

# COMODÍN IA - GUÍA DE RESOLUCIÓN DE PROBLEMAS

---

## ÍNDICE DE PROBLEMAS

---

1. [Problemas de Conexión y WhatsApp](#)
  2. [Errores de Autenticación y Usuarios](#)
  3. [Problemas de Inteligencia Artificial](#)
  4. [Errores de Pagos y Suscripciones](#)
  5. [Problemas de Base de Datos](#)
  6. [Errores del Sistema y Performance](#)
  7. [Problemas de SSL y Conectividad](#)
  8. [Herramientas de Diagnóstico](#)
- 

## METODOLOGÍA DE DIAGNÓSTICO

---

### Pasos Generales para Cualquier Problema

1. **Identificar:** ¿Qué exactamente no funciona?
2. **Logs:** Revisar logs relevantes
3. **Estado:** Verificar estado de servicios
4. **Recursos:** Verificar CPU, memoria, disco
5. **Acción:** Aplicar solución específica
6. **Validar:** Confirmar que el problema se resolvió

### Comandos de Diagnóstico Rápido

```
# Estado general del sistema
docker-compose ps
docker stats --no-stream
df -h
free -h

# Logs recientes de todos los servicios
docker-compose logs --tail=50

# Health check rápido
curl -s https://tudominio.com/api/health
```

## PROBLEMAS DE CONEXIÓN Y WHATSAPP

---

### PROBLEMA 1: Los clientes no pueden conectar WhatsApp por QR

#### Síntomas:

- El código QR no aparece

- El código QR aparece pero no funciona
- Error "QR Code expired" constantemente
- WhatsApp se desconecta frecuentemente

#### Diagnóstico:

```
# Verificar estado de Evolution API
docker-compose logs evolution-api --tail=100

# Verificar conectividad a Evolution API
curl -s https://tudominio.com/evolution/ | jq .

# Verificar instancias activas
curl -s -H "apikey: $EVOLUTION_API_KEY" \
https://tudominio.com/evolution/instance/fetchInstances
```

#### Soluciones:

##### Causa Probable: Evolution API no está corriendo

```
# Reiniciar Evolution API
docker-compose restart evolution-api

# Si no funciona, reiniciar con rebuild
docker-compose stop evolution-api
docker-compose up -d --force-recreate evolution-api

# Verificar logs
docker-compose logs -f evolution-api
```

##### Causa Probable: Problema de red o proxy

```
# Verificar configuración de nginx
docker-compose exec nginx nginx -t

# Reiniciar nginx si hay errores
docker-compose restart nginx

# Verificar que el puerto 8080 esté libre internamente
docker-compose exec app netstat -tlnp | grep 8080
```

##### Causa Probable: Base de datos de Evolution corrupta

```
# Limpiar datos de Evolution API
docker-compose stop evolution-api
docker volume rm comodin_ia_evolution_instances
docker volume rm comodin_ia_evolution_store
docker-compose up -d evolution-api
```

## PROBLEMA 2: Los mensajes de WhatsApp no se reciben en la plataforma

#### Síntomas:

- WhatsApp conectado pero mensajes no aparecen

- Delay excesivo en recepción de mensajes
- Mensajes aparecen duplicados

#### Diagnóstico:

```
# Verificar webhook de WhatsApp
docker-compose logs app --tail=100 | grep webhook

# Verificar cola de mensajes en Redis
docker-compose exec redis redis-cli KEYS "*whatsapp*"

# Verificar conexión a base de datos
docker-compose exec app npx prisma db seed --preview-feature
```

#### Soluciones:

##### Causa Probable: Webhook no está funcionando

```
# Verificar configuración de webhook en Evolution
curl -X GET \
  -H "apikey: $EVOLUTION_API_KEY" \
  https://tudominio.com/evolution/webhook/find/INSTANCE_NAME

# Reconfigurar webhook si es necesario
curl -X POST \
  -H "apikey: $EVOLUTION_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "url": "https://tudominio.com/api/webhooks/whatsapp",
    "enabled": true,
    "events": ["messages.upsert", "messages.update"]
  }' \
  https://tudominio.com/evolution/webhook/set/INSTANCE_NAME
```

##### Causa Probable: Sobrecarga del sistema

```
# Verificar uso de recursos
docker stats --no-stream

# Si memoria > 90%, reiniciar servicios
docker-compose restart app redis

# Limpiar cache de Redis
docker-compose exec redis redis-cli FLUSHDB
```



