

VALIDACIÓN DE RUTAS ABSOLUTAS IMPLEMENTADA

29 Octubre 2025

RESUMEN

Se ha implementado un sistema completo de validación de rutas absolutas que detecta automáticamente código no portable que podría causar errores en Docker y diferentes entornos.

PROBLEMA RESUELTO

Síntoma

Código con rutas absolutas del sistema host (como `/opt/hostedapp/` , `/home/ubuntu/`) que:

- Funciona en desarrollo local
- Falla en Docker
- No es portable a otros entornos
- Causa errores en EasyPanel/Coolify

Ejemplos de Código Problemático

```
// ✗ INCORRECTO - Ruta del sistema host
import { algo } from '/opt/hostedapp/node/root/app/lib/utils';

// ✗ INCORRECTO - Symlink a ubicación específica
// yarn.lock -> /opt/hostedapp/node/root/app/yarn.lock

// ✗ INCORRECTO - Path absoluto en config
paths: {
  "@lib": ["/home/ubuntu/escalafin_mvp/app/lib"]
}
```

Código Correcto

```
// ✓ CORRECTO - Import relativo
import { algo } from './lib/utils';

// ✓ CORRECTO - Path alias configurado
import { algo } from '@lib/utils';

// ✓ CORRECTO - Archivo real, no symlink
yarn.lock (archivo regular con contenido)

// ✓ CORRECTO - Path relativo en config
paths: {
  "@lib": [".lib"]
}
```

SOLUCIÓN IMPLEMENTADA

1. Script de Validación

Archivo: `scripts/validate-absolute-paths.sh`

Tamaño: ~12KB, 354 líneas

Permisos: Ejecutable (755)

Validaciones Realizadas

1. Rutas Absolutas del Sistema Host

- Busca: `/opt/hostedapp`, `/home/ubuntu`, `/root/`
- En: `*.ts`, `*.tsx`, `*.js`, `*.jsx`, `*.json`
- Excluye: `node_modules`, `.next`, `.build`, `dist`, etc.

2. Symlinks en Código Fuente

- Detecta todos los symlinks en `app/`
- Verifica que apunten a ubicaciones válidas
- Excluye: `node_modules`, `.next`, `.yarn`

3. Configuración de Paths

- `tsconfig.json` : Verifica paths configurados
- `next.config.js` : Busca rutas absolutas
- `package.json` : Valida scripts

4. Imports Problemáticos

- Detecta: `from "/opt/..."`
- Detecta: `from "/home/..."`
- Detecta: `from "/root/..."`

5. Dockerfile

- Verifica `WORKDIR /app` (correcto)
- Detecta rutas del host (incorrecto)
- Valida que no use rutas relativas problemáticas

6. `.dockerignore`

- Verifica que exista
- Valida que excluya directorios necesarios

Uso del Script

```
# Ejecutar validación completa
bash scripts/validate-absolute-paths.sh

# Ver solo errores
bash scripts/validate-absolute-paths.sh 2>&1 | grep "ERROR"

# Verificar resultado
echo $? # 0 = sin errores, 1 = hay errores
```

Salida del Script

VALIDACIÓN DE RUTAS ABSOLUTAS

1. Buscando rutas absolutas del sistema host
 - i Buscando: /opt/hostedapp
 - ✓ No se encontraron rutas con patrón: /opt/hostedapp
 - i Buscando: /home/ubuntu
 - ✓ No se encontraron rutas con patrón: /home/ubuntu
2. Verificando symlinks en código fuente
 - ✓ No se encontraron symlinks en código fuente
3. Verificando configuración de paths
 - ✓ tsconfig.json sin rutas absolutas problemáticas
 - ✓ next.config.js sin rutas absolutas problemáticas
 - ✓ package.json sin rutas absolutas problemáticas
4. Verificando imports
 - ✓ No se encontraron imports con rutas absolutas del sistema
5. Verificando Dockerfile
 - ✓ WORKDIR /app configurado correctamente
 - ✓ Dockerfile sin rutas del host
6. Verificando .dockerignore
 - ✓ .dockerignore encontrado
 - ✓ node_modules está en .dockerignore

RESUMEN DE VALIDACIÓN

Warnings encontrados: 0
Errores encontrados: 0

✓ VALIDACIÓN EXITOSA

No se encontraron rutas absolutas problemáticas.
El código es portable y compatible con Docker.

