

Configuración de Tareas Cron - INMOVA

Resumen Ejecutivo

Este documento detalla la configuración de las tareas programadas (cron jobs) para las automatizaciones de INMOVA. Las tareas automatizadas incluyen:

- 1. **Renovación de Contratos:** Alertas progresivas para contratos próximos a vencer
- 2. **Recordatorios de Pago:** Notificaciones escalonadas para pagos pendientes
- 3. **Mantenimiento Preventivo:** Avisos de tareas programadas próximas

Configuración de Seguridad

Variables de Entorno

En el archivo `.env`, asegúrate de tener configurada la variable:

```
CRON_SECRET=tu_token_secreto_muy_seguro_aqui
```

Importante: Genera un token seguro para producción. Puedes usar:

```
node -e "console.log(require('crypto').randomBytes(32).toString('hex'))"
```

Endpoints Disponibles

1. Renovación de Contratos

- **URL:** `/api/cron/process-contract-renewals`
- **Métodos:** GET, POST
- **Frecuencia Recomendada:** Diaria (6:00 AM)
- **Descripción:** Procesa contratos próximos a vencer y envía alertas progresivas según etapas:
- **90-61 días:** Alerta inicial (planificación)
- **60-31 días:** Seguimiento (evaluación)
- **30-16 días:** Urgente (acción requerida)
- **15-0 días:** Crítico (acción inmediata)

2. Recordatorios de Pago

- **URL:** `/api/cron/process-payment-reminders`
- **Métodos:** GET, POST
- **Frecuencia Recomendada:** Diaria (7:00 AM)
- **Descripción:** Identifica pagos pendientes y envía recordatorios progresivos:
- **3-6 días:** Recordatorio amigable

- **7-14 días:** Recordatorio formal
- **15-29 días:** Aviso urgente
- **30+ días:** Notificación legal

3. Mantenimiento Preventivo

- **URL:** `/api/cron/process-preventive-maintenance`
- **Métodos:** GET, POST
- **Frecuencia Recomendada:** Diaria (8:00 AM)
- **Descripción:** Notifica sobre tareas de mantenimiento programadas próximas (próximos 7 días):
- **1 día:** Prioridad ALTA (urgente)
- **2-3 días:** Prioridad ALTA (importante)
- **4-7 días:** Prioridad MEDIA

Métodos de Configuración

Opción 1: Vercel Cron Jobs (Recomendado para Vercel)

Si tu aplicación está desplegada en Vercel, crea un archivo `vercel.json` en la raíz del proyecto:

```
{
  "crons": [
    {
      "path": "/api/cron/process-contract-renewals",
      "schedule": "0 6 * * *"
    },
    {
      "path": "/api/cron/process-payment-reminders",
      "schedule": "0 7 * * *"
    },
    {
      "path": "/api/cron/process-preventive-maintenance",
      "schedule": "0 8 * * *"
    }
  ]
}
```

Nota: Vercel Cron está disponible en planes Pro y Enterprise.

Opción 2: Cron-job.org (Gratuito)

1. Regístrate en cron-job.org (<https://cron-job.org>)
2. Crea un nuevo cronjob para cada endpoint:
 - **URL:** `https://tu-dominio.com/api/cron/process-contract-renewals`
 - **Schedule:** `0 6 * * *` (diario a las 6:00 AM)
 - **Headers:** Añadir `Authorization: Bearer TU_CRON_SECRET`
3. Repite para los otros dos endpoints.

Opción 3: GitHub Actions

Crea `.github/workflows/cron-jobs.yml` :

```

name: INMOVA Cron Jobs

on:
  schedule:
    # Renovación de contratos - Diario a las 6:00 AM UTC
    - cron: '0 6 * * *'
    # Recordatorios de pago - Diario a las 7:00 AM UTC
    - cron: '0 7 * * *'
    # Mantenimiento preventivo - Diario a las 8:00 AM UTC
    - cron: '0 8 * * *'
  # Permitir ejecución manual
  workflow_dispatch:

jobs:
  run-cron-jobs:
    runs-on: ubuntu-latest
    steps:
      - name: Renovación de Contratos
        run: |
          curl -X POST "https://tu-dominio.com/api/cron/process-contract-renewals" \
            -H "Authorization: Bearer ${ secrets.CRON_SECRET }"

      - name: Recordatorios de Pago
        run: |
          curl -X POST "https://tu-dominio.com/api/cron/process-payment-reminders" \
            -H "Authorization: Bearer ${ secrets.CRON_SECRET }"

      - name: Mantenimiento Preventivo
        run: |
          curl -X POST "https://tu-dominio.com/api/cron/process-preventive-
maintenance" \
            -H "Authorization: Bearer ${ secrets.CRON_SECRET }"

```

Configuración:

1. Añade `CRON_SECRET` en GitHub Settings > Secrets > Actions
2. Ajusta el horario según tu zona horaria

Opción 4: Servidor Linux con Crontab

Edita el crontab:

```
crontab -e
```

Añade las siguientes líneas:

```

# INMOVA - Renovación de Contratos (6:00 AM)
0 6 * * * curl -X POST "https://tu-dominio.com/api/cron/process-contract-renewals" -H
"Authorization: Bearer TU_CRON_SECRET"

# INMOVA - Recordatorios de Pago (7:00 AM)
0 7 * * * curl -X POST "https://tu-dominio.com/api/cron/process-payment-reminders" -H
"Authorization: Bearer TU_CRON_SECRET"

# INMOVA - Mantenimiento Preventivo (8:00 AM)
0 8 * * * curl -X POST "https://tu-dominio.com/api/cron/process-preventive-mainten-
ance" -H "Authorization: Bearer TU_CRON_SECRET"

```

Pruebas en Desarrollo

Prueba Manual de Endpoints

Puedes probar cada endpoint manualmente usando curl:

```
# Con autenticación
curl -X POST "http://localhost:3000/api/cron/process-contract-renewals" \
  -H "Authorization: Bearer $(grep CRON_SECRET .env | cut -d '=' -f2)"

# Sin autenticación (si CRON_SECRET no está configurado)
curl -X POST "http://localhost:3000/api/cron/process-contract-renewals"
```

Endpoint de Prueba

También puedes crear un endpoint de desarrollo:

```
// app/api/test/cron/route.ts
import { NextRequest, NextResponse } from 'next/server';

export async function GET(request: NextRequest) {
  const results = {
    contractRenewals: await fetch(`${process.env.NEXT_PUBLIC_APP_URL}/api/cron/process-contract-renewals`, {
      headers: { 'Authorization': `Bearer ${process.env.CRON_SECRET}` }
    }).then(r => r.json()),
    paymentReminders: await fetch(`${process.env.NEXT_PUBLIC_APP_URL}/api/cron/process-payment-reminders`, {
      headers: { 'Authorization': `Bearer ${process.env.CRON_SECRET}` }
    }).then(r => r.json()),
    preventiveMaintenance: await fetch(`${process.env.NEXT_PUBLIC_APP_URL}/api/cron/process-preventive-maintenance`, {
      headers: { 'Authorization': `Bearer ${process.env.CRON_SECRET}` }
    }).then(r => r.json())
  };

  return NextResponse.json(results);
}
```

Monitoreo y Logs

Logs de Ejecución

Los endpoints están configurados para registrar:

- Inicio de ejecución
- Número de notificaciones procesadas
- Errores encontrados
- Tiempo de ejecución

Respuestas de Éxito

```
{
  "success": true,
  "message": "Alertas de renovación procesadas correctamente",
  "timestamp": "2025-01-15T06:00:00.000Z",
  "results": {
    "total": 15,
    "notificadas": 15,
    "errores": 0,
    "detalles": [...]
  }
}
```

Monitoreo Recomendado

1. **Uptime Monitoring:** Configura alertas si los endpoints fallan
2. **Log Aggregation:** Usa servicios como Datadog, Sentry, o LogRocket
3. **Email Notifications:** Configura alertas por email para errores críticos

Troubleshooting

Error 401: No Autorizado

Causa: Token CRON_SECRET incorrecto o no enviado.

Solución: Verifica que el header `Authorization: Bearer TU_TOKEN` esté correcto.

Error 500: Error Interno

Causa: Error en la lógica de procesamiento o en la base de datos.

Solución: Revisa los logs del servidor para detalles específicos.

No Se Envían Emails

Causa: Configuración SMTP incorrecta o emails no válidos.

Solución: Verifica las variables de entorno SMTP y que `emailContacto` esté configurado en las empresas.



Horarios Recomendados (Hora España - CET/CEST)

- **6:00 AM:** Renovación de Contratos
- **7:00 AM:** Recordatorios de Pago
- **8:00 AM:** Mantenimiento Preventivo

Nota: Ajusta los horarios UTC en tu configuración de cron:

- Invierno (CET = UTC+1): Resta 1 hora
- Verano (CEST = UTC+2): Resta 2 horas

Ejemplo para 6:00 AM CET (invierno):

```
0 5 * * * # UTC
```

Checklist de Implementación

- ☐ Configurar `CRON_SECRET` en `.env`
 - ☐ Configurar variables SMTP para envío de emails
 - ☐ Elegir y configurar el sistema de cron (Vercel, cron-job.org, GitHub Actions, etc.)
 - ☐ Probar cada endpoint manualmente
 - ☐ Verificar que se envíen las notificaciones
 - ☐ Configurar monitoreo de logs
 - ☐ Documentar horarios configurados
 - ☐ Informar al equipo sobre las automatizaciones activas
-

Soporte

Para dudas o problemas con la configuración de cron jobs, contacta al equipo de desarrollo de IN-MOVA.

Fecha: Diciembre 2025

Versión: 1.0

Última Actualización: 4 de diciembre de 2025