## ERRORES DE AUTENTICACIÓN Y USUARIOS

### PROBLEMA 3: Los usuarios no pueden hacer login

#### Síntomas:

- Error "Invalid credentials" con credenciales correctas
- Pantalla de login se recarga constantemente
- Error "Session expired" inmediatamente después del login

**Diagnóstico:**

```
# Verificar configuración de NextAuth
docker-compose logs app --tail=100 | grep -i "nextauth\|auth\|session"

# Verificar Redis (sesiones)
docker-compose exec redis redis-cli ping

# Verificar variables de entorno críticas
docker-compose exec app printenv | grep -E "NEXTAUTH|DATABASE"
```

**Soluciones:****Causa Probable: Problemas con sesiones en Redis**

```
# Limpiar sesiones en Redis
docker-compose exec redis redis-cli FLUSHALL

# Reiniciar Redis
docker-compose restart redis

# Verificar que Redis está funcionando
docker-compose exec redis redis-cli info server
```

**Causa Probable: Variable NEXTAUTH\_URL incorrecta**

```
# Verificar configuración
grep NEXTAUTH_URL .env

# Debe coincidir exactamente con tu dominio
# Correcto: NEXTAUTH_URL=https://tudominio.com
# Incorrecto: NEXTAUTH_URL=https://tudominio.com/

# Si está mal, corregir y reiniciar
nano .env
docker-compose restart app
```

**Causa Probable: Base de datos desactualizada**

```
# Ejecutar migraciones
docker-compose exec app npx prisma migrate deploy

# Verificar tablas de NextAuth
docker-compose exec postgres psql -U postgres -d comodin_ia -c "
SELECT tablename FROM pg_tables WHERE schemaname='public' AND tablename LIKE '%Ac-
count%';
"
```

**PROBLEMA 4: Invitaciones de usuarios no funcionan****Síntomas:**

- Emails de invitación no se envían
- Links de invitación expiran inmediatamente
- Error al aceptar invitaciones

**Diagnóstico:**

```
# Verificar tabla de invitaciones
docker-compose exec postgres psql -U postgres -d comodin_ia -c "
SELECT id, email, status, \"createdAt\" FROM \"Invitation\" ORDER BY \"createdAt\"
DESC LIMIT 10;
"

# Verificar logs de envío de email
docker-compose logs app --tail=100 | grep -i "mail\\|invitation"
```

### Soluciones:

#### Causa Probable: Configuración de email faltante

```
# Verificar variables SMTP en .env
grep SMTP .env

# Si no están configuradas, agregar:
echo "SMTP_HOST=smtp.gmail.com" >> .env
echo "SMTP_PORT=587" >> .env
echo "SMTP_USER=tu-email@gmail.com" >> .env
echo "SMTP_PASS=tu_password_app" >> .env

docker-compose restart app
```

#### Causa Probable: Invitaciones expiradas

```
# Limpiar invitaciones expiradas
docker-compose exec postgres psql -U postgres -d comodin_ia -c "
DELETE FROM \"Invitation\" WHERE \"createdAt\" < NOW() - INTERVAL '7 days';
"
```



## PROBLEMAS DE INTELIGENCIA ARTIFICIAL

### PROBLEMA 5: Las automatizaciones de IA no se ejecutan

#### Síntomas:

- IA no responde a mensajes
- Saldo de billetera no se debita
- Error "AI service unavailable"
- Respuestas de IA muy lentas (>30 segundos)

#### Diagnóstico:

```
# Verificar configuración de IA
docker-compose logs app --tail=100 | grep -i "ai\|abacus\|openai"

# Verificar saldo de billetera
docker-compose exec postgres psql -U postgres -d comodin_ia -c "
SELECT o.name, aw.balance, aw.\"updatedAt\"
FROM \"Organization\" o
JOIN \"AIWallet\" aw ON o.id = aw.\"organizationId\"
ORDER BY aw.\"updatedAt\" DESC;
"

# Verificar uso de IA reciente
docker-compose exec postgres psql -U postgres -d comodin_ia -c "
SELECT type, \"tokensUsed\", cost, \"createdAt\"
FROM \"AIUsage\"
ORDER BY \"createdAt\" DESC
LIMIT 10;
"
```

