

Arquitectura Integral para el Despliegue de Servidores de Minecraft de Alta Disponibilidad en Entornos Docker: Estrategias de Interoperabilidad Cross-Platform y Gestión de Progresión Sostenible

1. Introducción y Definición del Alcance del Proyecto

La administración de servidores de Minecraft ha evolucionado desde simples despliegues de binarios Java hasta convertirse en una disciplina compleja que abarca la orquestación de contenedores, la gestión de redes híbridas y el diseño de experiencias de usuario (UX) para ecosistemas dispares. El presente informe técnico aborda la solicitud de diseñar una infraestructura robusta para un servidor de Minecraft alojado en Docker, con requisitos específicos que desafían las convenciones tradicionales de la administración de servidores de juegos.

El objetivo central es la creación de un entorno de juego persistente que priorice la accesibilidad desde la plataforma **Bedrock Edition** (consolas, móviles, Windows 10/11) sin sacrificar la profundidad de contenido asociada históricamente a la **Java Edition** y su comunidad de "modding". Esta dicotomía presenta un reto técnico significativo: los "mods" tradicionales de Java (basados en cargadores como Forge o Fabric) requieren modificaciones en el cliente del usuario, algo imposible de realizar en consolas como Xbox o PlayStation debido a las restricciones de los ecosistemas cerrados ("Walled Gardens"). Por lo tanto, la arquitectura propuesta se aleja de los modpacks tradicionales client-side para adoptar una estrategia de **Modding Server-Side** (Lado del Servidor), utilizando la API de Paper/Spigot y plugins avanzados que emulan la experiencia de mods técnicos e industriales.

Además, se aborda la problemática psicológica de la "desmotivación por disparidad de progreso" (conocida en diseño de juegos como el problema del "Power Creep" o "Snowballing"). En servidores persistentes, los jugadores con alta disponibilidad de tiempo ("Hardcore") tienden a alcanzar el "End-Game" (juego tardío) en cuestión de días, saturando la economía y trivializando el contenido para los jugadores casuales ("Slow Burn"). Para mitigar esto, este reporte desarrolla una **Estrategia de Progresión Escalonada (Staged Progression System)**, implementando barreras temporales, tecnológicas y geográficas que extienden artificial pero orgánicamente la vida útil del servidor.

El alcance de este documento cubre desde la selección del hardware y la configuración del kernel de Linux para Docker, hasta la ingeniería social de la economía del servidor y la implementación técnica de protocolos de traducción de paquetes en tiempo real mediante **GeyserMC**.

2. Marco Teórico y Selección Tecnológica

2.1 El Paradigma de la Virtualización: Docker en Servidores de Juego

La elección de Docker como plataforma de despliegue no es meramente una preferencia de gestión, sino una necesidad operativa para garantizar la reproducibilidad y la estabilidad. En la administración de servidores de Minecraft, la "Containerización" ofrece ventajas críticas sobre la instalación en "bare metal" (servidor dedicado tradicional):

1. **Aislamiento de Dependencias:** Minecraft Java Edition es extremadamente sensible a la versión del Java Development Kit (JDK). Las versiones modernas (1.20.5+) requieren estrictamente Java 21¹, mientras que versiones anteriores o ciertos plugins legacy pueden depender de Java 8 o 17. Docker permite encapsular el runtime exacto necesario sin contaminar el sistema operativo anfitrión.
2. **Gestión Efímera y Persistencia:** Mediante el uso de volúmenes de Docker, separamos estrictamente la lógica del servidor (el binario .jar y las librerías) de los datos del estado del mundo (/world, /logs). Esto facilita las copias de seguridad y las migraciones: destruir el contenedor no destruye el mundo.
3. **Orquestación de Recursos:** Docker permite imponer límites duros (Hard Limits) de CPU y RAM. En un entorno donde un error de "Memory Leak" en un plugin mal optimizado podría congelar todo el servidor físico, Docker contiene el fallo al contenedor específico, protegiendo otros servicios.

