



Informe Técnico: Implementación Ágil del EduLink Offline Hub

Curso: TOOLS FOR INNOVATION

Profesor(a): LEIDY YOHANA FLÓREZ GÓMEZ

Estudiante: Rafael Lopez Cordova

Fecha: 7 de Diciembre de 2025

1. Resumen Ejecutivo

El presente informe técnico documenta el diseño e implementación del **EduLink Offline Hub**, un dispositivo de borde (*Edge Device*) autónomo concebido para democratizar el acceso a la educación digital en zonas rurales sin conectividad a internet. Este proyecto aplica metodologías ágiles (Scrum) y principios de ingeniería frugal para desarrollar una solución técnicamente viable y económicamente sostenible.

El sistema integra hardware de bajo consumo energético (Raspberry Pi 5), almacenamiento de alta velocidad (NVMe SSD), y un sistema de energía autónoma basado en paneles solares y baterías LiFePO4, garantizando operación continua 24/7 sin dependencia de la red eléctrica. La arquitectura de software implementa un stack completo que incluye servidor web (Nginx), sistema de gestión de contenidos educativos offline (Kiwix), sincronización inteligente de datos, y una plataforma LMS (Learning Management System) local.

El proyecto se estructura en 5 épicas principales que abarcan desde la infraestructura física hasta la seguridad del sistema, desglosadas en 20 historias de usuario con tareas técnicas específicas. La propuesta garantiza atributos críticos de calidad: **Disponibilidad del 99.9%, Capacidad para 50 usuarios concurrentes, Continuidad operativa de 24 horas, y Seguridad mediante modelo "Walled Garden"**.

Con un costo optimizado de materiales (BOM) de aproximadamente **\$338 USD por unidad**, el EduLink Offline Hub representa una alternativa escalable y sostenible frente a soluciones satelitales costosas, haciendo viable su despliegue masivo en escuelas públicas con presupuestos limitados.

2. Definición de la Propuesta de Valor (Utility & Warranty)

2.1. Descripción de la Solución

El **EduLink Offline Hub** es un dispositivo de borde (*Edge Device*) diseñado para descentralizar el acceso a la educación digital. Actúa como un servidor local autónomo que almacena, gestiona y distribuye contenido educativo vía Wi-Fi en zonas sin cobertura de internet, integrando un mecanismo de sincronización asíncrona para mantener los recursos actualizados.

2.2. Análisis de Utilidad (Funcionalidad)

- **Acceso a Conocimiento Offline:** Funcionalidad crítica que permite a docentes y estudiantes navegar por bibliotecas digitales masivas (Wikipedia, Khan Academy, Simuladores PhET) alojadas localmente en el dispositivo, eliminando la dependencia de la conectividad en tiempo real.
- **Sincronización Diferencial Asíncrona:** El sistema detecta automáticamente redes externas esporádicas (ej. un módem móvil temporal o visita técnica) y descarga únicamente las actualizaciones de contenido y sube los reportes de progreso, optimizando el uso de datos.

2.3. Análisis de Garantía (Atributos de Calidad)

- **Disponibilidad (Availability):** El sistema garantiza una disponibilidad del 99.9% durante el horario escolar mediante una arquitectura de hardware robusta y un software de monitoreo (*Watchdog*) que reinicia automáticamente los servicios críticos (servidor web, base de datos) ante cualquier fallo, sin intervención humana.
- **Capacidad (Capacity):** Diseñado con una arquitectura de servidor web ligero (Nginx con *Reverse Proxy*), el dispositivo soporta hasta 50 conexiones concurrentes de baja latencia, permitiendo que un aula completa consuma video streaming simultáneamente sin degradación del servicio.
- **Continuidad (Continuity):** La operatividad está asegurada frente a fallos de la red eléctrica mediante un sistema UPS híbrido (Panel Solar de 100W + Batería LiFePO4), garantizando autonomía energética de 24 horas y protección de datos ante apagones repentinos.
- **Seguridad (Security):** Implementa un modelo de "Jardín Vallado" (*Walled Garden*). La red Intranet está aislada, protegiendo a los menores de contenidos inadecuados externos, malware y ciberataques. Además, el sistema de archivos es de "solo lectura" para los estudiantes, evitando desconfiguraciones accidentales.

