedDB y Cache Storage, permitiendo que el usuario continúe trabajando sin conexión.

- Cuando el dispositivo se reconecta, un módulo de sincronización se encarga de enviar los cambios pendientes al servidor, resolver posibles conflictos y actualizar la base de datos central.

- La entrega de contenidos pesados, como videos o documentos, se realiza a través de un CDN con carga diferida (lazy loading) y compresión para reducir el consumo de datos.

Ventajas

La arquitectura SPA con API REST ofrece varias ventajas relevantes para el contexto de la brecha digital:

- **Simplicidad operativa:** Ya que se trabaja con un backend único y modular.
- **Excelente experiencia de usuario:** En escenarios de red débil, gracias a la navegación sin recargas y la sincronización transparente.
- **Compatibilidad multiplataforma,** asegurando que la aplicación funcione en dispositivos de distintos niveles de capacidad.
- **Rapidez de implementación:** Lo que permite reducir el tiempo de llegada al usuario final (Time-to-Market).

Limitaciones

Entre las principales limitaciones se encuentran

- La escalabilidad del sistema se orienta más al crecimiento vertical que al horizontal, lo cual podría dificultar su adaptación a un número muy elevado de usuarios en fases avanzadas.
- Existe un mayor acoplamiento lógico dentro del backend modular, ya que los diferentes servicios comparten ciclos de despliegue y mantenimiento.

Casos de uso ideales

Esta arquitectura es especialmente adecuada para:

- Pilotos comunitarios en zonas rurales con poca conectividad.
- Fases iniciales de implementación en una o varias regiones limitadas.
- Funcionalidades como acceso a catálogos de contenido educativo, participación en foros moderados y descarga de materiales para uso offline

Propuesta 2: Arquitectura Híbrida SPA + Microservicios

La segunda propuesta consiste en evolucionar la arquitectura SPA hacia un modelo híbrido basado en microservicios, donde el frontend sigue siendo una Aplicación de Página Única (SPA) con capacidades PWA y soporte offline-first, mientras que el backend se fragmenta en múltiples servicios independientes. Cada microservicio gestiona un dominio específico de la plataforma: autenticación, contenidos, comunidades, sincronización, notificaciones y analítica, entre otros.

Un API Gateway / BFF se encarga de centralizar la seguridad, la autenticación y la orquestación de peticiones, ofreciendo al frontend una interfaz unificada.

Para el despliegue de estos microservicios se propone el uso de Docker, lo que permite encapsular cada servicio en contenedores independientes y portables. De esta forma, cada microservicio puede escalarse de manera individual, reiniciarse automáticamente en caso de fallas y mantenerse aislado de los demás. A medida que la solución crezca, puede integrarse un orquestador de contenedores como Kubernetes para gestionar la disponibilidad, escalabilidad y balanceo de carga.