Para este proyecto, utilizaremos la imagen itzg/minecraft-server. Esta imagen es el estándar de facto en la comunidad open-source debido a su capacidad de autogestión: descarga automáticamente la versión correcta de Paper/Spigot, acepta el EULA mediante variables de entorno y ajusta las banderas de la JVM (Java Virtual Machine) dinámicamente según la RAM asignada.²

2.2 La Arquitectura Híbrida: Java como Núcleo, Bedrock como Cliente Prioritario

La solicitud exige explícitamente "priorizar la versión Bedrock". Esto invierte la lógica tradicional donde Bedrock es un añadido secundario. Sin embargo, no existe un software de servidor "Nativo de Bedrock" que soporte mods complejos comparables a los de Java. El software oficial bedrock-server de Mojang es extremadamente limitado en cuanto a plugins, y soluciones de terceros como NukkitX no tienen el ecosistema de mods técnicos que

buscamos.

Por consiguiente, la única arquitectura viable es un **Núcleo Java con Traducción de Protocolo**.

- **Servidor Backend:** Minecraft Java Edition corriendo sobre **PaperMC**. Paper es un "fork" (bifurcación) de Spigot optimizado para alto rendimiento, esencial para manejar la carga extra de la traducción de paquetes.
- **Capa de Traducción:** **GeyserMC**. Este middleware actúa como un intérprete en tiempo real. Intercepta los paquetes de red UDP (User Datagram Protocol) enviados por los clientes Bedrock, los deserializa, los traduce a paquetes TCP (Transmission Control Protocol) comprensibles para Java, y viceversa.²
- **Capa de Autenticación:** **Floodgate**. Dado que los jugadores de Bedrock se autentican vía Xbox Live y no tienen necesariamente una cuenta de Java, Floodgate permite el paso de estas credenciales, generando un perfil UUID especial para que el servidor Java los reconozca como usuarios legítimos sin exigir una licencia de Java.⁵

Esta arquitectura permite que el servidor tenga toda la potencia y complejidad de Java (Slimefun, WorldGuard, CoreProtect), mientras que el usuario final en su iPad o Nintendo Switch percibe una experiencia nativa de Bedrock.

2.3 La Solución de "Mods" Server-Side: Slimefun 4

Aquí abordamos el requisito de "incluir mods" y un "modpack tardado". Como se estableció, los clientes de consola no pueden instalar mods. La solución es **Slimefun 4**.

Slimefun es un plugin monolítico que funciona, a todos los efectos, como un modpack técnico. Añade reactores nucleares, sistemas de logística, magia, runas y máquinas eléctricas.

- **Mecanismo:** Slimefun utiliza cabezas de jugador personalizadas (Custom Skulls) y mapas de texturas aplicados a ítems base para crear "nuevos" bloques e ítems.
- **Compatibilidad Bedrock:** Este es el punto crítico. Normalmente, Bedrock no vería las texturas personalizadas de Slimefun, viendo solo cabezas genéricas. Sin embargo, mediante un **Resource Pack Converter** y la configuración adecuada en Geyser, podemos forzar que el cliente Bedrock descargue las texturas, permitiendo que vean las máquinas tal como son.⁷ Esto cumple el requisito de priorizar la experiencia visual de Bedrock incluso en un entorno modificado.

3. Requisitos de Hardware y Dimensionamiento del Servidor

Antes de proceder con la instalación lógica, es imperativo definir el hardware. Minecraft es una aplicación engañosamente pesada; aunque sus gráficos son simples, su lógica de

simulación (Ticks) es intensiva en CPU.

3.1 Análisis de Carga Computacional

Un servidor híbrido Slimefun + Geyser tiene tres fuentes principales de carga:

- El Loop Principal (Main Tick Loop):** Minecraft es predominantemente "Single-Threaded". Toda la lógica del mundo (movimiento de entidades, redstone, crecimiento de cultivos) ocurre en un solo hilo del procesador.
- La Sobre carga de Slimefun:** Slimefun añade miles de comprobaciones adicionales por tick. Cada máquina, cada red de energía y cada transportador de carga consume ciclos de CPU.
- La Traducción de Geyser:** Geyser debe traducir paquetes para cada jugador de Bedrock conectado. Aunque Geyser es multihilo, añade latencia y consumo de CPU, especialmente durante el movimiento rápido o carga de chunks.⁹