3. Arquitectura del Sistema y Desglose de Épicas

3.1. Visión General de la Arquitectura (Hardware & Software)

El EduLink Offline Hub integra una arquitectura híbrida que combina componentes de hardware de bajo consumo energético con un stack de software optimizado para operación autónoma. El núcleo del sistema es una placa de computación de borde (Raspberry Pi 5 u Orange Pi 5 Plus) que ejecuta un sistema operativo Linux embebido, servicios de red local, servidores de contenido educativo y mecanismos de sincronización inteligente.

3.2. Definición de Épicas (Requerimientos de Alto Nivel)

3.2.1. Épica 01 (E01): Infraestructura de Hardware y Energía Autónoma

Descripción: Diseño e integración de los componentes físicos del dispositivo, incluyendo la placa de computación, almacenamiento, sistema de energía solar y protección ambiental.

Alcance Técnico: Selección de hardware robusto (Raspberry Pi 5 8GB, SSD NVMe 500GB), ensamblaje en carcasa resistente con protección IP54, integración de panel solar de 100W con controlador PWM y batería LiFePO4 de 30Ah para operación continua 24/7 sin dependencia de la red eléctrica.

3.2.2. Épica 02 (E02): Sistema Operativo y Despliegue de Red Local

Descripción: Configuración del entorno de software base y la infraestructura de red inalámbrica.

Alcance Técnico: Instalación de sistema operativo basado en Linux (distribución ligera tipo Raspberry Pi OS Lite) optimizado para operación *headless* (sin monitor). Configuración de `hostapd` para crear el punto de acceso Wi-Fi y `dnsmasq` para la asignación de direcciones IP y resolución de nombres (DNS) local. Implementación de un portal cautivo que redirige automáticamente cualquier solicitud del navegador a la página de inicio del EduLink.

3.2.3. Épica 03 (E03): Gestión y Sincronización Inteligente de Contenidos

Descripción: Lógica de backend para el almacenamiento eficiente y actualización de recursos educativos.

Alcance Técnico: Implementación de servidores de contenido estático (Kiwix para archivos ZIM) y dinámico. Desarrollo de scripts en Python para la "Sincronización Oportunista": detección de conectividad externa, ejecución de `rsync` para actualizaciones diferenciales (bajando solo los bits nuevos para ahorrar ancho de banda) y compresión de logs de telemetría para su envío a la nube central.

3.2.4. Épica 04 (E04): Plataforma LMS y Experiencia de Usuario (Frontend)

Descripción: Interfaz gráfica a través de la cual los usuarios interactúan con el sistema.

Alcance Técnico: Desarrollo de una interfaz web *Responsive* (HTML5/CSS3/JS) accesible desde cualquier navegador móvil. Integración de un LMS ligero (Learning Management System) local que permita a los docentes subir archivos PDF propios, crear exámenes autocorregibles y visualizar tableros de control con el progreso académico de los estudiantes.

3.2.5. Épica 05 (E05): Seguridad, Administración y Telemetría

Descripción: Mecanismos de protección del sistema y herramientas de mantenimiento.

Alcance Técnico: *Hardening* del sistema operativo (cierre de puertos innecesarios, firewall ufw). Creación de roles de usuario con privilegios escalonados (Admin, Docente, Estudiante). Implementación de un dashboard técnico oculto que reporte salud del sistema: temperatura de CPU, voltaje de batería, ciclos de carga y espacio en disco.

4. Product Backlog: Historias de Usuario y Tareas Técnicas

A continuación, se detallan los requerimientos funcionales y no funcionales (RNF) organizados por Épicas.