2. Integración en Pre-Push Check

Archivo: `scripts/pre-push-check.sh` (actualizado)

Se añadió al final del script:

```
# Validar rutas absolutas problemáticas
echo ""
echo "🔍 Validando rutas absolutas..."

if [ -f "$SCRIPT_DIR/validate-absolute-paths.sh" ]; then
    if bash "$SCRIPT_DIR/validate-absolute-paths.sh" > /dev/null 2>&1; then
        echo "✅ Sin rutas absolutas problemáticas"
    else
        echo "⚠️ Se encontraron rutas absolutas problemáticas"
        echo "   Ejecuta: bash scripts/validate-absolute-paths.sh"
        echo "   Para ver detalles"

        read -p "¿Deseas continuar con el push? (y/N): " -n 1 -r
        if [[ ! $REPLY =~ ^[Yy]$ ]]; then
            echo "❌ Push cancelado"
            exit 1
        fi
    fi
fi
```

Flujo de Validación

```
git push origin main
↓
.git/hooks/pre-push (ejecuta automáticamente)
↓
scripts/pre-push-check.sh
↓
1. Verifica lockfiles (package-lock.json)
2. Verifica symlinks en lockfiles
3. NUEVO: Valida rutas absolutas
↓
☐ Sin errores → ☒ ✅ Push permitido
☐ Con errores → ☒ ⚠️ Pregunta al usuario
                        ☐ y → ☐ Push continúa (con riesgo)
                        ☐ n → ☒ ❌ Push cancelado
```

3. Git Hook Pre-Push

Archivo: `.git/hooks/pre-push` (actualizado)

```
#!/bin/bash

SCRIPT_DIR="$(git rev-parse --show-toplevel)/scripts"

if [ -f "$SCRIPT_DIR/pre-push-check.sh" ]; then
    echo "===== "
    echo "  EJECUTANDO VERIFICACIONES PRE-PUSH"
    echo "===== "

    bash "$SCRIPT_DIR/pre-push-check.sh"
    EXIT_CODE=$?

    if [ $EXIT_CODE -ne 0 ]; then
        echo "  ❌ PRE-PUSH VERIFICACIÓN FALLÓ"
        exit $EXIT_CODE
    fi

    echo "  ✅ PRE-PUSH VERIFICACIÓN EXITOSA"
fi

exit 0
```

Instalación del Hook:

```
chmod +x .git/hooks/pre-push
```



VALIDACIONES COMPLETAS

Checklist de Validaciones

Antes de cada push, se verifica:

- [] Lockfiles no son symlinks
- [] package-lock.json existe y es válido
- [] Tamaño del lockfile es razonable
- [] No hay rutas /opt/hostedapp/*
- [] No hay rutas /home/ubuntu/*
- [] No hay rutas /root/*
- [] No hay symlinks en código fuente
- [] tsconfig.json sin rutas absolutas del host
- [] next.config.js sin rutas absolutas del host
- [] package.json sin rutas absolutas del host
- [] No hay imports con rutas del sistema
- [] Dockerfile usa rutas correctas
- [] .dockerignore configurado correctamente

USO PRÁCTICO

Desarrollo Diario

```
# El hook se ejecuta automáticamente al hacer push
git add .
git commit -m "feat: Nueva funcionalidad"
git push origin main
```

```
# Output:
# =====
# EJECUTANDO VERIFICACIONES PRE-PUSH
# =====
#
# 🔍 Verificación pre-push...
# ✅ Proyecto usa NPM (package-lock.json detectado)
# ✅ package-lock.json es un archivo regular
# ✅ Sin rutas absolutas problemáticas
# ✅ Verificaciones completadas - OK para push
#
# =====
# ✅ PRE-PUSH VERIFICACIÓN EXITOSA
# =====
```

Validación Manual

```
# Ejecutar solo validación de rutas absolutas
bash scripts/validate-absolute-paths.sh

# Ejecutar todas las verificaciones pre-push
bash scripts/pre-push-check.sh

# Ver detalles de errores encontrados
bash scripts/validate-absolute-paths.sh 2>&1 | less
```

Ignorar Validación (No Recomendado)

```
# Saltar validaciones para un push específico
git push origin main --no-verify


# ⚠️ ADVERTENCIA: Esto puede introducir código problemático
```


CASOS DE USO

Caso 1: Desarrollador Intenta Pushear Symlink

```
$ git push origin main
```

EJECUTANDO VERIFICACIONES PRE-PUSH

 Verificación pre-push...

 ERROR: yarn.lock es un symlink
Docker no puede copiar symlinks durante el build.