3.2 Especificaciones Recomendadas (Escenario: 10-20 Jugadores Simultáneos)

Componente	Requisito Mínimo	Requisito Óptimo	Justificación Técnica
CPU	Intel Core i5 (8va Gen) / Ryzen 5 3600	Ryzen 7 5800X / Intel i9 12900K	Se prioriza el rendimiento "Single-Core" (IPC). Más núcleos no ayudan significativamente al TPS (Ticks Per Second) del servidor principal, aunque sí ayudan a Geyser y a la generación de terreno asíncrona. ¹
RAM	8 GB DDR4	16 GB - 24 GB DDR4/DDR5	Slimefun es devorador de memoria. Un servidor vanilla funciona con 4GB, pero Slimefun +

			Geyser requiere al menos 10-12GB dedicados al contenedor para evitar que el "Garbage Collector" de Java se sature y cause micro-congelamientos. ¹¹
Almacenamiento	100 GB SSD SATA	250 GB NVMe SSD	La velocidad de lectura/escritura (I/O) es crítica para la carga de chunks. Los discos mecánicos (HDD) son inaceptables para servidores modernos. NVMe reduce drásticamente el lag al volar con Elytras.
Red	100 Mbps Simétrico	1 Gbps Fibra Óptica	Bedrock usa UDP, que es sensible a la pérdida de paquetes. Una conexión inestable causará que los jugadores de Bedrock vean "teletransportaciones" o desconexiones.

Nota sobre la Memoria: Asignar demasiada RAM (ej. 64GB) puede ser contraproducente en Java, ya que el ciclo de limpieza de memoria (Garbage Collection) tardará más tiempo cuando se ejecute, causando "Lag Spikes" masivos. El "Sweet Spot" para este tipo de servidor suele estar entre 12GB y 16GB.¹¹

4. Ingeniería de la Solución: Roadmap de Implementación

A continuación, se detalla el procedimiento técnico paso a paso para el despliegue. Este roadmap asume un sistema operativo anfitrión basado en Linux (Ubuntu 22.04 LTS o Debian 12), que es el entorno nativo recomendado para Docker.

4.1 Fase 1: Preparación del Entorno y Seguridad

Antes de desplegar contenedores, el host debe ser optimizado para cargas de trabajo de servidor de juegos.

1. **Ajuste de Swappiness:** Reducir la tendencia del kernel a usar memoria virtual (disco) en lugar de RAM. En /etc/sysctl.conf, establecer vm.swappiness=10. Minecraft en disco es Minecraft lento.
2. **Firewall y Puertos:**
 - TCP 25565: Entrada predeterminada para jugadores Java.
 - UDP 19132: Entrada crítica para jugadores Bedrock. **Advertencia:** Muchos paneles de control de hosting olvidan abrir el puerto UDP. Sin esto, Geyser no funcionará externamente.⁴

4.2 Fase 2: Definición de la Infraestructura Docker (Docker Compose)

El corazón de la instalación es el archivo docker-compose.yml. Este archivo define no solo el servidor, sino también los servicios auxiliares como la base de datos para logs y backups.

A continuación, se presenta una configuración altamente optimizada y comentada para satisfacer los requisitos del usuario:

YAML

```
version: "3.8"

services:
  # Servicio Principal: Minecraft Java con Paper
  mc:
    image: itzg/minecraft-server:java21 # Usar etiqueta específica de Java 21 para 1.20.6/1.21
    container_name: mc-hybrid-core
    hostname: mc-server
    restart: unless-stopped
```

```
ports:
  - "25565:25565"      # Puerto Java
  - "19132:19132/udp"   # Puerto Bedrock (Geyser)

environment:
  EULA: "TRUE"
  VERSION: "1.21.1"      # Versión estable recomendada para Slimefun actual
  TYPE: "PAPER"          # Paper ofrece optimizaciones críticas sobre Vanilla
  MEMORY: "12G"           # Asignación generosa para Slimefun

# Banderas de Aikar (Aikar's Flags): Optimizaciones de JVM estándar de la industria
# Estas reducen los parones del Garbage Collector.
  JVM_OPTS: "-XX:+UseG1GC -XX:+ParallelRefProcEnabled -XX:MaxGCPauseMillis=200
-XX:+UnlockExperimentalVMOptions -XX:+DisableExplicitGC -XX:+AlwaysPreTouch
-XX:G1NewSizePercent=30 -XX:G1MaxNewSizePercent=40 -XX:G1HeapRegionSize=8M
-XX:G1ReservePercent=20 -XX:G1HeapWastePercent=5 -XX:G1MixedGCCountTarget=4
-XX:InitiatingHeapOccupancyPercent=15 -XX:G1MixedGCLiveThresholdPercent=90
-XX:G1RSetUpdatingPauseTimePercent=5 -XX:SurvivorRatio=32 -XX:+PerfDisableSharedMem
-XX:MaxTenuringThreshold=1"

# Configuración de Dificultad y Juego
  DIFFICULTY: "hard"      # Necesario para el balance de "progresión lenta"
  VIEW_DISTANCE: 10        # Distancia de visión equilibrada
  SIMULATION_DISTANCE: 6   # Reducido para ahorrar CPU en granjas técnicas
  MAX_PLAYERS: 20
  MOTD: "§6Servidor Híbrido §bBedrock/Java §aSlimefun Survival"

# Auto-Instalación de Plugins Base desde Spiget (IDs de recursos)
# Nota: Slimefun suele requerir descarga manual de sus builds de desarrollo,
# pero instalamos los auxiliares aquí.
  SPIGET_RESOURCES: |
    51253, # GeyserMC (Verificar ID actual o usar URL directa)
    52119, # Floodgate
    28140, # LuckPerms (Gestión de permisos)
    81403, # Aurelum Skills (Sistema de Skills para progresión)
    19063, # EssentialsX
    34315, # WorldBorder (Gestión de fronteras)

volumes:
  -./minecraft-data:/data # Persistencia de datos del mundo y plugins

# Límites de recursos para estabilidad del host
deploy:
  resources:
    limits:
```

```

cpus: "4.0" # Limitar a 4 hilos lógicos
memory: 14G # Límite duro del contenedor (2GB overhead sobre la JVM)

# Servicio de Base de Datos (MariaDB)
# Esencial para plugins de registro como CoreProtect y LuckPerms
# Permite analizar griefing y gestionar permisos complejos
db:
  image: mariadb:10.6
  container_name: mc-database
  restart: unless-stopped
  environment:
    MYSQL_ROOT_PASSWORD: ${DB_ROOT_PASS}
    MYSQL_DATABASE: minecraft
    MYSQL_USER: minecraft
    MYSQL_PASSWORD: ${DB_PASS}
  volumes:
    -./db-data:/var/lib/mysql
  command: --max_connections=500

# Servicio de Backups Automatizados
backups:
  image: itzg/mc-backup
  container_name: mc-backup-service
  restart: unless-stopped
  environment:
    BACKUP_INTERVAL: "24h" # Backup diario
    PRUNE_BACKUPS_DAYS: 7 # Retener 7 días
    RCON_PASSWORD: ${RCON_PASSWORD} # Para forzar guardado antes del backup
  volumes:
    -./minecraft-data:/data:ro
    -./backups:/backups
  depends_on:
    - mc

```

Este archivo docker-compose establece una base sólida. Define el servidor de juego, una base de datos para el almacenamiento eficiente de logs (mucho más rápido que archivos planos XML/JSON) y un sistema de backups rotativos automáticos.

4.3 Fase 3: Integración de Slimefun (El "Modpack")

Slimefun no se encuentra siempre actualizado en Spiget, por lo que se recomienda instalar las "Dev Builds" (versiones de desarrollo) para máxima compatibilidad con las últimas versiones de Minecraft.

1. **Descarga:** Obtener Slimefun4.jar desde el repositorio oficial (GitHub o TheBusyBiscuit builds).
2. **Instalación:** Colocar el archivo manualmente en la carpeta generada ./minecraft-data/plugins/.
3. **Librería CS-CoreLib:** Slimefun solía requerir esta librería, aunque versiones modernas la integran. Verificar la documentación de la build específica.