4.1. Bloque A: Infraestructura y Red (Épicas E01 y E02)

ID	Historia de Usuario (HU)	Tareas Técnicas (Definition of Done)
HU01	(Hardware) Como Técnico de Campo, quiero ensamblar los componentes en una carcasa "rugged" para asegurar que el	1. Seleccionar carcasa con certificación IP54 o superior y mecanizar orificios para ventilación.

ID	Historia de Usuario (HU)	Tareas Técnicas (Definition of Done)
	dispositivo soporte el polvo y golpes en zonas rurales.	<ol style="list-style-type: none"> 2. Diseñar e imprimir en 3D los soportes internos (brackets) para fijar la Raspberry Pi y el disco NVMe. 3. Instalar disipadores térmicos pasivos en el CPU y controlador de carga. 4. Sellar puertos externos con tapones de goma para evitar ingreso de partículas. 5. Realizar pruebas de caída (<i>Drop Test</i>) desde 1 metro para validar resistencia estructural.
HU02	(Energía) Como Director de Escuela , quiero que el dispositivo funcione con energía solar para garantizar el servicio durante cortes de luz.	<ol style="list-style-type: none"> 1. Conectar el panel solar de 100W al controlador de carga PWM. 2. Integrar la batería LiFePO4 de 30Ah y calibrar el voltaje de corte en el BMS. 3. Configurar script de Python para leer el estado de carga vía GPIO. 4. Instalar fusible de protección de 15A en la línea de alimentación. 5. Realizar prueba de descarga continua (<i>Burn-in</i>) de 12 horas sin sol.
HU03	(Red) Como Estudiante , quiero conectarme a la red Wi-Fi "EduLink" sin contraseña compleja para acceder rápidamente a mis clases.	<ol style="list-style-type: none"> 1. Instalar y configurar el servicio hostapd en Linux para crear el punto de acceso. 2. Configurar dnsmasq para asignar IPs dinámicas (DHCP)

ID	Historia de Usuario (HU)	Tareas Técnicas (Definition of Done)
		<p>en el rango 192.168.1.x.</p> <p>3. Definir el SSID visible como "EduLink_Escuela".</p> <p>4. Implementar redirección DNS (<i>Captive Portal</i>) para forzar la carga de la página de inicio.</p> <p>5. Validar conexión estable con dispositivos Android de gama baja.</p>
HU04	<p>(RNF - Capacidad) Como Sistema, debo soportar 50 conexiones concurrentes para servir a toda un aula simultáneamente sin latencia.</p>	<p>1. Realizar <i>tuning</i> de parámetros del kernel de Linux (<code>sysctl.conf</code>) para optimizar conexiones TCP.</p> <p>2. Configurar Nginx como <i>reverse proxy</i> con caché de archivos estáticos en RAM.</p> <p>3. Ejecutar pruebas de estrés de red usando Apache JMeter simulando 50 usuarios.</p> <p>4. Implementar QoS (Quality of Service) para limitar el ancho de banda por usuario a 2Mbps.</p> <p>5. Monitorear uso de CPU/RAM bajo carga máxima para asegurar estabilidad.</p>
HU05	<p>(Mantenimiento) Como Técnico, quiero poder actualizar el firmware mediante una memoria USB para arreglar fallos sin internet.</p>	<p>1. Crear script udev que detecte automáticamente la inserción de un USB autorizado.</p> <p>2. Validar firma digital del paquete de actualización para evitar malware.</p> <p>3. Programar rutina de copia de binarios y backup de</p>

ID	Historia de Usuario (HU)	Tareas Técnicas (Definition of Done)
		<p>configuración previa.</p> <p>4. Implementar reinicio seguro (<i>Soft Reboot</i>) tras la actualización.</p> <p>5. Generar log de auditoría (<code>update.log</code>) con el resultado del proceso.</p>