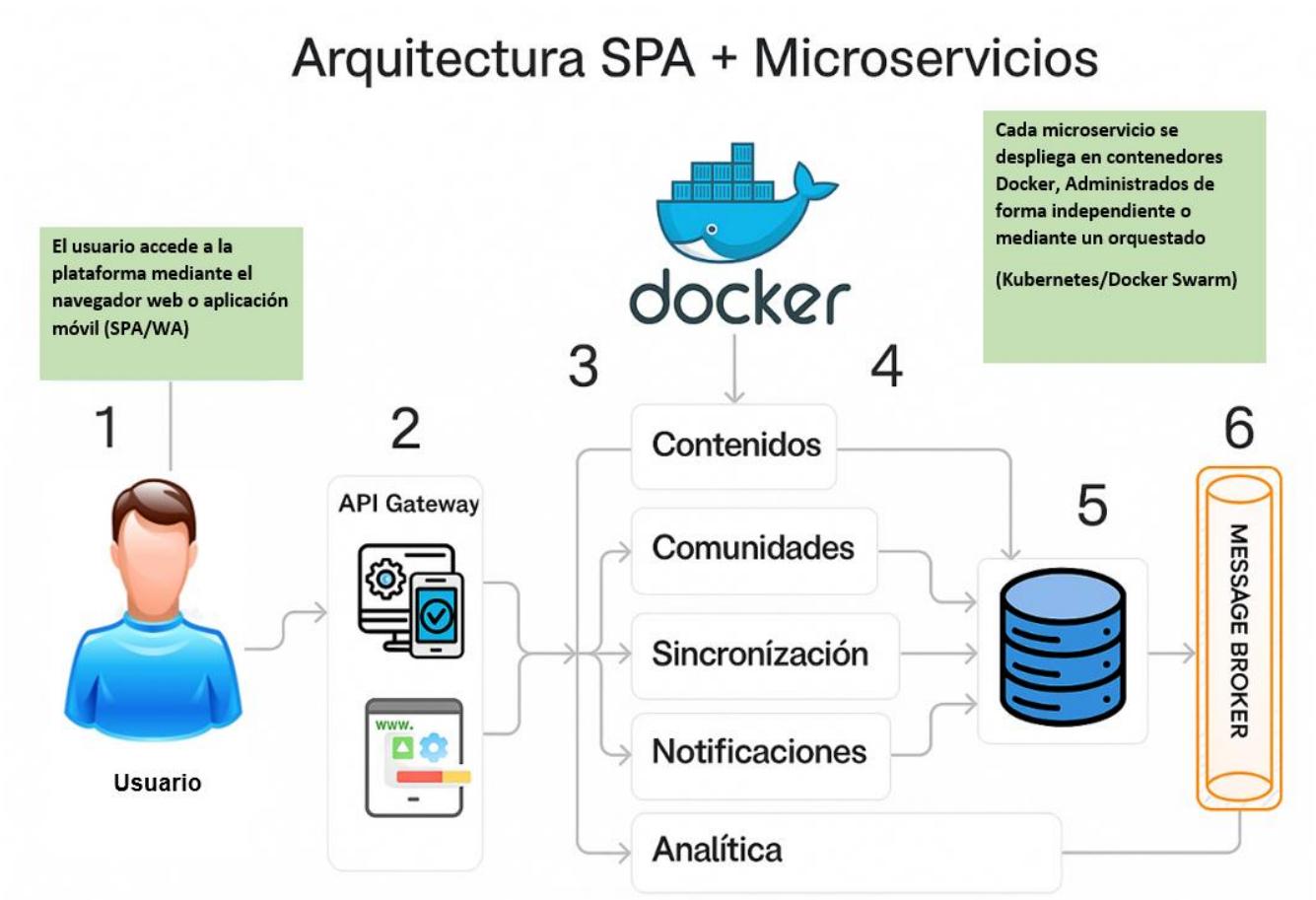
Justificación

El uso de Docker en esta propuesta aporta varios beneficios esenciales:

- **Portabilidad:** cada servicio se ejecuta en un contenedor con todas sus dependencias, garantizando que funcione igual en entornos de desarrollo, pruebas y producción.
- **Escalabilidad selectiva:** los servicios con mayor demanda, como “Contenidos” o “Sincronización”, pueden escalarse rápidamente sin necesidad de replicar todo el sistema.
- **Mantenimiento independiente:** actualizaciones o cambios en un microservicio no afectan al resto de la plataforma.
- **Orquestación avanzada:** herramientas como Kubernetes o Docker Swarm permiten desplegar múltiples réplicas de servicios, aplicar balanceo de carga y garantizar alta disponibilidad.

Este enfoque es especialmente adecuado en contextos donde la plataforma se expanda hacia diferentes comunidades rurales de México y Latinoamérica, pues facilita una infraestructura flexible y sostenible para enfrentar demandas crecientes.