Configuración Crítica de Slimefun para Rendimiento

En ./minecraft-data/plugins/Slimefun/config.yml, es vital realizar los siguientes ajustes para evitar lag:

- uridium-feature: false: Desactivar la generación automática de estructuras si el mundo ya está generado.
- research-difficulty-multiplier: 2.0: **Clave para la solicitud de "progresión lenta".** Esto duplica la cantidad de experiencia necesaria para desbloquear nuevas máquinas, obligando a los jugadores a jugar más tiempo para avanzar tecnológicamente.

4.4 Fase 4: La Capa de Interoperabilidad (Geyser y Floodgate)

Esta es la fase donde se cumple la prioridad de Bedrock.

1. **Configuración de Floodgate:**
 - Al iniciarse, Floodgate genera un archivo key.pem en su carpeta de configuración.
 - En plugins/Geyser-Spigot/config.yml, buscar la sección de autenticación y cambiar auth-type a floodgate. Esto permite que Geyser delegue la seguridad a Floodgate, permitiendo el acceso a usuarios de Xbox sin cuenta Java.⁵
2. **Mapeo de Puertos:** Asegurar que bedrock.port en la config de Geyser coincide con el puerto expuesto en Docker (19132).
3. **Optimización de Red:** En config.yml, establecer clone-remote-port: true si se desea mantener consistencia, aunque para Docker es mejor especificarlo manualmente.

5. Estrategia de Contenido: El Modpack "Slow Burn" con Slimefun

Para satisfacer la necesidad de un "modpack con muchas horas de juego, pero tardado de progresar", no basta con instalar Slimefun base. Debemos expandirlo con **Addons** que añadan capas de complejidad y tiempo.

La arquitectura modular de Slimefun permite añadir "expansiones" que funcionan igual que mods. A continuación, se presenta la selección curada de addons para este servidor, diseñada para maximizar la longevidad.

5.1 Selección de Addons para Progresión Extendida

Addon	Función en el Ecosistema	Impacto en la Curva de Progresión	Compatibilidad Bedrock (Geyser)
InfinityExpansion	Añade niveles "infinitos" de tecnología y herramientas.	Extremo. Requiere cantidades masivas de recursos para craftear el equipo final. Convierte el "End-Game" en un viaje de meses. ⁷	Requiere mapeo de texturas avanzado.
Galactifun	Exploración espacial y colonización de planetas.	Alto. Añade la necesidad de viajar a nuevas dimensiones para obtener recursos específicos (ej. Marte, Luna). Rompe la monotonía del Overworld. ⁷	Bueno, usa bloques vanilla retexturizados.
ExoticGarden	Agricultura, comida y cocina compleja.	Medio. Añade una dimensión horizontal a la progresión. Los jugadores deben buscar frutas y árboles por el mundo, fomentando la exploración. ¹²	Excelente, usa cabezas personalizadas.
DyeBench	Tintes y personalización estética.	Bajo (Cosmético). Permite a los constructores tener metas a largo plazo sin afectar el poder.	Bueno.

EcoPower	Generación de energía alternativa (Viento, Vapor).	Medio-Alto. Ofrece alternativas a los reactores nucleares, diversificando las rutas tecnológicas. ¹²	Bueno.
-----------------	--	--	--------

5.2 El Reto de las Texturas en Bedrock (Crucial)

Para priorizar Bedrock, **los jugadores deben ver las máquinas de Slimefun**. Por defecto, verán cabezas genéricas de "Steve" o bloques de barreras, lo cual arruina la experiencia.

Solución Implementada: Sincronización Automática de Resource Packs

1. **Generación:** Usar una herramienta automatizada o script (como los disponibles en la comunidad de Geyser) para convertir el Resource Pack de Slimefun (Java) a formato .mcpack (Bedrock).¹³
2. **Despliegue:**
 - Colocar el archivo .mcpack generado en plugins/Geyser-Spigot/packs/.
 - En la configuración de Geyser, no es necesario activar nada especial; Geyser detecta automáticamente los packs en esta carpeta y los "inyecta" al cliente Bedrock cuando se conecta.¹⁴
3. **Resultado:** Cuando un jugador de Xbox entra al servidor, verá una pantalla de "Descargando Recursos". Al terminar, verá las máquinas de Slimefun con sus texturas correctas, iconos en el inventario y modelos 3D, logrando una paridad casi total con la experiencia Java.