4.2. Bloque B: Gestión de Contenidos y Backend (Épica E03)

ID	Historia de Usuario (HU)	Tareas Técnicas (Definition of Done)
HU06	(Contenido) Como Docente , quiero tener una copia local de Wikipedia para que mis alumnos investiguen sin internet.	<ol style="list-style-type: none"> 1. Descargar el archivo <i>dump</i> más reciente de Wikipedia en español (formato <code>.ZIM</code>). 2. Instalar y configurar el servidor Kiwix-serve en el puerto 8080. 3. Indexar el contenido para permitir búsquedas de texto completo rápidas. 4. Integrar el buscador de Kiwix en la interfaz principal del EduLink. 5. Verificar la correcta visualización de imágenes y fórmulas matemáticas offline.
HU07	(Sincronización) Como Sistema , quiero actualizar los contenidos automáticamente cuando detecte una red externa para mantener la información vigente.	<ol style="list-style-type: none"> 1. Desarrollar script en Python que haga <i>ping</i> recurrente a servidores conocidos (ej. Google DNS). 2. Configurar cliente Wi-Fi para

ID	Historia de Usuario (HU)	Tareas Técnicas (Definition of Done)
		<p>conectarse a redes pre-guardadas (ej. módem del docente).</p> <p>3. Implementar rsync para descargar solo las diferencias de los archivos (delta updates).</p> <p>4. Programar la ejecución mediante Cron Jobs en horarios nocturnos.</p> <p>5. Gestionar reintentos automáticos en caso de interrupción de conexión.</p>
HU08	(Video) Como Estudiante , quiero ver videos educativos de Khan Academy sin interrupciones (<i>buffering</i>) para aprender mejor.	<p>1. Transcodificar biblioteca de videos a formato MP4 H.264 optimizado para web.</p> <p>2. Organizar estructura de directorios en el NVMe por materia y nivel.</p> <p>3. Implementar reproductor HTML5 ligero (ej. Video.js) con precarga adaptativa.</p> <p>4. Configurar cabeceras de caché HTTP en Nginx para entrega eficiente.</p> <p>5. Realizar pruebas de reproducción simultánea en 20 dispositivos.</p>
HU09	(Búsqueda) Como Estudiante , quiero un buscador interno unificado para encontrar recursos de video y texto rápidamente.	<p>1. Diseñar base de datos SQLite para indexar títulos y descripciones de recursos.</p> <p>2. Desarrollar API de búsqueda en Python (Flask/FastAPI).</p> <p>3. Crear interfaz de búsqueda</p>

ID	Historia de Usuario (HU)	Tareas Técnicas (Definition of Done)
		<p>en el Frontend con filtros por asignatura.</p> <p>4. Implementar autocompletado en la barra de búsqueda.</p> <p>5. Optimizar consultas SQL para tiempos de respuesta menores a 200ms.</p>
HU10	(RNF - Continuidad) Como Sistema , debo guardar los datos de progreso cada 5 minutos para evitar pérdidas si se agota la batería.	<p>1. Configurar persistencia de base de datos (Redis/SQL) con <i>Write-Ahead Logging</i>.</p> <p>2. Crear script de monitoreo de voltaje que fuerce un <i>sync</i> a disco al detectar batería baja (10%).</p> <p>3. Optimizar escrituras en disco para extender la vida útil del NVMe.</p> <p>4. Implementar sistema de recuperación automática tras un apagado forzoso.</p> <p>5. Validar integridad de datos simulando cortes de energía abruptos.</p>

4.3. Bloque C: Plataforma LMS y Experiencia de Usuario (Épica E04)

ID	Historia de Usuario (HU)	Tareas Técnicas (Definition of Done)
HU11	(LMS) Como Docente , quiero subir mis propios archivos PDF y tareas al servidor para compartirlos con la clase.	<p>1. Desarrollar módulo de "Upload" en el backend validando tipos de archivo (PDF, DOCX).</p>

