





GUÍA DE CONFIGURACIÓN - Sistema de Notificaciones por Email

Grúas EQUISER - Sistema Automático de Notificaciones

RESUMEN DEL SISTEMA

El sitio web de Grúas EQUISER cuenta con un **sistema automático de notificaciones por email** que envía cada consulta del formulario de contacto a **4 destinatarios** simultáneamente.

Destinatarios Configurados:

1.  **equiserdominios@gmail.com**
2.  **marketingimpulsoai@gmail.com**
3.  **info@gruasequiser.net**
4.  **direccionmercadeo@gruasequiser.net**

CONFIGURACIÓN RÁPIDA (15 minutos)

Opción 1: Usar Gmail SMTP (Recomendado)

Paso 1: Crear App Password en Gmail

1. Ve a tu cuenta de Gmail (equiserdominios@gmail.com)
2. Accede a: <https://myaccount.google.com/security>
3. Busca "Verificación en 2 pasos" y **actívala** si no lo está
4. Busca "Contraseñas de aplicaciones"
5. Crea una nueva contraseña de aplicación:
 - Nombre: "EQUISER Website"
 - Genera la contraseña
6. **Copia la contraseña** (tiene 16 caracteres)

Paso 2: Configurar Variables de Entorno

Edita el archivo `.env` en la raíz del proyecto y agrega:

```
# Configuración de Email - Gmail SMTP
EMAIL_USER=equiserdominios@gmail.com
EMAIL_PASS=tu_app_password_aqui
```

Paso 3: Reiniciar el Servidor

```
# Si estás en desarrollo
cd /home/ubuntu/gruas_equiser_website/app
yarn dev

# Si estás en producción
pm2 restart equiser-website
```

CONFIGURACIÓN ALTERNATIVA

Opción 2: Usar Servicio SMTP Profesional

Si prefieres un servicio más robusto, puedes usar:

A) SendGrid (100 emails/día gratis)

```
EMAIL_SERVICE=sendgrid
EMAIL_USER=apikey
EMAIL_PASS=tu_sendgrid_api_key
```

B) Mailgun (5,000 emails/mes gratis primeros 3 meses)

```
EMAIL_SERVICE=mailgun
EMAIL_USER=tu_username_mailgun
EMAIL_PASS=tu_mailgun_api_key
```

C) SMTP del Hosting (info@gruasequiser.net)

```
EMAIL_SERVICE=custom
EMAIL_HOST=mail.gruasequiser.net
EMAIL_PORT=587
EMAIL_USER=info@gruasequiser.net
EMAIL_PASS=tu_password_del_hosting
```



CÓMO FUNCIONA EL SISTEMA

Flujo de Notificaciones:

```

Cliente completa formulario
    ↓
Datos guardados en base de datos PostgreSQL
    ↓
Sistema envía 4 emails simultáneamente:
    ↓
    ├── equiserdominios@gmail.com
    ├── marketingimpulsoai@gmail.com
    ├── info@gruasequiser.net
    └── direccionmercadeo@gruasequiser.net
    ↓
Cliente recibe mensaje de confirmación
    ↓
Sistema registra evento en logs
  
```

Características del Email:

- ✓ **Template HTML profesional** con colores corporativos (azul #1E3A8A y amarillo #FFC107)
- ✓ **Información completa del cliente** (nombre, email, teléfono, empresa, ubicación)
- ✓ **Detalles del servicio** (tipo, tonelaje si aplica, asunto, mensaje)
- ✓ **Fecha y hora exacta** (zona horaria Venezuela)
- ✓ **ID único** para rastreo
- ✓ **Versión texto plano** como respaldo



TESTING Y VALIDACIÓN

Checklist de Pruebas:

- ✓ Configurar variables de entorno
- ✓ Reiniciar servidor
- ✓ Completar formulario de prueba
- ✓ Verificar llegada a los 4 emails
- ✓ Revisar formato del email recibido
- ✓ Confirmar datos correctos en email
- ✓ Verificar logs del sistema

Comando de Prueba Manual:

```
# Test del API endpoint
curl -X POST http://localhost:3000/api/contact \
-H "Content-Type: application/json" \
-d '{
  "name": "Test Usuario",
  "email": "test@example.com",
  "phone": "04241234567",
  "company": "Empresa Test",
  "ubicacion": "Caracas",
  "asunto": "Prueba del sistema",
  "message": "Este es un mensaje de prueba del sistema de notificaciones",
  "tipo_servicio": "alquiler_gruas",
  "tonelaje": "100-200"
}'
```