6. Estrategia de Balance y "Anti-Snowballing" (Progresión Lenta)

El usuario destacó un problema crítico en servidores SMP (Survival Multiplayer): la desmotivación causada por jugadores avanzados ("Speedrunners") que eclipsan a los nuevos. Para solucionar esto, diseñaremos un sistema de **Gating** (Restricción Escalonada) basado en tres pilares.

6.1 Pilar 1: Bloqueo Temporal de Dimensiones (Time-Gating)

La ruta más rápida al "End-Game" es matar al Dragón y obtener Elytras. Las Elytras trivializan la exploración y la recolección de recursos.

- **Implementación:** Usar el plugin **DimensionOpening** o configurar scripts de **Skript**.
- **Regla:**

- **Semana 1-2:** Solo Overworld accesible. Nether y End cerrados.
- **Semana 3:** Apertura del Nether.
- **Semana 5:** Apertura del End.
- **Efecto:** Obliga a toda la comunidad a desarrollar infraestructuras terrestres (caminos, trenes, barcos) durante el primer mes. Los jugadores avanzados no pueden "escapar" al End; deben convivir y comerciar en el Overworld.¹⁵

6.2 Pilar 2: Frontera de Mundo Dinámica (Geographic Gating)

Los jugadores avanzados suelen viajar miles de bloques para saquear todas las estructuras generadas (Mansiones, Templos marinos), dejando el mapa "seco" para los nuevos.

- **Implementación:** Plugin **ChunkyBorder** o scripts vanilla.
- **Configuración:**
 - Radio inicial: 2,000 bloques.
 - Expansión diaria: +50 bloques.
- **Efecto:** Cada día se genera terreno nuevo y virgen. Un jugador que entra al servidor 3 meses después del inicio tendrá acceso a recursos frescos en la nueva frontera. Esto democratiza la exploración.¹⁷

6.3 Pilar 3: Rendimientos Decrecientes en Habilidades (Skill Gating)

Utilizaremos **Aurelium Skills** para reemplazar la progresión basada puramente en equipo por una basada en el personaje.

- **Configuración "Slow Burn":**
 - En skills.yml, ajustar la fórmula de experiencia para que sea altamente exponencial.
 - Los primeros 20 niveles deben ser rápidos (enganche inicial).
 - Del nivel 20 al 50, la curva debe volverse muy empinada.
 - **Diminishing Returns:** Configurar las recompensas (ej. daño extra, probabilidad de doble drop) para que den menos beneficio por punto a niveles altos. Ejemplo: Nivel 1 da +1% de suerte, Nivel 50 da +0.1% de suerte. Esto evita que los jugadores veteranos sean matemáticamente invencibles.¹⁹

6.4 Pilar 4: Eliminación de la Inflación Económica

Minecraft Vanilla tiene una falla económica grave: los aldeanos. La curación de aldeanos zombies permite comprar armaduras de diamante por 1 esmeralda, rompiendo cualquier intento de progresión lenta.

- **Solución:** Configurar paper.yml para desactivar o limitar drásticamente los descuentos de curación (fix-curing-zombie-villager-discount-exploit: true).
- **Alternativa:** Usar el plugin **VillagerNerf** para deshabilitar el comercio de equipo de alto nivel, forzando a los jugadores a craftearlo usando las máquinas de Slimefun (lo cual consume energía y recursos, siendo mucho más lento y satisfactorio).

7. Optimización y Configuración Técnica Fina

Para garantizar que este sistema complejo funcione suavemente, se requieren ajustes precisos en los archivos de configuración.