ID	Historia de Usuario (HU)	Tareas Técnicas (Definition of Done)
		2. Crear interfaz de gestión de archivos protegida por contraseña para el docente. 3. Implementar escaneo básico de archivos (validación de cabeceras) por seguridad. 4. Asignar permisos de lectura automática a los usuarios estudiantes. 5. Generar vista previa (thumbnail) de los documentos subidos.
HU12	(Evaluación) Como Estudiante , quiero realizar exámenes tipo test y ver mi nota al instante para autoevaluarme.	1. Diseñar esquema de base de datos para Preguntas, Respuestas y Resultados. 2. Desarrollar lógica de calificación automática en el backend. 3. Crear interfaz de examen interactiva que bloquee la navegación hacia atrás. 4. Mostrar retroalimentación inmediata al finalizar el intento. 5. Exportar resultados a un archivo CSV local para el docente.
HU13	(Analytics) Como Docente , quiero ver un tablero con el progreso de mis alumnos para identificar quiénes necesitan ayuda.	1. Construir consultas SQL para agregar datos de actividad por estudiante. 2. Integrar librería de gráficos (Chart.js) en el dashboard del docente. 3. Implementar filtros por fecha,

ID	Historia de Usuario (HU)	Tareas Técnicas (Definition of Done)
		<p>materia y estudiante.</p> <p>4. Añadir alertas visuales para estudiantes con calificaciones bajas.</p> <p>5. Asegurar que el dashboard sea visible correctamente en tabletas.</p>
HU14	(Perfil) Como Estudiante , quiero tener un perfil de usuario para guardar mi historial de avance.	<ol style="list-style-type: none"> 1. Crear tabla de Usuarios y sistema de sesión local (Cookies/Token). 2. Desarrollar interfaz de "Mi Perfil" con resumen de cursos completados. 3. Implementar función de cambio de contraseña simple. 4. Permitir personalización básica (ej. elegir avatar predefinido). 5. Probar persistencia de sesión tras reinicio del dispositivo.
HU15	(Usabilidad) Como Estudiante de Primaria , quiero una interfaz basada en iconos grandes para navegar fácilmente sin leer mucho texto.	<ol style="list-style-type: none"> 1. Diseñar UI/UX centrada en accesibilidad (botones grandes, alto contraste). 2. Implementar navegación por categorías visuales (Matemáticas = Icono Calculadora). 3. Minimizar la cantidad de clics necesarios para llegar al contenido. 4. Realizar pruebas de usabilidad con niños del rango de edad objetivo.

ID	Historia de Usuario (HU)	Tareas Técnicas (Definition of Done)
		5. Optimizar CSS para carga rápida en navegadores antiguos.

4.4. Bloque D: Seguridad y Administración (Épica E05)

ID	Historia de Usuario (HU)	Tareas Técnicas (Definition of Done)
HU16	(RNF - Seguridad) Como Administrador , debo garantizar que los estudiantes no accedan a configuraciones del sistema para evitar daños.	<ol style="list-style-type: none"> 1. Deshabilitar accesos SSH y puertos no utilizados en la interfaz de red interna. 2. Configurar firewall ufw para denegar tráfico no HTTP/DNS. 3. Proteger el acceso al BIOS/Bootloader con contraseña. 4. Separar privilegios de usuario Linux (root vs www-data). 5. Encriptar partición de administración del sistema.
HU17	(RNF - Disponibilidad) Como Sistema , debo tener un tiempo de actividad (Uptime) del 99.9% durante el horario escolar.	<ol style="list-style-type: none"> 1. Configurar <i>Hardware Watchdog</i> en la Raspberry Pi para reinicio automático si se congela. 2. Implementar servicios systemd con política de <code>Restart=always</code>. 3. Realizar pruebas de estabilidad de 48 horas continuas. 4. Crear script de limpieza de memoria caché periódica. 5. Documentar procedimiento de "Hard Reset" físico para emergencias.

