

Runbook de Incidentes - INMOVA

Guía de Respuesta a Incidentes

Este documento describe los procedimientos para diagnosticar y resolver incidentes comunes en INMOVA. Todos los tiempos están en minutos desde la detección del incidente.

Niveles de Severidad

Severidad 1 (Crítico)

- **Impacto:** Servicio completamente caído o pérdida de datos
- **Tiempo de respuesta:** 15 minutos
- **Tiempo de resolución objetivo:** 2 horas
- **Escalación:** Inmediata a CTO

Severidad 2 (Alto)

- **Impacto:** Funcionalidad crítica degradada (pagos, autenticación)
- **Tiempo de respuesta:** 30 minutos
- **Tiempo de resolución objetivo:** 4 horas
- **Escalación:** Si no se resuelve en 2 horas

Severidad 3 (Medio)

- **Impacto:** Funcionalidad no crítica afectada
- **Tiempo de respuesta:** 2 horas
- **Tiempo de resolución objetivo:** 24 horas
- **Escalación:** Si no se resuelve en 8 horas

Severidad 4 (Bajo)

- **Impacto:** Problema menor o cosmético
 - **Tiempo de respuesta:** Siguiendo día hábil
 - **Tiempo de resolución objetivo:** 1 semana
 - **Escalación:** No requiere
-

Procedimientos de Diagnóstico

Checklist Inicial (Todos los Incidentes)

- ☐ Confirmar el reporte del usuario
- ☐ Determinar severidad
- ☐ Verificar estado de servicios externos (Stripe, AWS, etc.)
- ☐ Revisar logs de servidor (Vercel/servidor)
- ☐ Revisar logs de aplicación (Sentry si configurado)
- ☐ Comprobar métricas de base de datos
- ☐ Identificar hora de inicio del problema
- ☐ Determinar número de usuarios afectados
- ☐ Crear ticket de incidente (si Sev 1-2)

Acceso a Logs

Logs de Aplicación (Servidor)

```
# Desarrollo
cd /home/ubuntu/homming_vidaro/nextjs_space
yarn dev
# Observar consola del servidor

# Producción (Vercel ejemplo)
vercel logs <deployment-url> --follow

# 0 en servidor propio
cd /opt/hostedapp/node/root/app
pm2 logs inmovia
```

Logs de Base de Datos

```
# PostgreSQL logs
sudo tail -f /var/log/postgresql/postgresql-14-main.log

# Queries lentas
SELECT * FROM pg_stat_statements
ORDER BY total_exec_time DESC
LIMIT 10;
```

Logs de Sentry (si configurado)

```
https://sentry.io/organizations/[org]/issues/
Filtrar por:
- Timestamp del incidente
- Severidad
- Endpoint afectado
```

Incidente: Aplicación Completamente Caída

Severidad: 1 (Crítico)

Síntomas

- Página no carga (500, 502, 503, 504)
- Timeout en todas las peticiones
- Error de conexión

Diagnóstico

1. Verificar infraestructura:

```
```bash
Verificar servidor web
curl -I https://inmovia.app
Si falla, servidor web caído

Verificar DNS
nslookup inmovia.app
Si falla, problema de DNS
```
```

1. Verificar proceso de Next.js:

```
bash
# En servidor de producción
pm2 status
# Si "stopped" o "errored", reiniciar
```

2. Verificar base de datos:

```
bash
# Conectar a PostgreSQL
psql -U inmovia_user -d inmovia -h localhost
# Si falla, BD caída
```

Resolución

Causa: Servidor Web Caído

```
# Reiniciar PM2 (si se usa)
pm2 restart inmovia

# O reiniciar servicio
sudo systemctl restart inmovia

# Verificar
curl -I https://inmovia.app
```

Causa: Base de Datos Caída

```
# Reiniciar PostgreSQL
sudo systemctl restart postgresql

# Verificar conexión
psql -U inmovia_user -d inmovia -h localhost

# Verificar integridad
SELECT pg_database_size('inmovia');
```

Causa: Memoria/CPU Agotada

```
# Verificar recursos
top
free -h
df -h

# Liberar memoria
sudo sync && sudo sysctl -w vm.drop_caches=3

# Si disco lleno, limpiar logs
sudo journalctl --vacuum-time=7d
sudo find /var/log -name "*.log" -mtime +30 -delete
```

Causa: Error de Despliegue

```
# Rollback a versión anterior (Vercel)
vercel rollback <deployment-id>

# 0 en servidor propio
cd /opt/hostedapp/node/root
rm -rf app
tar -xzf app_backup.tgz
pm2 restart inmovia
```

Comunicación

```
**Estado:** 🚨 Incidente Crítico
**Inicio:** [hora]
**Afectados:** Todos los usuarios
**Acción:** Investigando y trabajando en solución
**ETA:** [estimación]
```

Actualizaciones cada 15 minutos.