Esquema global



La arquitectura híbrida con Docker se compone de:

- **Cliente SPA/PWA:** interfaz accesible y responsiva, con soporte offline mediante almacenamiento local y Service Workers.
- **API Gateway / BFF:** actúa como punto de entrada único, validando la seguridad y orquestando las peticiones hacia los microservicios.
- **Microservicios de dominio (contenedorizados con Docker):**
  - **Auth:** registro, inicio de sesión y políticas de seguridad.
  - **Contenidos:** catálogo educativo, descargas y gestión de versiones.
  - **Comunidades:** foros, moderación y participación estudiantil.
  - **Sincronización:** ingesta de datos diferidos y reconciliación de conflictos.
  - **Notificaciones:** envío de alertas, correos y mensajes push.
  - **Analítica:** métricas de uso y tableros de aprendizaje.
- **Bases de datos independientes:** gestionadas por cada microservicio (polyglot persistence: SQL o NoSQL según necesidad).

- **Sistema de mensajería/streaming:** Kafka o RabbitMQ en contenedores, para coordinar eventos asincrónicos entre microservicios.
- **CDN/nube:** distribución de recursos estáticos (videos, PDFs, imágenes).
- **Orquestador (opcional, fase avanzada):** Kubernetes para automatizar despliegues, escalado y monitoreo de los contenedores.

Ventajas

- **Aislamiento y portabilidad:** cada microservicio corre en su propio contenedor Docker.
- **Alta escalabilidad:** es posible escalar selectivamente los servicios críticos.
- **Resiliencia:** fallos en un servicio no afectan el resto del sistema.
- **Flexibilidad tecnológica:** cada microservicio puede usar la base de datos o framework que mejor se ajuste a su función.
- **Automatización con orquestadores:** Kubernetes permite monitorizar, escalar y reiniciar contenedores de forma autónoma.

Limitaciones

- **Complejidad operativa:** la administración de múltiples contenedores y microservicios requiere experiencia en DevOps.
- **Costos iniciales más altos:** debido al monitoreo, infraestructura y necesidad de orquestadores.
- **Consistencia eventual:** los datos pueden no sincronizarse en tiempo real entre servicios, lo que exige mecanismos robustos de reconciliación.

Casos de uso ideales

Esta arquitectura es idónea para:

- Proyectos que demanden alta disponibilidad y resiliencia en múltiples comunidades.
- Expansión regional en Latinoamérica, con capacidad de atender a miles de usuarios simultáneamente.
- Entornos donde se requiera innovación constante, añadiendo nuevos módulos sin interrumpir el funcionamiento del sistema principal.

Complementos

Además de la descripción de las propuestas arquitectónicas, es importante presentar esquemas globales y representaciones gráficas que permitan visualizar la estructura del sistema. Estos complementos facilitan la comprensión de la solución planteada y muestran de manera clara cómo se organizan las distintas capas y componentes.

Topología SPA

La **topología de una arquitectura SPA (Single Page Application)** se basa en tres niveles principales:

1. **Capa de presentación (cliente):** Representada por los dispositivos de los usuarios (computadoras, tabletas o smartphones). En esta capa se encuentra la interfaz de usuario, que permite la interacción directa con el sistema. La aplicación se carga inicialmente en el navegador y, a partir de ese momento, todas las interacciones se realizan de manera dinámica sin necesidad de recargar la página completa.

2. **Capa de aplicación (servidores):** Incluye el servidor que aloja la aplicación web, el API Gateway y los servicios backend. Aquí se procesan las solicitudes, se ejecutan las reglas de negocio y se gestiona la comunicación entre el cliente y la base de datos.

3. **Capa de datos (almacenamiento):** Compuesta por la base de datos y los servicios de almacenamiento de información. En este nivel se encuentran los contenidos educativos, las configuraciones de usuario y los datos de sincronización.