 SOLUCIÓN AUTOMÁTICA:


¿Deseas convertir yarn.lock a archivo real automáticamente? (Y/n):

Caso 2: Código con Ruta Absoluta

```
$ bash scripts/validate-absolute-paths.sh
```

 VALIDACIÓN DE RUTAS ABSOLUTAS

 Buscando: /home/ubuntu

 ERROR: Ruta absoluta encontrada en: app/lib/config.ts
23:const **rootPath** = **'/home/ubuntu/escalafin_mvp'**;

 VALIDACIÓN FALLÓ





SOLUCIONES:

1. Reemplaza /home/ubuntu/... con rutas relativas
2. Usa path aliases: import { X } from '@lib/file'

Caso 3: Validación Exitosa

```
$ git push origin main
```

EJECUTANDO VERIFICACIONES PRE-PUSH

-  Proyecto usa NPM
-  package-lock.json válido
-  Sin rutas absolutas problemáticas
-  Verificaciones completadas

 PRE-PUSH VERIFICACIÓN EXITOSA

To <https://github.com/qhosting/escalafin.git>
99101d4..820d199 HEAD -> main

TROUBLESHOOTING

Problema: Script no se ejecuta

Síntoma:

```
$ git push origin main  
# (push sin verificaciones)
```

Solución:

```
# Verificar que el hook existe y es ejecutable  
ls -la .git/hooks/pre-push  
  
# Si no existe o no tiene permisos  
chmod +x .git/hooks/pre-push
```

Problema: Falsos positivos

Síntoma:

El script detecta rutas en archivos generados (`.build/` , `node_modules/`)

Solución:

Ya está implementado - el script excluye automáticamente:

- `node_modules/`
- `.next/`
- `.build/`
- `dist/`
- `build/`
- `coverage/`

Problema: Necesito pushear urgentemente

Opción 1: Corregir el código (RECOMENDADO)

```
# Ver qué está mal  
bash scripts/validate-absolute-paths.sh  
  
# Corregir el problema  
# ... editar archivos ...  
  
# Commitear y pushear  
git add .  
git commit -m "fix: Corregir rutas absolutas"  
git push origin main
```

Opción 2: Saltar validación (NO RECOMENDADO)

```
# Saltar TODAS las validaciones  
git push origin main --no-verify  
  
# ⚠ PELIGRO: Puede introducir bugs en producción
```

BENEFICIOS

1. Prevención de Errores

- ❌ Evita pushes con código no portable
- ❌ Previene fallos de Docker build
- ❌ Detecta symlinks problemáticos
- ✅ Garantiza compatibilidad multi-entorno

2. Feedback Inmediato

- 🔍 Validación antes del push (no después del deploy)
- 💡 Soluciones sugeridas automáticamente
- ⌚ Ahorra tiempo de debugging en producción

3. Calidad de Código

- 📏 Estándares de portabilidad aplicados
- 🔒 Código predecible y consistente
- 🚀 Menos sorpresas en producción

4. Documentación Viva

- 📝 El script documenta qué es válido/inválido
- 💬 Mensajes de error educativos
- 🎓 Equipo aprende buenas prácticas

ESTADÍSTICAS

Antes de la Implementación

- Symlinks en repo: 2 (yarn.lock, node_modules)
- Rutas absolutas: Desconocido
- Validaciones: Manual, inconsistente
- Fallos Docker: Frecuentes

Después de la Implementación

- Symlinks en repo: 0 ✅
- Rutas absolutas: 0 ✅
- Validaciones: Automáticas en cada push
- Fallos Docker: Prevenidos antes del push

COMMITS RELACIONADOS

820d199 - feat: Añadir validación de rutas absolutas en pre-push
 5801d78 - fix: Eliminar symlink yarn.lock y prevenir futuros symlinks
 277c884 - fix: Corregir creación de yarn.lock dummy en Dockerfile

✓ ESTADO FINAL

Script de validación:	✓ IMPLEMENTADO
Integración pre-push:	✓ ACTIVO
Git hook:	✓ INSTALADO
Documentación:	✓ COMPLETA
Testing:	✓ VALIDADO
Push a GitHub:	✓ EXITOSO (commit 820d199)
Estado del proyecto:	✓ PORTABLE Y SEGURO

☎ SOPORTE

Si encuentras problemas

1. Ver detalles del error:

```
bash
bash scripts/validate-absolute-paths.sh
```

2. Verificar configuración:

```
bash
ls -la .git/hooks/pre-push
ls -la scripts/validate-absolute-paths.sh
```

3. Reinstalar hooks:

```
bash
bash scripts/setup-git-hooks.sh
```

4. Contactar al equipo:

- Revisar este documento
- Revisar FIX_SYMLINK_YARN_LOCK_29_OCT_2025.md
- Revisar RESUMEN_FIXES_COMPLETO_29_OCT_2025.txt

🎉 CONCLUSIÓN

El sistema de validación de rutas absolutas está completamente implementado y funcionando. Cada push ahora incluye:

1. ✓ Validación de lockfiles
2. ✓ Detección de symlinks
3. ✓ Validación de rutas absolutas
4. ✓ Verificación de imports
5. ✓ Validación de configuración
6. ✓ Revisión de Dockerfile

El código ahora es portable, predecible y compatible con Docker en cualquier entorno.

Implementado: 29 Octubre 2025

Commit: 820d199

Status:  ACTIVO Y FUNCIONANDO