



Explicación Detallada de las 7R

Categoría 7R	Descripción	Cuándo usarlo	Ventajas	Consideraciones
Retirar (Retire)	Eliminar aplicaciones que ya no aportan valor o están duplicadas.	<ul style="list-style-type: none"> Aplicación obsoleta o sin soporte. Uso bajo o nulo. Funcionalidad duplicada. 	<ul style="list-style-type: none"> Reducción de costes. Simplificación del portafolio. Menos superficie de ataque. 	<ul style="list-style-type: none"> Verificar dependencias. Exportar o preservar datos históricos. Controlar impacto en procesos.
Retener (Retain)	Mantener la aplicación en su estado actual temporalmente.	<ul style="list-style-type: none"> Restricciones legales. Dependencias técnicas que impiden migración. Migrar supone más riesgo que beneficio. 	<ul style="list-style-type: none"> Evita migraciones prematuras. Permite planificar mejoras. Minimiza riesgos operativos. 	<ul style="list-style-type: none"> Incrementa deuda técnica. Requiere plan de salida. Puede limitar adopción cloud.
Reubicar / Rehospedar (Rehost - Lift & Shift)	Mover la aplicación a la nube sin modificarla.	<ul style="list-style-type: none"> Necesidad de migración rápida. Cierre de datacenter. Aplicación estable en su estado actual. 	<ul style="list-style-type: none"> Bajo riesgo. Migración rápida. Automatizable. 	<ul style="list-style-type: none"> No aprovecha funcionalidades cloud avanzadas. Puede mantener costes altos. Puede necesitar optimización posterior.

Categoría 7R	Descripción	Cuándo usarlo	Ventajas	Consideraciones
Replataformar (Replatform – Lift & Reshape)	Hacer ajustes menores para usar servicios cloud.	<ul style="list-style-type: none"> • Mejorar rendimiento. • Reducir costes operativos. • Sustituir componentes por servicios gestionados. 	<ul style="list-style-type: none"> • Beneficio alto con esfuerzo moderado. • Mejora estabilidad y escalabilidad. • Reduce mantenimiento. 	<ul style="list-style-type: none"> • Requiere pruebas. • Riesgo de incompatibilidades. • Dependencia de servicios específicos cloud.
Recomprar (Repurchase - SaaS)	Sustituir la aplicación por un SaaS del mercado.	<ul style="list-style-type: none"> • Existen soluciones SaaS maduras. • Coste de mantener la aplicación interna es alto. • Se busca estandarización. 	<ul style="list-style-type: none"> • Actualizaciones automáticas. • Escalabilidad. • Mantenimiento mínimo. 	<ul style="list-style-type: none"> • Adaptación de procesos al estándar SaaS. • Migración de datos crítica. • Costes de suscripción.
Refactorizar (Refactor)	Modificar código para modernizar sin rediseñar totalmente.	<ul style="list-style-type: none"> • Deuda técnica elevada. • Problemas de rendimiento. • Limitaciones de escalabilidad. 	<ul style="list-style-type: none"> • Mejora eficiencia. • Incrementa estabilidad. • Abre camino a cloud-native. 	<ul style="list-style-type: none"> • Requiere equipo experto. • Puede ser costoso. • No soluciona limitaciones estructurales profundas.
Rediseñar (Rearchitect)	Reconstruir totalmente la aplicación en cloud-native.	<ul style="list-style-type: none"> • Aplicación crítica. • Limitaciones severas en el diseño actual. • Necesidad de alta resiliencia y escalabilidad. 	<ul style="list-style-type: none"> • Máxima eficiencia cloud. • Escalabilidad extrema. • Reducción de costes a largo plazo. 	<ul style="list-style-type: none"> • Alto coste inicial. • Proyecto complejo. • Necesita perfiles avanzados.