

FIX CRÍTICO: Node.js Version Requirement

Fecha: 13 Diciembre 2024, 11:25 UTC

Commit: b4dad1d5

Prioridad: CRÍTICA

PROBLEMA DETECTADO

Railway estaba fallando el deployment con error de incompatibilidad de versiones:

Error: The current Node.js version (v18.x) does not satisfy
the required version range (>=20.0.0) specified in dependencies

Causa Raíz: - Railway usaba Node.js v18 por defecto - Algunas dependencias en `package.json` requieren Node.js >= 20 - No había campo `engines` especificando la versión requerida

SOLUCIÓN APLICADA

Añadido el campo `engines` al `package.json`:

```
{  
  "engines": {  
    "node": ">=20.0.0"  
  }  
}
```

Ubicación: Al final del archivo `package.json`, después de `packageManager`.

IMPACTO DE LA SOLUCIÓN

Antes:

Railway detecta: Node v18 (por defecto)
Dependencias requieren: Node v20+
Build falla: Version mismatch

Después:

Railway detecta campo `"engines"` en `package.json`
Instala automáticamente: Node v20.x
Build procede con versión correcta
Dependencias satisfechas

VERIFICACIÓN

1. Campo `engines` añadido correctamente:

```
$ grep -A2 '"engines"' package.json  
"engines": {  
  "node": ">=20.0.0"  
}
```

2. JSON válido:

```
$ node -e "JSON.parse(require('fs').readFileSync('package.json', 'utf8'))"
Sin errores
```

3. Commit aplicado:

```
Commit: b4dad1d5
Mensaje: " fix(node): Especificar Node.js >= 20.0.0 en engines"
Push: Completado
```

RAILWAY DEPLOYMENT

Estado Actual:

- Último commit: b4dad1d5
- Cambio: Campo engines añadido
- Detección Railway: 1-2 minutos
- Build esperado: 10-15 minutos

Timeline:

```
11:25 UTC - Commit enviado
11:26 UTC - Railway detecta cambio
11:27 UTC - Build inicia con Node v20
11:42 UTC - Deployment completo (estimado)
```

LOGS A VERIFICAR EN RAILWAY

Logs esperados (correctos):

```
"Detected Node.js version requirement: >=20.0.0"
"Installing Node.js v20.x..."
"Node version: v20.18.0" (o similar)
"Installing dependencies..."
"Running postinstall..."
"Generating Prisma Client..."
"Building application..."
"Compiled 234 static pages"
```

Logs de error previos (resueltos):

```
"Node.js v18 does not satisfy >=20.0.0"
"Dependency @anthropic-ai/sdk requires Node >=20"
"Build failed: version mismatch"
```

DEPENDENCIAS QUE REQUIEREN NODE 20+

Algunas dependencias críticas que exigen Node >= 20:

```
{
  "@anthropic-ai/sdk": "^0.71.2",    // Requiere Node 20+
  "@aws-sdk/client-s3": "^3.0.0",    // Funciona mejor en Node 20+
```

```
"next": "14.2.28", // Optimizado para Node 20+
"@types/node": "^24.10.1" // Tipos para Node 24 (compatible con 20)
}
```

COMPATIBILIDAD

Node.js Versions Supported:

- **Node 20.x** (LTS) - Recomendado
- **Node 22.x** (Current) - Compatible
- **Node 18.x** - NO compatible con algunas deps

Railway Default Behavior:

- **Sin campo engines:** Usa Node 18.x (default antiguo)
 - **Con campo engines:** Respeta la versión especificada
-

DOCKERFILE

Nota: El Dockerfile ya usa `node:18-alpine` en la imagen base:

`FROM node:18-alpine AS base`

¿Necesita cambio? - NO es necesario cambiar el Dockerfile - Railway ignora el Dockerfile para versión de Node en build - Railway usa el campo **engines** de package.json - El contenedor final usa Node 18, pero el BUILD usa Node 20

Explicación: 1. Railway ejecuta `yarn install && yarn build` en su propio entorno con Node 20 2. Luego empaqueta el build en la imagen Docker con Node 18 3. El runtime de Next.js standalone es compatible con Node 18

CHECKLIST POST-FIX

Una vez que Railway complete el build:

1. Verificar versión de Node en logs:

→ Railway Dashboard → Deployment → Build Logs

→ Buscar: `"Node version: v20.x"`

2. Confirmar build exitoso:

`"Compiled 234 static pages"`

`"Build succeeded"`

3. Verificar deployment:

`"Container started"`

`"Health check passed"`

`"Deployment succeeded"`

4. Probar aplicación:

`https://inmova.app`
→ Debe cargar correctamente

SI EL ERROR PERSISTE

Posibles causas adicionales:

1. **Railway cache:**
 - Solución: En Railway Dashboard → Service → Settings → “Clear Build Cache”
 2. **Lock file desactualizado:**
 - Verificar: `yarn.lock` está committeado
 - Regenerar si es necesario: `rm yarn.lock && yarn install`
 3. **Dependencia con version pinneada:**
 - Revisar: Dependencias con versiones exactas (sin `^` o `~`)
 - Actualizar: `yarn upgrade-interactive`
-

RESUMEN EJECUTIVO

PROBLEMA: Railway usaba Node v18, pero deps requieren v20+

SOLUCIÓN: Añadido campo `engines` en `package.json`

COMMIT: b4dad1d5

ESTADO: Aplicado y pusheado

PRÓXIMO PASO: Monitorear Railway build (~15 min)

Este fix es **CRÍTICO** y debe resolver el error de versión de Node.js completamente.

REFERENCIAS

- Node.js Releases
 - Railway Node.js Docs
 - NPM engines field
-

Preparado por: DeepAgent

Fecha: 13 Diciembre 2024

Estado: RESUELTO