Incidente: Autenticación No Funciona

Severidad: 2 (Alto)

Síntomas

- Login falla con credenciales correctas
- “Invalid credentials” constante
- Redirección infinita en login
- Sesión no persiste

Diagnóstico

1. Verificar variables de entorno:

```
bash
# En servidor
env | grep NEXTAUTH
# Debe mostrar:
```

```
# NEXTAUTH_SECRET=...
# NEXTAUTH_URL=https://inmova.app
```

2. Verificar base de datos:

```
```sql
```

- Verificar que existen usuarios

```
SELECT id, email, name FROM "User" LIMIT 5;
```

- Verificar que passwords están hasheados

```
SELECT id, email, LENGTH(password) as pw_length FROM "User" LIMIT 5;
```

- pw\_length debe ser ~60 (bcrypt)

```
```
```

1. Verificar logs:

```
bash
```

```
# Buscar errores de autenticación
```

```
grep -i "nextauth" /var/log/inmova/app.log
```

```
grep -i "authentication" /var/log/inmova/app.log
```

Resolución

Causa: NEXTAUTH_URL Incorrecto

```
# Actualizar .env en producción
NEXTAUTH_URL="https://inmova.app"
```

```
# Reiniciar aplicación
pm2 restart inmova
```

Causa: NEXTAUTH_SECRET Faltante

```
# Generar nuevo secret
openssl rand -base64 32

# Agregar a .env
NEXTAUTH_SECRET="[secret generado]"

# Reiniciar
pm2 restart inmova
```

Causa: Problema de Cookies

```
// Verificar en navegador (DevTools > Application > Cookies)
// Debe existir: next-auth.session-token

// Si no existe, verificar configuración de cookies en lib/auth-options.ts
// Asegurar que:
cookies: {
  sessionToken: {
    name: `next-auth.session-token`,
    options: {
      httpOnly: true,
      sameSite: 'lax',
      path: '/',
      secure: process.env.NODE_ENV === 'production'
    }
  }
}
```

Causa: Base de Datos Desactualizada

```
# Ejecutar migraciones pendientes
cd /home/ubuntu/homming_vidaro/nextjs_space
yarn prisma migrate deploy
```

Workaround Temporal

Crear usuario de emergencia manualmente:

```
-- Generar password hasheado
-- En Node.js:
const bcrypt = require('bcryptjs');
const hash = await bcrypt.hash('password123', 10);
console.log(hash);

-- Insertar en BD
INSERT INTO "User" (id, email, name, password, role, "companyId", "createdAt", "updatedAt")
VALUES (
  gen_random_uuid(),
  'admin@inmova.com',
  'Admin Temporal',
  '[hash generado]',
  'superadmin',
  NULL,
  NOW(),
  NOW()
);
```



Incidente: Pagos con Stripe Fallan

Severidad: 2 (Alto)

Síntomas

- Botón de pago no responde

- Error “No such customer”
- Pago procesado pero no reflejado en INMOVA
- Webhook no se recibe

Diagnóstico

1. Verificar credenciales Stripe:

```
```bash
Verificar que existen
env | grep STRIPE

Debe mostrar:
STRIPE_SECRET_KEY=sk_live_...
STRIPE_PUBLISHABLE_KEY=pk_live_...
STRIPE_WEBHOOK_SECRET=whsec_...
```
```

1. Verificar Stripe Dashboard:

- <https://dashboard.stripe.com/payments>
- Buscar el payment intent
- Verificar estado (succeeded, failed, canceled)
- Revisar eventos relacionados

2. Verificar logs de webhooks:

- Dashboard > Developers > Webhooks > [endpoint]
- Ver intentos de entrega
- Verificar códigos de respuesta