### Soluciones:

#### Causa Probable: Clave de IA expirada o inválida

```
# Verificar clave de Abacus AI
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $ABACUSAI_API_KEY" \
  -d '{"messages":[{"role":"user","content":"test"}]}' \
  https://api.abacus.ai/v1/chat/completions

# Si da error 401, actualizar clave en .env
nano .env
# Buscar: ABACUSAI_API_KEY=...
docker-compose restart app
```

#### Causa Probable: Billetera sin saldo

```
# Verificar saldo de organizaciones
docker-compose exec postgres psql -U postgres -d comodin_ia -c "
SELECT o.name, aw.balance
FROM \"Organization\" o
LEFT JOIN \"AIWallet\" aw ON o.id = aw.\"organizationId\"
ORDER BY o.\"createdAt\";
"

# Agregar saldo de prueba (solo para testing)
docker-compose exec postgres psql -U postgres -d comodin_ia -c "
UPDATE \"AIWallet\" SET balance = 100.00 WHERE balance < 10.00;
"
```

#### Causa Probable: Sobrecarga del servicio de IA

```
# Verificar latencia de API
time curl -X POST \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $ABACUSAI_API_KEY" \
  -d '{"messages":[{"role":"user","content":"test"}]}' \
  https://api.abacus.ai/v1/chat/completions

# Si tarda >10 segundos, configurar timeout más largo
# En app/lib/ai-service.ts ajustar timeout a 60000ms
```

## PROBLEMA 6: Análisis de sentimientos incorrecto

### Síntomas:

- Mensajes positivos marcados como negativos
- Análisis de sentimientos siempre “neutral”
- Cache de análisis no se actualiza

### Soluciones:

```
# Limpiar cache de análisis de IA
docker-compose exec postgres psql -U postgres -d comodin_ia -c "
DELETE FROM \"AIAAnalysisCache\" WHERE \"createdAt\" < NOW() - INTERVAL '24 hours';
"

# Limpiar cache en Redis
docker-compose exec redis redis-cli KEYS \"*ai_analysis*\" | xargs docker-compose exec r
edis redis-cli DEL

# Reiniciar servicios de IA
docker-compose restart app
```



## ERRORES DE PAGOS Y SUSCRIPCIONES

### PROBLEMA 7: Pagos con Stripe no se procesan

#### Síntomas:

- Error “Payment failed” con tarjetas válidas
- Webhooks de Stripe no se reciben
- Suscripciones no se actualizan después del pago

#### Diagnóstico:

```
# Verificar configuración de Stripe
docker-compose logs app --tail=100 | grep -i stripe

# Verificar webhooks de Stripe
curl -s https://tudominio.com/api/webhooks/stripe -X POST \
  -H "Content-Type: application/json" \
  -d '{"test": "webhook"}'

# Verificar transacciones recientes
docker-compose exec postgres psql -U postgres -d comodin_ia -c "
SELECT id, \"paymentProvider\", status, amount, \"createdAt\"
FROM \"Payment\"
ORDER BY \"createdAt\" DESC
LIMIT 10;
"
```

### Soluciones:

#### Causa Probable: Webhook secret incorrecto

```
# Verificar webhook secret en dashboard de Stripe
# Comparar con variable en .env
grep STRIPE_WEBHOOK_SECRET .env

# Si es diferente, actualizar y reiniciar
nano .env
docker-compose restart app
```

#### Causa Probable: Claves de Stripe incorrectas

```
# Verificar que las claves sean de producción (live_)
grep STRIPE_ .env

# Para testing, usar claves de test (sk_test_, pk_test_)
# Para producción, usar claves live (sk_live_, pk_live_)
```

## PROBLEMA 8: MercadoPago no funciona en algunos países

### Síntomas:

- Error "Payment method not available"
- Redirección a MercadoPago falla
- Pagos quedan en estado "pending" indefinidamente

### Soluciones:



```
# Verificar configuración de MercadoPago
docker-compose exec postgres psql -U postgres -d comodin_ia -c "
SELECT country, currency FROM \"Organization\" GROUP BY country, currency;
"

# MercadoPago solo funciona en: AR, BR, CL, CO, MX, PE, UY
# Para otros países, deshabilitar MercadoPago y usar solo Stripe

# Verificar webhook de MercadoPago
curl -s https://tudominio.com/api/webhooks/mercadopago -X POST \
-H "Content-Type: application/json" \
-d '{"test": "webhook"}'
```