7.1 Optimización de server.properties

Properties

```
view-distance=10
simulation-distance=6 # Importante: Mantener bajo. Slimefun carga muchos chunks.
network-compression-threshold=256
allow-flight=true    # Crítico: Bedrock envía paquetes de movimiento ligeramente distintos. Si está en
false, el servidor kickeará a jugadores de consola por "fly hack" en situaciones de lag.
difficulty=hard
enable-command-block=true
```

7.2 Optimización de bukkit.yml y spigot.yml

- **Spawn Limits:** Reducir ligeramente los límites de spawn de mobs. Slimefun añade sus propios mobs y máquinas, por lo que demasiados mobs vanilla saturarán la CPU.
- **Entity Activation Range:** Reducir distancias de activación para que los mobs lejanos no consuman CPU (AI processing).

7.3 Configuración de Geyser (config.yml)

- mtu: 1400: Ajustar si los jugadores de Bedrock experimentan desconexiones. Redes móviles a veces requieren MTU más bajo.
- show-cooldown: false: Bedrock no tiene indicador de cooldown de ataque en la UI. Desactivarlo evita confusión visual (aunque la mecánica de cooldown de Java sigue aplicando en el servidor).
- above-bedrock-nether-building: false: Bedrock permite construir sobre el techo del Nether, Java no (por defecto). Mantener en false para paridad.

8. Guía de Solución de Problemas (Troubleshooting) para Bedrock

Dada la prioridad en Bedrock, el administrador debe estar preparado para incidencias específicas de esta plataforma.

8.1 Problema: "Unable to Connect to World"

- **Causa:** Generalmente es un problema de puertos o versiones.
- **Solución:** Verificar que el contenedor Docker tiene el puerto 19132/udp mapeado correctamente. Verificar que Geyser se ha actualizado a la última build. Si Minecraft Bedrock recibió una actualización (ej. martes de parches), Geyser DEBE actualizarse inmediatamente. Configurar un reinicio automático diario con pull de la imagen Docker ayuda a mitigar esto.

8.2 Problema: Texturas Invisibles o Bloques "Update"

- **Causa:** El Resource Pack de Slimefun no se cargó o está desactualizado en el cliente.
- **Solución:** En el menú de servidor de Bedrock, el jugador debe ir a "Storage" -> "Cached Data" y borrar los packs del servidor para forzar una redescarga limpia al entrar. El administrador debe regenerar el .mcpack si Slimefun se actualiza.

8.3 Problema: Jugadores de Bedrock mueren por "Lag" al usar Elytras

- **Causa:** La predicción de movimiento de Bedrock y la validación de Java entran en conflicto a altas velocidades.
- **Solución:** Instalar el plugin **Hurricane** (anteriormente GeyserHacks). Este plugin aplica parches en el servidor para ser más indulgente con los movimientos de Bedrock, corrigiendo hitboxes y problemas con barcos/caballos.⁹

9. Conclusión

La arquitectura presentada en este reporte ofrece una solución integral al desafío de crear un servidor de Minecraft moderno, híbrido y equilibrado. Al combinar la robustez de **Docker** con la flexibilidad de **PaperMC**, la capacidad de traducción de **Geyser**, y la profundidad de contenido de **Slimefun**, se logra un producto final que respeta las restricciones técnicas de las consolas mientras ofrece la complejidad deseada por los veteranos.

La estrategia de **Progresión Lenta** mediante el bloqueo de dimensiones, la expansión dinámica de fronteras y el ajuste de curvas de experiencia asegura que el servidor mantenga su viabilidad y salud comunitaria a largo plazo, evitando el agotamiento prematuro del contenido. Este diseño no solo es viable técnicamente, sino que representa el estado del arte en la administración de comunidades de Minecraft heterogéneas.

Apéndice: Referencias de Configuración

Las citas y configuraciones referenciadas en este documento se basan en la documentación técnica de los proyectos involucrados y mejores prácticas de la industria.² Se recomienda revisar los repositorios oficiales de itzg/docker-minecraft-server y GeyserMC semanalmente para cambios en las variables de entorno o parámetros de red.