Resolución

Causa: Customer No Existe

```
// Verificar en BD
SELECT id, "stripeCustomerId" FROM "Tenant" WHERE id = '[tenant_id]';

// Si stripeCustomerId es NULL, ejecutar:
// (desde API /api/stripe/sync-customers)
POST /api/stripe/sync-customers
```

Causa: Webhook No Configurado

```
# En Stripe Dashboard
# Developers > Webhooks > Add endpoint

URL: https://inmova.app/api/stripe/webhook
Eventos:
- payment_intent.succeeded
- payment_intent.payment_failed
- customer.subscription.created
- customer.subscription.updated
- invoice.payment_succeeded

# Copiar Signing Secret y actualizar .env
STRIPE_WEBHOOK_SECRET="whsec_..."
```

Causa: Modo Test vs Live

```
# Verificar que keys coinciden
# Si en producción, usar sk_live_ y pk_live_
# Si en desarrollo, usar sk_test_ y pk_test_

# NUNCA mezclar test y live
```

Causa: Sincronización Fallida

```
-- Actualizar pago manualmente
UPDATE "Payment"
SET estado = 'pagado',
    "fechaPago" = NOW(),
    "stripePaymentIntentId" = '[payment_intent_id]'
WHERE id = '[payment_id]';

-- Actualizar contrato si es el último pago
UPDATE "Contract"
SET estado = 'activo'
WHERE id = (SELECT "contractId" FROM "Payment" WHERE id = '[payment_id]');
```

Prevención

- Configurar alertas en Stripe para webhooks fallidos
- Implementar retry logic en webhook handler
- Logging detallado de todos los eventos de pago

Incidente: Error de Base de Datos

Severidad: 1-2 (depende del error)

Síntomas

- “Connection refused” o “Connection timeout”
- “too many clients”
- Queries extremadamente lentas
- “relation does not exist”

Diagnóstico

1. Verificar conexión:

```
bash
psql -U inmovia_user -d inmovia -h localhost
```

2. Verificar conexiones activas:

```
sql
SELECT count(*) FROM pg_stat_activity;
SELECT * FROM pg_stat_activity WHERE state = 'active';
```

3. Verificar queries lentas:

```
sql
SELECT pid, now() - query_start as duration, query
```



```
FROM pg_stat_activity
WHERE state = 'active'
ORDER BY duration DESC;
```

Resolución

Causa: Too Many Connections

```
-- Ver límite
SHOW max_connections;
-- Default: 100

-- Terminar conexiones idle
SELECT pg_terminate_backend(pid)
FROM pg_stat_activity
WHERE state = 'idle'
AND state_change < NOW() - INTERVAL '10 minutes';

-- Aumentar límite (temporal)
ALTER SYSTEM SET max_connections = 200;
SELECT pg_reload_conf();
```

Causa: Queries Lentas Bloqueando

```
-- Identificar queries bloqueantes
SELECT blocked.pid AS blocked_pid,
       blocked.query AS blocked_query,
       blocking.pid AS blocking_pid,
       blocking.query AS blocking_query
FROM pg_stat_activity AS blocked
JOIN pg_stat_activity AS blocking
  ON blocking.pid = ANY(pg_blocking_pids(blocked.pid));

-- Terminar query bloqueante (con precaución)
SELECT pg_terminate_backend([blocking_pid]);
```

Causa: Tabla/Índice Faltante

```
# Ejecutar migraciones
cd /home/ubuntu/homming_vidaro/nextjs_space
yarn prisma migrate deploy

# Si falla, verificar estado
yarn prisma migrate status

# Resetear (ATENCIÓN: BORRA DATOS)
yarn prisma migrate reset
```

Causa: Disco Lleno

```
# Verificar espacio
df -h

# Limpiar logs antiguos
sudo journalctl --vacuum-time=7d
sudo find /var/log -name "*.log" -mtime +30 -delete

# Vacuum de PostgreSQL
VACUUM FULL ANALYZE;
```

Recuperación de Desastres

Backup Manual

```
# Crear backup
pg_dump -U inmovia_user -d inmovia > backup_$(date +%Y%m%d_%H%M%S).sql

# Restaurar desde backup
psql -U inmovia_user -d inmovia < backup_20260115_120000.sql
```



Incidente: Emails No Se Envían

Severidad: 3 (Medio)