## PROBLEMAS DE BASE DE DATOS

### PROBLEMA 9: Error “Database connection failed”

#### Síntomas:

- Error 500 en toda la aplicación
- “PrismaClientInitializationError”
- “Connection terminated unexpectedly”

#### Diagnóstico:

```
# Verificar estado de PostgreSQL
docker-compose ps postgres

# Verificar logs de base de datos
docker-compose logs postgres --tail=100

# Intentar conexión manual
docker-compose exec postgres psql -U postgres -d comodin_ia -c "SELECT NOW();"
```

#### Soluciones:

##### Causa Probable: PostgreSQL no está ejecutándose

```
# Reiniciar PostgreSQL
docker-compose restart postgres

# Si no inicia, verificar espacio en disco
df -h

# Si disco lleno, limpiar logs y cachés
docker system prune -f
docker-compose logs postgres > /tmp/postgres_logs.txt
docker-compose logs postgres --tail=0
```

##### Causa Probable: Demasiadas conexiones abiertas

```
# Verificar conexiones activas
docker-compose exec postgres psql -U postgres -d comodin_ia -c "
SELECT count(*) as active_connections FROM pg_stat_activity;
"

# Si >100 conexiones, reiniciar aplicación
docker-compose restart app

# Aumentar límite de conexiones (si es necesario)
docker-compose exec postgres psql -U postgres -c "
ALTER SYSTEM SET max_connections = 200;
SELECT pg_reload_conf();
"
```

## PROBLEMA 10: Migraciones de Prisma fallan

### Síntomas:

- Error al ejecutar `prisma migrate deploy`
- "Migration failed to apply cleanly"
- Tablas faltantes después de actualización

### Soluciones:

```
# Verificar estado de migraciones
docker-compose exec app npx prisma migrate status

# Si hay migraciones pendientes, aplicar manualmente
docker-compose exec app npx prisma migrate deploy --force

# Si hay conflictos, resetear migraciones (SOLO EN EMERGENCIA)
docker-compose exec app npx prisma migrate reset --force
docker-compose exec app npx prisma db seed
```

---

## ERRORES DEL SISTEMA Y PERFORMANCE

---

## PROBLEMA 11: Aplicación muy lenta o no responde

### Síntomas:

- Timeouts constantes (>30 segundos)
- CPU al 100% constantemente
- Memoria RAM agotada
- Error "Service Unavailable"

### Diagnóstico:

```
# Verificar recursos del sistema
docker stats --no-stream
free -h
df -h
top -n 1

# Verificar procesos que consumen más recursos
docker-compose exec app ps aux --sort=-%cpu | head -10

# Verificar logs de performance
docker-compose logs app --tail=100 | grep -i "slow\|timeout\|performance"
```

### Soluciones:

#### Causa Probable: Memoria insuficiente

```
# Reiniciar servicios para liberar memoria
docker-compose restart app redis

# Si el problema persiste, aumentar memoria del servidor
# 0 configurar límites de memoria en docker-compose.yml

# Verificar memory leaks en aplicación
docker-compose exec app node --expose-gc -e "
setInterval(() => {
  console.log(process.memoryUsage());
  global.gc && global.gc();
}, 5000);
"
```

#### Causa Probable: Base de datos lenta

```
# Verificar queries lentas en PostgreSQL
docker-compose exec postgres psql -U postgres -d comodin_ia -c "
SELECT query, mean_time, calls
FROM pg_stat_statements
ORDER BY mean_time DESC
LIMIT 10;
"

# Verificar índices faltantes
docker-compose exec app npx prisma db execute --stdin < migrations/add_indexes.sql

# Limpiar tablas de logs grandes
docker-compose exec postgres psql -U postgres -d comodin_ia -c "
DELETE FROM \"AIUsage\" WHERE \"createdAt\" < NOW() - INTERVAL '30 days';
VACUUM ANALYZE;
"
```

## PROBLEMA 12: “Out of disk space” o errores de espacio

### Síntomas:

- Error “No space left on device”
- Aplicación no puede escribir logs
- Backup fails con error de espacio

### Soluciones:

```
# Verificar uso de disco
df -h
du -sh /var/lib/docker/
du -sh ./

# Limpiar Docker
docker system prune -a -f
docker volume prune -f

# Limpiar logs antiguos
find . -name "*.log" -mtime +7 -delete
docker-compose logs app --tail=0

# Limpiar backups antiguos
find ./backups -name "*.gz" -mtime +30 -delete

# Limpiar archivos temporales
docker-compose exec app find /tmp -type f -mtime +1 -delete
```



