

Guía de Gestión de APIs - Clic Inmobiliaria

Arquitectura Actual

Servicios Corriendo

- **API Gateway** → Puerto 8080 (`/opt/api-gateway/`)
- **API Principal** → Puerto 3001 (`/opt/email-service/`)
- **API Imágenes** → Puerto 3002 (`/opt/image-compression-api/`)
- **VS Code Server** → Puerto 8443

URLs de Acceso

- **Dashboard Principal:** `https://dev.api.clicinmobiliaria.com/`
 - **API de Imágenes:** `https://dev.api.clicinmobiliaria.com/images/`
 - **VS Code Editor:** `https://dev.api.clicinmobiliaria.com/ide/`
 - **API Principal:** `https://api.clicinmobiliaria.com/`
-

Crear Nueva API

1. Crear Estructura de Carpetas

```
bash
cd /opt
mkdir nueva-api-nombre
cd nueva-api-nombre
```

2. Inicializar Proyecto Node.js

```
bash
npm init -y
npm install express cors helmet morgan
```

3. Crear archivo server.js

```
javascript
```

```
const express = require('express');
const cors = require('cors');
const helmet = require('helmet');
const morgan = require('morgan');

const app = express();
const port = 3003; // Usar siguiente puerto disponible

// Middlewares
app.use(helmet());
app.use(cors());
app.use(morgan('combined'));
app.use(express.json({ limit: '10mb' }));

// Ruta principal
app.get('/', (req, res) => {
  res.json({
    service: "Nombre de tu API",
    status: "online",
    version: "1.0.0",
    endpoints: {
      // Listar endpoints disponibles
    }
  });
});

// Tus rutas aquí
app.get('/test', (req, res) => {
  res.json({ message: 'API funcionando correctamente' });
});

app.listen(port, () => {
  console.log(`🚀 API corriendo en puerto ${port}`);
});
```

4. Agregar al API Gateway

Editar `/opt/api-gateway/server.js` y agregar:

```
javascript
```

```
// Agregar después de las otras configuraciones de proxy
app.use('/nueva-ruta', createProxyMiddleware({
  target: 'http://127.0.0.1:3003', // Puerto de tu nueva API
  changeOrigin: true,
  pathRewrite: { '^/nueva-ruta': '' }
}));
```

5. Actualizar Dashboard HTML

En el mismo archivo del gateway, agregar nueva tarjeta en el HTML:

```
html

<a href="/nueva-ruta/" class="card">
  <h3> 🔥 Nueva API</h3>
  <p>Descripción de tu nueva API</p>
</a>
```

6. Reiniciar Gateway

```
bash

# Buscar proceso del gateway
ps aux | grep "api-gateway"

# Matar proceso anterior
kill [PID_DEL_GATEWAY]

# Ejecutar nuevo gateway
cd /opt/api-gateway
node server.js &
```

7. Ejecutar Nueva API

```
bash

cd /opt/nueva-api-nombre
node server.js &
```

Modificar API Existente

1. Acceder al código

- Ve a VS Code: <https://dev.api.clicinmobiliaria.com/ide/>
- Abre la carpeta de la API que quieres modificar

2. Hacer cambios

- Edita el código según necesites
- Guarda los archivos (Ctrl+S)

3. Reiniciar API (si es necesario)

```
bash

# Ver procesos corriendo
ps aux | grep node

# Identificar PID de tu API
# Matar proceso
kill [PID]

# Ejecutar de nuevo
cd /opt/tu-api
node server.js &
```

Gestión de Procesos

Ver APIs Corriendo

```
bash

netstat -tlnp | grep -E "8080|3001|3002|3003"
ps aux | grep node
```

Matar Proceso Específico

```
bash
```

```
kill [PID]
# Si no responde:
kill -9 [PID]
```

Ejecutar API en Segundo Plano

```
bash
cd /opt/tu-api
node server.js &
```

Ver Logs en Tiempo Real




```
bash
tail -f /var/log/nginx/access.log
tail -f /var/log/nginx/error.log
```

Estructura de Carpetas Recomendada

```
/opt/
├── api-gateway/      # Router central (puerto 8080)
│   ├── server.js
│   └── package.json
├── email-service/    # API original (puerto 3001)
├── image-compression-api/ # API imágenes (puerto 3002)
├── clients-api/      # Futura API clientes (puerto 3003)
├── properties-api/   # Futura API propiedades (puerto 3004)
└── users-api/        # Futura API usuarios (puerto 3005)
```

Puertos Asignados

Puerto	Servicio	Estado
3001	API Principal (emails)	✔ Usado
3002	API Imágenes	✔ Usado
3003	Disponible	○ Libre
3004	Disponible	○ Libre

Puerto	Servicio	Estado
3005	Disponible	 Libre
8080	API Gateway	 Usado
8443	VS Code Server	 Usado

Comandos Útiles

Verificar Conectividad

```
bash
# Probar API localmente
curl http://localhost:3002

# Probar a través del gateway
curl -k https://dev.api.clicinmobiliaria.com/images/
```

Monitoreo

```
bash
# Ver uso de recursos
top
htop

# Ver espacio en disco
df -h
```

Backups (Recomendado)

```
bash
# Crear backup de tus APIs
tar -czf apis-backup-$(date +%Y%m%d).tar.gz /opt/
```

Solución de Problemas

API no responde

1. Verificar que el proceso esté corriendo: `ps aux | grep node`

2. Verificar que el puerto esté libre: `netstat -tlnp | grep [PUERTO]`
3. Revisar logs de nginx: `tail -f /var/log/nginx/error.log`
4. Reiniciar la API específica

Gateway no redirecciona

1. Verificar que la configuración de proxy esté correcta
2. Verificar que la API destino esté corriendo
3. Reiniciar el gateway

Error 502/503

1. La API destino no está corriendo
2. Puerto incorrecto en la configuración
3. Problema de conectividad interna

Ventajas de esta Arquitectura

- ✅ **No más nginx manual** - Todo se maneja desde código
- ✅ **Desarrollo visual** - Usar VS Code en navegador
- ✅ **Escalabilidad** - Agregar APIs fácilmente
- ✅ **Centralizado** - Un solo punto de entrada
- ✅ **Mantenimiento simple** - Todo en `/opt/`

Próximos Pasos Sugeridos

1. **PM2** para gestión de procesos - Auto-restart y logs
2. **Base de datos compartida** - PostgreSQL/MySQL
3. **Autenticación centralizada** - JWT tokens
4. **Monitoreo** - Health checks automáticos
5. **CI/CD** - Deploy automatizado