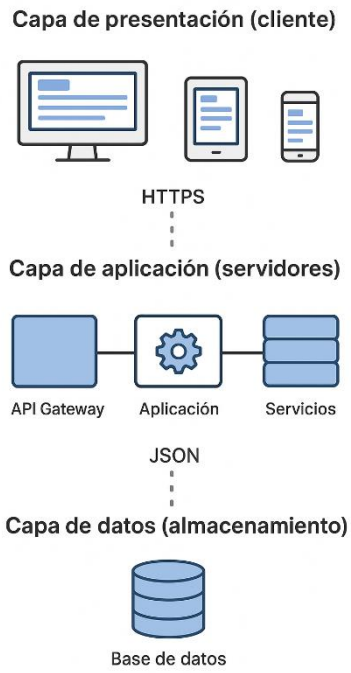


Ilustración 4: TopologÍa SPA

En esta topología, la comunicación entre capas se realiza de forma segura mediante protocolos como HTTPS, y los datos viajan en formato JSON a través de llamadas REST.

La topología SPA destaca por ser ligera y eficiente, ya que permite al cliente manejar la mayor parte de la interacción, mientras que el servidor se limita a procesar peticiones específicas y devolver datos. Esto reduce el consumo de recursos y mejora la experiencia de usuario en entornos con conectividad limitada.

Capas SPA

Dentro de la arquitectura SPA, es posible identificar tres capas lógicas que dividen la funcionalidad del sistema:

1. **Capa de presentación (Frontend):**
    - Desarrollada con frameworks como React, Angular o Vue.
    - Incluye el diseño de la interfaz de usuario, la navegación y los componentes visuales.
    - Gestiona la interacción offline a través de tecnologías como Service Workers y IndexedDB.
    - Se enfoca en la usabilidad y la accesibilidad, incorporando principios de diseño inclusivo para garantizar que cualquier usuario, sin importar su nivel de alfabetización digital, pueda interactuar con la plataforma.
2. **Capa de aplicación (Lógica de negocio):**
    - Ubicada en el servidor, maneja la lógica central de la aplicación.
    - Incluye controladores, servicios y módulos responsables de gestionar autenticación, manejo de contenidos, foros y sincronización.
    - Aplica políticas de seguridad, autorización y validación de datos.
    - Garantiza que las operaciones sean consistentes, incluso cuando se trabaja con datos provenientes de sesiones offline.
3. **Capa de datos (Persistencia):**
    - Corresponde a las bases de datos y servicios de almacenamiento.
    - Puede ser implementada con PostgreSQL (estructurados) y MongoDB (no estructurados), según las necesidades.

- Contiene tanto datos transaccionales (usuarios, registros, descargas) como datos analíticos (métricas de uso, progresos).

- Se diseñará con mecanismos de respaldo y replicación para asegurar disponibilidad y durabilidad.

La división en capas permite una mejor organización del sistema, incrementa la mantenibilidad y facilita el escalamiento futuro. Además, la independencia de cada capa asegura que se puedan realizar cambios en una de ellas sin afectar a las demás, lo que resulta crucial en arquitecturas modernas.

Beneficios de representar complementos

La incorporación de topologías y esquemas de capas no es solo un requisito formal, sino una herramienta clave para:

- **Explicar la arquitectura a usuarios no técnicos:** docentes, administradores o autoridades comunitarias pueden comprender la solución de manera gráfica.

- **Reducir ambigüedades en el diseño:** los diagramas muestran los puntos de comunicación y posibles cuellos de botella.

- **Facilitar la implementación:** los equipos de desarrollo pueden usar los diagramas como referencia al construir cada módulo.

- **Servir como documentación viva:** permiten mantener una visión global del sistema durante todo el ciclo de vida del software.

Conclusión

El presente documento ha permitido analizar, estructurar y proponer una solución tecnológica que atienda de manera específica la problemática de la brecha digital en comunidades marginadas. A lo largo de su desarrollo, se han abordado desde la descripción clara de la problemática hasta la definición de requerimientos funcionales y no funcionales, pasando por la justificación de cada uno de ellos y la presentación de alternativas arquitectónicas viables.

La brecha digital representa uno de los retos más importantes de la sociedad contemporánea. Las diferencias en acceso a la tecnología, conectividad y alfabetización digital generan desigualdades educativas, económicas y sociales que se ven reflejadas en la exclusión de miles de personas. Resolver esta problemática no solo implica mejorar la infraestructura, sino también diseñar plataformas de software accesibles, seguras y adaptables a contextos con recursos limitados.