## PROBLEMAS DE SSL Y CONECTIVIDAD

### PROBLEMA 13: “Your connection is not secure” o errores SSL

#### Síntomas:

- Navegador muestra advertencia de seguridad
- Error “NET::ERR\_CERT\_AUTHORITY\_INVALID”
- SSL certificate expired

#### Diagnóstico:

```
# Verificar certificado SSL
openssl x509 -in ./ssl/fullchain.pem -text -noout | grep -A 2 "Validity"

# Verificar configuración de nginx
docker-compose exec nginx nginx -t

# Test SSL externo
curl -I https://tudominio.com
```

#### Soluciones:

##### Causa Probable: Certificado expirado

```
# Renovar certificado con Let's Encrypt
docker-compose stop nginx
certbot renew --standalone
cp /etc/letsencrypt/live/tudominio.com/fullchain.pem ./ssl/
cp /etc/letsencrypt/live/tudominio.com/privkey.pem ./ssl/
docker-compose start nginx
```

##### Causa Probable: Configuración de nginx incorrecta

```
# Verificar configuración
docker-compose exec nginx nginx -t

# Si hay errores, verificar nginx.conf
nano nginx.conf

# Reiniciar nginx
docker-compose restart nginx
```

## PROBLEMA 14: Cannot connect to domain o DNS issues

### Síntomas:

- Error "This site can't be reached"
- DNS\_PROBE\_FINISHED\_NXDOMAIN
- Connection timeout

### Soluciones:

```
# Verificar resolución DNS
nslookup tudominio.com
dig tudominio.com

# Verificar que nginx esté escuchando en puertos correctos
netstat -tlnp | grep :80
netstat -tlnp | grep :443

# Verificar firewall
sudo ufw status
# Los puertos 80 y 443 deben estar abiertos

# Verificar registros DNS en tu proveedor
# A record: tudominio.com -> IP_DEL_SERVIDOR
# CNAME record: www.tudominio.com -> tudominio.com
```

---

## **HERRAMIENTAS DE DIAGNÓSTICO**

---

### **Script de Diagnóstico Automático**

```
#!/bin/bash
# Guardar como: diagnostico-completo.sh
# Uso: ./diagnostico-completo.sh

echo "=== COMODÍN IA - DIAGNÓSTICO COMPLETO ==="
echo "Fecha: $(date)"
echo "Servidor: $(hostname -f)"
echo ""

# 1. Estado de contenedores
echo "=== ESTADO DE CONTENEDORES ==="
docker-compose ps
echo ""

# 2. Uso de recursos
echo "=== USO DE RECURSOS ==="
echo "CPU y Memoria:"
docker stats --no-stream
echo ""
echo "Disco:"
df -h
echo ""

# 3. Conectividad
echo "=== PRUEBAS DE CONECTIVIDAD ==="
echo "Aplicación principal:"
curl -s -I https://tudominio.com | head -1
echo "API de salud:"
curl -s https://tudominio.com/api/health
echo "Evolution API:"
curl -s https://tudominio.com/evolution/ | head -1
echo ""

# 4. Base de datos
echo "=== ESTADO DE BASE DE DATOS ==="
echo "Conexión a PostgreSQL:"
docker-compose exec -T postgres psql -U postgres -d comodin_ia -c "SELECT NOW();" 2>/dev/null && echo "OK" || echo "ERROR"
echo "Número de usuarios:"
docker-compose exec -T postgres psql -U postgres -d comodin_ia -c "SELECT count(*) FROM \"User\";" 2>/dev/null | grep -E '^\\s*[0-9]+'
echo "Conexión a Redis:"
docker-compose exec -T redis redis-cli ping 2>/dev/null || echo "ERROR"
echo ""

# 5. SSL
echo "=== CERTIFICADO SSL ==="
openssl x509 -in ./ssl/fullchain.pem -noout -subject -dates 2>/dev/null || echo "No SSL certificate found"
echo ""

# 6. Logs recientes de errores
echo "=== ERRORES RECIENTES ==="
echo "Aplicación:"
docker-compose logs --tail=20 app 2>/dev/null | grep -i error | tail -5
echo "Evolution API:"
docker-compose logs --tail=20 evolution-api 2>/dev/null | grep -i error | tail -5
echo ""

# 7. Variables de entorno críticas
echo "=== CONFIGURACIÓN CRÍTICA ==="
echo "NEXTAUTH_URL: $(grep NEXTAUTH_URL .env | cut -d'=' -f2)"
```

```
echo "Database configured: $(grep DATABASE_URL .env >/dev/null && echo 'YES' || echo 'NO')"
```

```
echo "Stripe configured: $(grep STRIPE_SECRET_KEY .env >/dev/null && echo 'YES' || echo 'NO')"
```

```
echo "AI configured: $(grep ABACUSAI_API_KEY .env >/dev/null && echo 'YES' || echo 'NO')"
```

```
echo ""
```

```
echo "=== FIN DEL DIAGNÓSTICO ==="
```