Síntomas

- Usuarios no reciben emails de confirmación
- Recordatorios de pago no llegan
- Emails van a spam

Diagnóstico

1. Verificar SendGrid:

```
```bash
Verificar API key
env | grep SENDGRID

Probar conexión
curl -X POST https://api.sendgrid.com/v3/mail/send \
-H "Authorization: Bearer $SENDGRID_API_KEY" \
-H "Content-Type: application/json" \
-d '{
 "personalizations": [{ "to": [{ "email": "test@example.com" }] }],
 "from": { "email": "noreply@inmovia.app" },
 "subject": "Test",
 "content": [{ "type": "text/plain", "value": "Test email" }]
}'
```
```

1. Verificar Activity en SendGrid:

- https://app.sendgrid.com/email_activity

- Buscar por email del destinatario
- Verificar estado (delivered, bounced, dropped)

Resolución

Causa: API Key Inválida

```
# Regenerar API key en SendGrid
# Settings > API Keys > Create API Key
# Actualizar .env
SENDGRID_API_KEY="SG.nuevo_key..."
```

Causa: Dominio No Autenticado

Settings > Sender Authentication > Authenticate Your Domain
 Seguir pasos para agregar registros **DNS**:

- CNAME para verificación
- TXT para SPF
- CNAME para DKIM

Causa: Emails en Spam

****Solución corto plazo:****

- Pedir a usuarios agregar noreply@inmova.app a contactos
- Añadir texto "Añadir a contactos" en emails

****Solución largo plazo:****

- Autenticar dominio completamente
- Calentar IP gradualmente
- Mejorar contenido de emails (menos palabras "spammy")
- Implementar List-Unsubscribe header



Incidente: Integraciones de Terceros Fallan

Severidad: 3 (Medio)

AWS S3

Síntomas

- "Access Denied" al subir archivos
- Imágenes no se muestran

Diagnóstico

```
# Verificar credenciales
env | grep AWS

# Probar acceso (con AWS CLI)
aws s3 ls s3://$AWS_BUCKET_NAME/$AWS_FOLDER_PREFIX
```

Resolución

```
# Verificar permisos del bucket
aws s3api get-bucket-policy --bucket $AWS_BUCKET_NAME

# Verificar IAM role
aws iam get-role --role-name innova-s3-access

# Si falta permiso, actualizar policy
```

Google Analytics

Síntomas

- No aparecen datos en GA4

Resolución

```
# Verificar Measurement ID
NEXT_PUBLIC_GA_MEASUREMENT_ID="G-XXXXXXXXXX"

# Verificar en navegador (DevTools > Network)
# Buscar peticiones a google-analytics.com/g/collect

# Si no hay peticiones, revisar script en app/layout.tsx
```



Incidente: Rendimiento Degradado

Severidad: 3 (Medio)

Síntomas

- Páginas cargan muy lento (>5 segundos)
- Timeouts ocasionales
- Dashboard tarda en cargar

Diagnóstico

1. Verificar recursos del servidor:

```
bash
top
free -h
iostat
```

2. Identificar queries lentas:

```
sql
SELECT query, mean_exec_time, calls
FROM pg_stat_statements
ORDER BY mean_exec_time DESC
LIMIT 10;
```

3. Verificar logs de Next.js:

```
bash
# Buscar warnings de performance
```

```
grep -i "slow" /var/log/inmova/app.log
grep -i "timeout" /var/log/inmova/app.log
```

Resolución

Causa: Queries N+1

```
// MALO (N+1)
const buildings = await prisma.building.findMany();
for (const building of buildings) {
  const units = await prisma.unit.findMany({ where: { buildingId: building.id } });
}

// BUENO (1 query)
const buildings = await prisma.building.findMany({
  include: { units: true }
});
```

Causa: Falta de Índices

```
-- Identificar queries que necesitan índices
SELECT * FROM pg_stat_user_tables
WHERE seq_scan > 1000
ORDER BY seq_scan DESC;

-- Crear índice (ejemplo)
CREATE INDEX idx_payment_estado_fecha
ON "Payment" (estado, "fechaVencimiento");
```

Causa: Caché No Configurado

```
// Implementar caching en APIs críticas
import { withCache, invalidateCache } from '@lib/api-cache-helpers';

// En GET
export const GET = withCache(
  async (req: Request) => { /* ... */ },
  { ttl: 300, key: 'dashboard-stats' }
);

// En POST/PUT/DELETE
invalidateCache(['dashboard-stats', 'buildings-list']);
```