Obras citadas

1. Server/Requirements - Minecraft Wiki - Fandom, fecha de acceso: febrero 2, 2026, <https://minecraft.fandom.com/wiki/Server/Requirements>
2. docker-minecraft-server/examples/geyser/docker-compose.yml at master - GitHub, fecha de acceso: febrero 2, 2026, <https://github.com/itzg/docker-minecraft-server/blob/master/examples/geyser/docker-compose.yml>
3. Minecraft Server in Docker: Adulting Made Easy, fecha de acceso: febrero 2, 2026, https://serialized.net/2021/02/minecraft_server_docker/
4. Possible to use itzg/minecraft with GeyserMC? · Issue #213 - GitHub, fecha de acceso: febrero 2, 2026, <https://github.com/itzg/minecraft-server-charts/issues/213>
5. Examples - Minecraft Server on Docker (Java Edition), fecha de acceso: febrero 2, 2026, <https://docker-minecraft-server.readthedocs.io/en/latest/misc/examples/>
6. Integrate Floodgate Local Linking · Issue #3422 · itzg/docker-minecraft-server - GitHub, fecha de acceso: febrero 2, 2026, <https://github.com/itzg/docker-minecraft-server/issues/3422>
7. Addons · Slimefun/Slimefun4 Wiki - GitHub, fecha de acceso: febrero 2, 2026, <https://github.com/Slimefun/Slimefun4/wiki/Addons>
8. SofiaRedmond/Slimefun-Geyser-Extension - GitHub, fecha de acceso: febrero 2, 2026, <https://github.com/SofiaRedmond/Slimefun-Geyser-Extension>
9. Current Limitations - GeyserMC, fecha de acceso: febrero 2, 2026, <https://geysermc.org/wiki/geyser/current-limitations/>
10. Is there a performance difference between Java and Bedrock servers? : r/admincraft - Reddit, fecha de acceso: febrero 2, 2026, https://www.reddit.com/r/admincraft/comments/1lcjnot/is_there_a_performance_difference_between_java/
11. recommended specs for minecraft modded server : r/selfhosted - Reddit, fecha de acceso: febrero 2, 2026, https://www.reddit.com/r/selfhosted/comments/1m3hxba/recommended_specs_for.minecraft_modded_server/
12. Slimefun Ep. #22 - We are back with ALL THE ADD-ONS. - YouTube, fecha de acceso: febrero 2, 2026, <https://www.youtube.com/watch?v=UwJ4or9EaQ0>
13. FAQ · GeyserMC/Geyser Wiki · GitHub, fecha de acceso: febrero 2, 2026, <https://github.com/GeyserMC/Geyser/wiki/FAQ/74f485cea103ae6d05551e00357b24999b696d26>
14. Using Resource Packs with Geyser - GeyserMC, fecha de acceso: febrero 2,

- 2026, <https://geysermc.org/wiki/geyser/packs/>
- 15. DimensionOpening | SpigotMC - High Performance Minecraft Software, fecha de acceso: febrero 2, 2026,
<https://www.spigotmc.org/resources/dimensionopening.129752/>
 - 16. Plugin to disable access to End Dimension : r/admincraft - Reddit, fecha de acceso: febrero 2, 2026,
https://www.reddit.com/r/admincraft/comments/1dxpcxr/plugin_to_disable_acces_s_to_end_dimension/
 - 17. Minecraft but the world border increases based on the amount of players online - Reddit, fecha de acceso: febrero 2, 2026,
https://www.reddit.com/r/admincraft/comments/1l30zxy/minecraft_but_the_world_border_increases_based_on/
 - 18. Freedom Plugin - Dynamic World Border Management 1.21.X - GitHub, fecha de acceso: febrero 2, 2026, <https://github.com/wv3ntlyk/FreedomPlugin>
 - 19. Aurelium Skills Minecraft Plugin - Apex Hosting, fecha de acceso: febrero 2, 2026,
<https://apexminecrafthosting.com/guides/minecraft/plugins/aurelium-skills-minecrafter-plugin/>
 - 20. Skills | Aurelium Wiki, fecha de acceso: febrero 2, 2026,
<https://wiki.aurelium.dev/auraskills/skills/>
 - 21. Hurricane | GeyserMC, fecha de acceso: febrero 2, 2026,
<https://geysermc.org/wiki/other/hurricane/>