## Script de Reinicio de Emergencia

```
#!/bin/bash
# Guardar como: reinicio-emergencia.sh
# Uso: ./reinicio-emergencia.sh

echo "=== REINICIO DE EMERGENCIA COMODÍN IA ==="
echo "ADVERTENCIA: Esto causará downtime de ~2-3 minutos"
echo "¿Continuar? (y/N)"
read -r response

if [[ "$response" =~ ^([yY][eE][sS]|[yY])$ ]]; then
    echo "Iniciando reinicio de emergencia..."

    # 1. Detener todos los servicios
    echo "Deteniendo servicios..."
    docker-compose down --remove-orphans

    # 2. Limpiar recursos
    echo "Limpiando recursos..."
    docker system prune -f

    # 3. Verificar disco
    echo "Verificando espacio en disco..."
    df -h | grep -E '^/dev' | awk '{if ($5 > "90%") print "ADVERTENCIA: Disco casi
llo en " $1 ": " $5}'

    # 4. Levantar servicios críticos primero
    echo "Iniciando base de datos..."
    docker-compose up -d postgres redis
    sleep 30

    # 5. Verificar BD
    echo "Verificando base de datos..."
    docker-compose exec postgres psql -U postgres -d comodin_ia -c "SELECT NOW();" >/
dev/null 2>&1
    if [ $? -eq 0 ]; then
        echo "Base de datos OK"
    else
        echo "ERROR: Base de datos no responde"
        exit 1
    fi

    # 6. Levantar aplicación
    echo "Iniciando aplicación..."
    docker-compose up -d

    # 7. Verificar estado final
    sleep 60
    echo "Verificando estado final..."
    docker-compose ps

    echo "=== REINICIO COMPLETADO ==="
    echo "Verificar funcionamiento en: https://tudominio.com"
else
    echo "Operación cancelada"
fi
```

## Comandos de Emergencia Rápida

```
# REINICIO RÁPIDO (< 30 segundos de downtime)
docker-compose restart

# REINICIO COMPLETO (2-3 minutos de downtime)
docker-compose down && docker-compose up -d

# LIMPIAR SOLO CACHE (sin downtime)
docker-compose exec redis redis-cli FLUSHALL

# BACKUP DE EMERGENCIA
docker-compose exec postgres pg_dump -U postgres comodin_ia | gzip >
emergency_backup_$(date +%Y%m%d_%H%M%S).sql.gz

# VERIFICAR SALUD DEL SISTEMA
curl -s https://tudominio.com/api/health && echo " - API OK" || echo " - API ERROR"
```

## MATRIZ DE ESCALAMIENTO

### Nivel 1 - Problemas Menores (Resolver internamente)

- Usuarios no pueden hacer login → Reiniciar servicio de autenticación
- WhatsApp se desconecta → Reiniciar Evolution API
- Página carga lentamente → Verificar recursos y reiniciar si es necesario

### Nivel 2 - Problemas Moderados (Considerar ayuda externa)

- Base de datos corrupta → Restaurar desde backup
- Certificados SSL expirados → Renovar certificados
- Múltiples servicios fallan → Reinicio completo del sistema

### Nivel 3 - Problemas Críticos (Contactar soporte técnico)

- Pérdida de datos sin backup reciente
- Servidor comprometido o hacked
- Problemas de infraestructura del proveedor de cloud

**NOTA CRÍTICA:** Antes de hacer cualquier cambio mayor, **SIEMPRE hacer un backup completo**. La mayoría de los problemas se pueden resolver con reiniciar servicios. Si un problema persiste después de 3 intentos diferentes, es momento de buscar ayuda especializada.