ID	Historia de Usuario (HU)	Tareas Técnicas (Definition of Done)
HU18	(Control) Como Docente , quiero bloquear el acceso a contenidos recreativos durante las horas de examen.	<ol style="list-style-type: none"> 1. Desarrollar módulo de "Modo Examen" que aplica reglas de firewall temporales. 2. Crear botón de activación/desactivación en el panel del docente. 3. Configurar temporizador para desactivación automática tras X minutos. 4. Notificar a los estudiantes conectados que el modo examen está activo. 5. Validar que los videos recreativos sean inaccesibles durante este modo.
HU19	(Auditoría) Como Administrador , quiero ver logs de errores del sistema para diagnosticar fallos.	<ol style="list-style-type: none"> 1. Configurar rsyslog para centralizar logs de Nginx, Sistema y Aplicación. 2. Crear interfaz web oculta para visualización de logs en tiempo real. 3. Implementar rotación de logs (logrotate) para no llenar el disco. 4. Configurar alertas críticas (ej. fallo de disco) en el dashboard admin. 5. Permitir exportación de logs a USB en formato texto.
HU20	(Integración) Como Desarrollador , quiero una API REST documentada para poder crear futuras apps móviles.	<ol style="list-style-type: none"> 1. Definir arquitectura de endpoints JSON para recursos y usuarios. 2. Implementar autenticación basada en Tokens para la API. 3. Documentar endpoints usando Swagger/OpenAPI.

ID	Historia de Usuario (HU)	Tareas Técnicas (Definition of Done)
		4. Realizar pruebas de integración con herramientas como Postman. 5. Implementar <i>Rate Limiting</i> para evitar saturación por peticiones API.

5. Marco de Trabajo y Gestión: Metodología Scrum

Para la ejecución del proyecto "EduLink Offline Hub", se adoptará el marco de trabajo ágil Scrum, permitiendo iteraciones rápidas (Sprints) para validar tanto el hardware como el software en entornos reales.

5.1. Definición de Roles

- **Product Owner (Dueño del Producto):** Representado por el **Coordinador Pedagógico / Líder de la ONG**.
 - *Responsabilidad:* Define "el Qué". Gestiona el Product Backlog, priorizando las historias de usuario según el valor educativo (ej. priorizar Wikipedia Offline sobre el Dashboard de analítica en la primera fase). Valida que el entregable cumpla con las necesidades de los estudiantes rurales.
- **Scrum Master:** Representado por el **Líder de Proyecto Técnico**.
 - *Responsabilidad:* Define "el Cómo" se protege el proceso. Elimina impedimentos técnicos (ej. retrasos en la importación de componentes electrónicos, bloqueos en la configuración de Linux) y asegura que el equipo no sea interrumpido por tareas externas. Facilita los eventos de Scrum.
- **Development Team (Equipo de Desarrollo):** Equipo multidisciplinario de 3 especialistas.
 - *Ingeniero de Sistemas (Backend/Redes):* Encargado de la configuración de la Raspberry Pi, Nginx y seguridad.

- *Desarrollador Frontend*: Encargado de la interfaz web y experiencia de usuario (UX) para niños.
- *Técnico de Hardware/Energía*: Encargado del ensamblaje físico, pruebas de la batería LiFePO4 y el panel solar.