En este sentido, la propuesta arquitectónica presentada constituye un paso significativo hacia la construcción de soluciones inclusivas. La primera alternativa, basada en una SPA con API REST, ofrece una arquitectura ligera y rápida de implementar, con soporte offline que garantiza acceso a los recursos incluso en escenarios de baja conectividad. La segunda alternativa, una arquitectura híbrida entre SPA y microservicios, se plantea como una evolución natural que permitirá mayor escalabilidad, resiliencia y flexibilidad a medida que el sistema crezca y se expanda hacia más comunidades.

Ambas propuestas han sido acompañadas por la definición detallada de requerimientos funcionales —como el registro de usuarios, la descarga de contenidos offline o la creación de comunidades digitales—, así como de requerimientos no funcionales que aseguran calidad, rendimiento y sostenibilidad —como la seguridad de datos, la escalabilidad del sistema y la accesibilidad universal. La justificación extensa de cada RNF demuestra cómo la calidad del software depende no solo de las funcionalidades visibles para el usuario, sino también de las características que permiten su confiabilidad y permanencia en el tiempo.

Asimismo, se incorporaron complementos gráficos (topologías, capas y esquemas arquitectónicos) que brindan una visión clara y estructurada de la solución. Estos elementos son fundamentales para comunicar de manera sencilla y eficaz cómo se organiza el sistema y cómo se garantizará su desempeño.

En conclusión, la propuesta planteada no se limita a un ejercicio académico, sino que representa una visión concreta y realista de cómo la arquitectura de software puede ser un agente de cambio social. Reducir la brecha digital implica combinar creatividad, responsabilidad social y rigor técnico, lo que hace de este proyecto una muestra de cómo la ingeniería en desarrollo y gestión de software puede contribuir de manera directa al bienestar y progreso de las comunidades.

En el futuro, la implementación de esta plataforma podría ampliarse con nuevas funcionalidades, como la incorporación de microcontenidos educativos dinámicos, la integración de analítica de datos para medir impacto

social, o la posibilidad de incluir modelos de sostenibilidad económica mediante servicios premium. No obstante, lo más importante será mantener el enfoque en la inclusión, accesibilidad y seguridad, asegurando que el sistema siga respondiendo a las necesidades reales de los usuarios.

La tecnología por sí sola no cierra la brecha digital, pero puede ser la herramienta que catalice el cambio. Con un diseño arquitectónico bien fundamentado, acompañado de voluntad social y política, es posible construir un futuro más equitativo donde la conectividad y las oportunidades digitales no sean privilegios, sino derechos universales.

Bibliografía

- Auth0. (2020). *Single Page Apps: Overview*. Recuperado de https://auth0.com/docs/secure
- Fowler, M. (2014). *Microservices: a definition of this new architectural term*. Recuperado de https://martinfowler.com/articles/microservices.html
- Mozilla Developer Network. (s.f.). *Single Page Applications*. Recuperado de https://developer.mozilla.org/en-US/docs/Glossary/SPA
- What is OpenTelemetry? (2025, 24 junio). OpenTelemetry. https://opentelemetry.io/docs/what-is-opentelemetry/
- Web.dev. (s.f.). *Learn PWA*. Recuperado de https://web.dev/learn/pwa/
- Mika, A. (2025, 22 julio). Single-Page Application Architecture. Web Design, UI/UX, Branding, and App Development Blog. https://www.ramotion.com/blog/single-page-application-architecture/
- Arquitectura de SPA | Desarrollo de servicios digitales. (s. f.). https://desarrollo.juntadeandalucia.es/soluciones-tecnologicas/arquitectura-global/arquitectura-global-contexto/arquitecturas-referencia/arquitectura-spa
- PostgreSQL Global Development Group. (s.f.). *PostgreSQL 16.2 Documentation*. Recuperado de https://www.postgresql.org/docs/16/index.html