

# Compatibilidad de OpenSSL en Node.js

Al ejecutar `ng build` con **Node.js 22.9.0** y **npm 10.8.3**, se encuentra el error:

```
error 0308010C digital envelope routines::unsupported
```

Este error generalmente surge debido a un problema de compatibilidad entre versiones más recientes de Node.js (como 22.9.0, que utiliza OpenSSL 3.0 o posterior) y ciertos algoritmos criptográficos utilizados por herramientas como Webpack, que Angular utiliza bajo el capó. A continuación, se presenta una guía paso a paso para solucionar este problema.

---

## Solución

**Paso 1: Habilitar el Proveedor de OpenSSL Legacy** La solución más sencilla y más inmediata es habilitar el proveedor de OpenSSL legacy estableciendo la variable de entorno `NODE_OPTIONS`. Esto permite que Node.js utilice algoritmos criptográficos más antiguos que pueden no estar soportados por defecto en OpenSSL 3.0+.

- **Para macOS/Linux:** Abre tu terminal y ejecuta:

```
export NODE_OPTIONS=--openssl-legacy-provider
ng build
```

- **Para Windows (Símbolo del sistema):** Abre el Símbolo del sistema y ejecuta:

```
set NODE_OPTIONS=--openssl-legacy-provider
ng build
```

- **Para Windows (PowerShell):** Abre PowerShell y ejecuta:

```
$env:NODE_OPTIONS="--openssl-legacy-provider"
ng build
```

Después de establecer la variable de entorno, ejecuta `ng build` nuevamente. Esto debería resolver el error en la mayoría de los casos permitiendo que Node.js procese las rutinas no soportadas.

---

**Paso 2: Verificar y Actualizar Angular CLI (Si es Necesario)** Si el error persiste después del Paso 1, tu versión de Angular CLI podría no ser completamente compatible con Node.js 22.9.0. Actualizarla a la versión más reciente puede ayudar.

- Verifica tu versión actual de Angular CLI:

```
ng --version
```

- Actualiza Angular CLI globalmente:

```
npm install -g @angular/cli
```

- Luego, intenta ejecutar `ng build` nuevamente.
- 

**Paso 3: Verificar y Actualizar Dependencias del Proyecto (Opcional)** Si el problema aún no se ha resuelto, las dependencias desactualizadas en tu proyecto podrían estar causando el problema. Para abordar esto:

- Abre tu archivo `package.json` y revisa las versiones de tus dependencias (por ejemplo, `@angular/core`, `@angular/cli`, etc.).
- Actualízalas cuidadosamente a sus versiones compatibles más recientes:

```
npm install
```

o, si deseas actualizar todas las dependencias:

```
npm update
```

- Ejecuta `ng build` nuevamente para probar.

*Nota:* Ten cuidado con las actualizaciones, ya que podrían introducir cambios disruptivos. Prueba tu aplicación a fondo después.

---

**Paso 4: Degradar Node.js (Último Recurso)** Si ninguna de las soluciones anteriores funciona, Node.js 22.9.0 podría no estar completamente soportado por la configuración actual de tu proyecto. Degradar a una versión de soporte a largo plazo (LTS) de Node.js, como **20.x** o **18.x**, podría resolver el problema.

- Desinstala tu versión actual de Node.js.
- Descarga e instala una versión LTS desde el sitio web oficial de Node.js.

- Verifica la versión instalada:

```
node -v
```

- Reinstala las dependencias de tu proyecto:

```
npm install
```

- Ejecuta `ng build` nuevamente.
- 

## Enfoque Recomendado

Comienza con el **Paso 1** (estableciendo `NODE_OPTIONS`), ya que es la solución más rápida y te permite seguir utilizando Node.js 22.9.0. Si eso no funciona, procede con el **Paso 2** y **Paso 3**. Usa el **Paso 4** solo si es necesario, ya que degradar Node.js podría limitar el acceso a características más recientes.

Esto debería resolver el error “error 0308010C digital envelope routines::unsupported” y hacer que tu comando `ng build` funcione nuevamente.