5.2. Eventos de Scrum (Ceremonias)

- **Sprint Planning (Planificación)**: Reunión al inicio de cada Sprint (ciclo de 2 semanas). El equipo selecciona las Historias de Usuario del *Product Backlog* (ej. HU01, HU02) y las descompone en tareas técnicas específicas para comprometerse a una entrega viable.
- **Daily Scrum (Reunión Diaria)**: Reunión de 15 minutos al inicio de la jornada. Cada miembro responde: 1) ¿Qué avancé ayer en el prototipo? 2) ¿Qué haré hoy? 3) ¿Tengo algún bloqueo (ej. "me falta el driver del adaptador Wi-Fi")?
- **Sprint Review (Revisión)**: Al finalizar el Sprint, se presenta el **Incremento** funcionando al Product Owner. *Ejemplo*: Se demuestra en vivo cómo el dispositivo enciende con energía solar y permite conectar un celular, sin usar diapositivas, solo producto real.
- **Sprint Retrospective (Retrospectiva)**: El equipo analiza internamente qué salió bien y qué falló en el proceso (ej. "Subestimamos el tiempo de impresión 3D de la carcasa") para aplicar mejoras en el siguiente Sprint.

5.3. Artefactos de Scrum

- **Product Backlog**: Lista ordenada y viva de todos los requisitos del proyecto (las 20 Historias de Usuario detalladas en la sección 4), priorizadas por su valor para el usuario final.
- **Sprint Backlog**: El subconjunto de historias y tareas técnicas seleccionadas exclusivamente para el Sprint actual. Es el plan de acción táctico del equipo de ingeniería.
- **Incremento**: La versión operativa del *EduLink Offline Hub* al final de cada Sprint. Debe cumplir con la "Definition of Done" (DoD): Código probado, hardware ensamblado, funcional y seguro.

6. Análisis de Viabilidad Técnica y Económica

A diferencia de las soluciones satelitales costosas, esta propuesta se basa en la **Ingeniería Frugal**.

6.1. Optimización de Hardware

Se ha sustituido la especificación inicial de computadoras industriales costosas por una **Raspberry Pi 5 (8GB)**. Esto no solo reduce el consumo energético, sino que asegura soporte comunitario a largo plazo (LTS).

6.2. Eficiencia Energética

La implementación de baterías **LiFePO4** (Litio-Ferrofosfato) en lugar de Plomo-Ácido o Li-Ion estándar garantiza una vida útil de +2000 ciclos (aprox. 5-7 años) frente a los 500 ciclos de las baterías tradicionales, reduciendo drásticamente el costo de mantenimiento (OPEX).

6.3. Presupuesto Optimizado

Mediante la selección estratégica de componentes, se ha logrado reducir el Costo de Materiales (BOM) a **~\$338 USD por unidad**, haciendo viable su despliegue masivo en escuelas públicas con presupuestos limitados.

6.4. Stack Tecnológico Seleccionado

La elección de la Raspberry Pi 5 frente a competidores como Orange Pi 5 Plus se fundamenta en:

- **Ecosistema maduro:** Amplia documentación oficial y soporte comunitario activo
- **Disponibilidad:** Cadena de suministro estable a través de distribuidores autorizados
- **Compatibilidad:** Soporte nativo para sistemas operativos optimizados (Raspberry Pi OS)

- **Consumo energético:** Perfil de potencia optimizado para operación con energía solar