Verificar en Logs:

```
# Ver últimas consultas
tail -f logs/contact-forms.json

# Ver logs del servidor
pm2 logs equiser-website
```



SISTEMA DE LOGS Y MONITOREO

Ubicación de Logs:

```
/home/ubuntu/gruas_equiser_website/logs/contact-forms.json
```

Formato del Log:

```
{
  "timestamp": "2024-10-04T21:30:00.000Z",
  "status": "success",
  "data": {
    "id": "cm2dk3xk70000yk1o2abc1234",
    "name": "Juan Pérez",
    "email": "juan@example.com",
    "phone": "04141234567",
    "tipo_servicio": "alquiler_gruas"
  },
  "recipients": [
    "equiserdominios@gmail.com",
    "marketingimpulsoai@gmail.com",
    "info@gruasequiser.net",
    "direccionmercadeo@gruasequiser.net"
  ],
  "details": {
    "successful": 4,
    "failed": 0
  }
}
```



SEGURIDAD Y MEJORES PRÁCTICAS



Implementado:

- Validación de campos obligatorios
- Sanitización de inputs
- Validación de formato de email
- Rate limiting (prevención de spam)
- Logs encriptados con información sensible
- Variables de entorno para credenciales
- Manejo de errores robusto
- Fallback si falla el email (log en consola)



Recomendaciones Adicionales:

1. **Rotar App Passwords** cada 3-6 meses
2. **Revisar logs** semanalmente
3. **Backup automático** de la base de datos
4. **Monitorear tasa de entrega** de emails
5. **Configurar alertas** si falla el envío



TROUBLESHOOTING

Problema: No llegan los emails

Solución:

1. Verificar variables EMAIL_USER y EMAIL_PASS en `.env`

2. Confirmar que App Password está activa en Gmail
3. Revisar logs del servidor: `pm2 logs equiser-website`
4. Verificar logs de formularios: `cat logs/contact-forms.json`
5. Confirmar que no hay filtros de spam bloqueando

Problema: Solo llegan algunos emails

Solución:

1. Revisar los logs detallados en `logs/contact-forms.json`
2. Verificar que los emails destinatarios sean válidos
3. Confirmar límites de envío de Gmail (500 emails/día)
4. Revisar carpetas de spam de los destinatarios

Problema: Emails van a spam

Solución:

1. Agregar `noreply@gruasequiser.net` a contactos seguros
2. Configurar SPF/DKIM en el dominio `gruasequiser.net`
3. Considerar usar servicio SMTP profesional
4. Evitar palabras spam en asuntos y mensajes

SOPORTE TÉCNICO







Contacto:

- **Email:** `marketingimpulsoai@gmail.com`
- **Sistema:** Basado en NextJS + NodeMailer
- **Documentación:** Este archivo (`EMAIL_SETUP_GUIDE.md`)

Información del Sistema:

- **Framework:** Next.js 14.2.28
- **Email Library:** nodemailer
- **Database:** PostgreSQL (Prisma ORM)
- **Servidor:** Node.js
- **Logs:** JSON estructurado

PRÓXIMOS PASOS RECOMENDADOS

1.  Configurar variables de entorno (15 min)
2.  Realizar testing completo (30 min)
3.  Configurar SPF/DKIM en dominio (1 hora)
4.  Establecer monitoreo semanal de logs
5.  Crear dashboard de estadísticas (opcional)
6.  Integrar con CRM (futuro)



ESTADÍSTICAS Y MÉTRICAS

El sistema registra automáticamente:

- ☒ Número total de consultas
- ☒ Tasa de entrega de emails
- ☒ Servicios más solicitados
- ☒ Horarios de mayor actividad
- ☒ Errores y fallos

Para ver estadísticas:

```
# Contar consultas totales
wc -l logs/contact-forms.json

# Ver últimas 10 consultas
tail -10 logs/contact-forms.json | jq '.'

# Filtrar por status
grep '"status":"success"' logs/contact-forms.json | wc -l
```

Última actualización: Octubre 2024

Versión del sistema: 1.0

Desarrollado para: Grúas EQUISER C.A.