Herramientas de Diagnóstico

Scripts de Utilidad

```
# /home/ubuntu/homming_vidaro/scripts/health-check.sh
#!/bin/bash
echo "=== Health Check ==="
echo "App Status:"
curl -I https://inmova.app
echo ""
echo "Database Status:"
psql -U inmova_user -d inmova -c "SELECT 1;"
echo ""
echo "Disk Usage:"
df -h | grep -E '(Filesystem|/$)'
echo ""
echo "Memory Usage:"
free -h
echo ""
echo "Active Connections:"
psql -U inmova_user -d inmova -c "SELECT count(*) FROM pg_stat_activity;"
```

Comandos Útiles

```
# Ver usuarios conectados
who

# Ver procesos de Node.js
ps aux | grep node

# Ver puertos abiertos
netstat -tulpn | grep LISTEN

# Ver logs en tiempo real
tail -f /var/log/inmova/app.log

# Reinicio rápido
pm2 restart inmova && pm2 logs inmova
```

Escalación y Contactos

Equipo de Guardia

Nivel 1 - Soporte Técnico

- Email: soporte@inmova.com
- Teléfono: +34 XXX XXX XXX
- Disponibilidad: L-V 9:00-18:00 CET

Nivel 2 - Ingeniería

- Email: dev@inmova.com
- Teléfono: +34 XXX XXX XXX
- Disponibilidad: 24/7 para Sev 1-2

Nivel 3 - CTO

- Email: cto@inmova.com
- Teléfono: +34 XXX XXX XXX
- Escalación: Sev 1 o Sev 2 sin resolución en 2h

Proveedores Externos**Stripe**

- Soporte: <https://support.stripe.com>
- Teléfono: +1 XXX XXX XXXX
- Dashboard: <https://dashboard.stripe.com>

AWS

- Soporte: <https://console.aws.amazon.com/support>
- Teléfono: Según plan de soporte

SendGrid

- Soporte: <https://support.sendgrid.com>
- Email: support@sendgrid.com

**Post-Mortem Template**

Después de cada incidente Sev 1-2, completar:

Post-Mortem: [Título del Incidente]**## Resumen**

- ****Fecha:**** [fecha]
- ****Duración:**** [XX minutos/horas]
- ****Severidad:**** [1-4]
- ****Usuarios afectados:**** [número o porcentaje]

Timeline

- [HH:MM] - Incidente detectado
- [HH:MM] - Diagnóstico iniciado
- [HH:MM] - Causa raíz identificada
- [HH:MM] - Solución aplicada
- [HH:MM] - Verificación completada
- [HH:MM] - Incidente resuelto

Causa Raíz

[Descripción detallada de qué causó el problema]

Impacto

- Usuarios afectados: [número]
- Funcionalidades afectadas: [lista]
- Pérdida de datos: [sí/no, detalles]
- Impacto financiero: [estimación]

Resolución

[Pasos específicos que se tomaron para resolver]

Lecciones Aprendidas**### Qué Funcionó Bien**

- [punto 1]
- [punto 2]

Qué No Funcionó Bien

- [punto 1]
- [punto 2]

Acciones Preventivas

1. [] [Acción 1] - Responsable: [nombre] - Fecha: [fecha]
2. [] [Acción 2] - Responsable: [nombre] - Fecha: [fecha]
3. [] [Acción 3] - Responsable: [nombre] - Fecha: [fecha]



Checklist de Prevención

Diaria

- ☐ Revisar logs de errores
- ☐ Verificar métricas de rendimiento
- ☐ Comprobar espacio en disco
- ☐ Verificar backups automáticos

Semanal

- ☐ Revisar alertas de Sentry
- ☐ Analizar queries lentas

- ☐ Verificar integraciones externas
- ☐ Revisar logs de webhooks

Mensual

- ☐ Actualizar dependencias de seguridad
- ☐ Revisar certificados SSL (expiración)
- ☐ Auditar usuarios y permisos
- ☐ Verificar y limpiar datos antiguos
- ☐ Prueba de restauración de backup

Mantenido por: Enxames Investments SL - Equipo de Ingeniería

Última actualización: 15 de Enero de 2026

Versión: 2.0