7. Referencias Bibliográficas

- 9to5Linux. (2025). *Raspberry Pi 5 single-board computer now available with 1GB RAM for \$45*. Recuperado de <https://9to5linux.com/raspberry-pi-5-single-board-computer-now-available-with-1gb-ram-for-45-usd>
- Adafruit. (2023). *Raspberry Pi 5 – 8 GB RAM (ID: 5813)*. Recuperado de <https://www.adafruit.com/product/5813>
- Adafruit. (2025). *Official Raspberry Pi A2-Class microSD card – 32GB blank*. Recuperado de <https://www.adafruit.com/product/6010>
- CanaKit. (s.f.). *Raspberry Pi 5 8GB – CanaKit*. Recuperado en diciembre de 2025 de <https://www.canakit.com/raspberry-pi-5-8gb.html>
- Eco-Worthy. (2023). *100W 12V Monocrystalline Solar Panel*. Recuperado de <https://www.eco-worthy.com/products/100w-12v-monocrystalline-solar-panel>
- Eco-Worthy. (2024). *Eco-Worthy 12V 30Ah LiFePO4 battery, over 2000 cycles, lightweight LiFePO4 battery*. Walmart. Recuperado de <https://www.walmart.com/ip/ECO-WORTHY-12V-30Ah-Lithium-Phosphate-Battery-Over-2000-Cycles-Lightweight-LiFePO4-Battery/3558414823>
- Excltic. (2017, 1 de febrero). *Introducción a Scrum... en menos de 5 minutos* [Video]. YouTube. Recuperado de <https://www.youtube.com/watch?v=P25JP0u6UKw>
- LiTime. (2025). *LiTime 12V 100Ah LiFePO4 lithium deep cycle battery*. Recuperado de <https://www.litime.com/products/litime-12v-100ah-lithium-lifepo4-battery>
- MrSolar. (2025). *60W to 80W – Solar panels by wattage*. Recuperado de <https://www.mrsolar.com/60w-to-80w/>
- Newegg. (2022). *nvme ssd 500gb – Search results*. Recuperado de <https://www.newegg.com/p/pl?d=nvme+ssd+500gb>
- Orange Pi. (2025). *Orange Pi 5 Plus 8GB RAM*. Recuperado de <https://orangepi.net/product/orange-pi-5-plus-8gb-ram>

- PiShop US. (s.f.). *Raspberry Pi 5 boards | PiShop US – Official reseller*. Recuperado en diciembre de 2025 de <https://www.pishop.us/product-category/raspberry-pi/raspberry-pi-5/raspberry-pi-5-boards/>
- PowMr. (2024). *10A 12Volt/24Volt PWM solar charge controller*. Recuperado de <https://powmr.com/products/pwm-solar-charge-controller-10amp-12v>
- Project Management Institute. (2017). *Agile Practice Guide*. Project Management Institute.
- Raspberry Pi Foundation. (2025). *1GB Raspberry Pi 5 now available at \$45, and memory-driven price rises*. Recuperado de <https://www.raspberrypi.com/news/1gb-raspberry-pi-5-now-available-at-45-and-memory-driven-price-rises/>
- Raspberry Pi Foundation. (2025). *Buy a Raspberry Pi 5*. Recuperado de <https://www.raspberrypi.com/products/raspberry-pi-5/>
- Raspberry Pi Foundation. (2025). *Raspberry Pi 5 product brief (RP-008348)* [PDF]. Recuperado de <https://pip.raspberrypi.com/documents/RP-008348-DS-3-raspberry-pi-5-product-brief.pdf>
- Raspberry Pi Foundation. (2024). *Raspberry Pi 5 Technical Documentation*. Recuperado de <https://www.raspberrypi.com/documentation/>
- Samsung. (2022). *SSD 970 EVO NVMe M.2 500GB*. Recuperado de <https://www.samsung.com/us/computing/memory-storage/solid-state-drives/ssd-970-evo-nvme-m2-500gb-mz-v7e500bw/>
- Shop Solar Kits. (2025). *Eco-Worthy 12V LiFePO4 battery | 10–30Ah | BMS | 3-year warranty*. Recuperado de <https://shopsolarkits.com/products/eco-worthy-12v-lifepo4-lithium-battery>
- SparkFun Electronics. (2024). *Raspberry Pi 5 – 8GB*. Recuperado de <https://www.sparkfun.com/raspberry-pi-5-8gb.html>
- SparkFun Electronics. (2024). *Raspberry Pi A2-Class SD Card – 32GB*. Recuperado de <https://www.sparkfun.com/raspberry-pi-a2-class-sd-card-32gb.html>
- Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide*. Recuperado de <https://scrumguides.org/scrum-guide.html>

- Scrum.org. (s.f.). *What is Scrum?* Recuperado en diciembre de 2025 de <https://www.scrum.org/resources/what-is-scrum>