

Resumen de Fixes Pre-Deploy - 30 Octubre 2025

Fecha: 30 de Octubre de 2025

Commit: bf32ecf

Estado:  COMPLETADO Y VERIFICADO

Problemas Identificados

Durante la revisión pre-deploy con los scripts de validación, se identificaron los siguientes problemas críticos que impedían un deploy exitoso:

1. Rutas Absolutas Hardcodeadas




Archivos afectados:

- app/app/api/admin/storage/test/route.ts (línea 42)
- app/app/api/admin/storage/config/route.ts (línea 66)
- app/app/api/files/[...path]/route.ts (líneas 28, 31)

Problema:

```
//  ANTES - Ruta absoluta hardcodeada  
const uploadDir = '/home/ubuntu/escalafin_mvp/uploads'
```

Impacto:




-  No funciona en contenedores Docker (ruta no existe)
-  No funciona en servidores de producción
-  Causa errores de "Cannot find directory"

2. Yarn.lock en Proyecto NPM

Problema:

- El proyecto usa **NPM** (tiene package-lock.json)
- Pero también tenía yarn.lock (conflicto)
- Esto confunde a los sistemas de build y puede causar inconsistencias

Impacto:

-  Builds inconsistentes
-  Posibles conflictos de dependencias
-  Mayor tamaño del repositorio innecesariamente

3. Ruta Absoluta en Prisma Schema

Problema:

```
generator client {
  provider = "prisma-client-js"
  binaryTargets = ["native", "linux-musl-arm64-openssl-3.0.x"]
  output = "/home/ubuntu/escalafin_mvp/app/node_modules/.prisma/client" ← PROBLEMÁTICO
}
```

Impacto:

- ❌ No funciona en Docker/producción
- ❌ Prisma Client genera en ruta incorrecta
- ❌ Causa "Cannot find @prisma/client" en runtime

✓ Soluciones Implementadas

1. Corrección de Rutas Absolutas

app/app/api/admin/storage/test/route.ts

ANTES:

```
const uploadDir = config.uploadDir || '/home/ubuntu/escalafin_mvp/uploads'
```

DESPUÉS:

```
const uploadDir = config.uploadDir || path.join(process.cwd(), 'uploads')
```

Beneficios:

- ✓ Usa el directorio de trabajo actual (funciona en cualquier entorno)
- ✓ Respeto variable de entorno `config.uploadDir` si está definida
- ✓ Compatible con Docker, desarrollo y producción

app/app/api/admin/storage/config/route.ts

ANTES:

```
local: settings.local || {
  uploadDir: '/home/ubuntu/escalafin_mvp/uploads',
  baseUrl: '/api/files/serve',
  maxSize: 10
}
```

DESPUÉS:

```
local: settings.local || {
  uploadDir: process.env.LOCAL_UPLOAD_DIR || './uploads',
  baseUrl: '/api/files/serve',
  maxSize: 10
}
```

Beneficios:

- ✓ Usa variable de entorno `LOCAL_UPLOAD_DIR` si está definida

- ☒ Fallback a ruta relativa `./uploads`
- ☒ Configurable por entorno

app/app/api/files/[...path]/route.ts

ANTES:

```
const fullPath = path.join(
  process.env.LOCAL_UPLOAD_DIR || '/home/ubuntu/escalafin_mvp/uploads',
  filePath
)
const uploadDir = path.resolve(
  process.env.LOCAL_UPLOAD_DIR || '/home/ubuntu/escalafin_mvp/uploads'
)
```

DESPUÉS:

```
const defaultUploadDir = path.join(process.cwd(), 'uploads')
const fullPath = path.join(
  process.env.LOCAL_UPLOAD_DIR || defaultUploadDir,
  filePath
)
const uploadDir = path.resolve(
  process.env.LOCAL_UPLOAD_DIR || defaultUploadDir
)
```

Beneficios:

- ☒ Define fallback una sola vez (DRY - Don't Repeat Yourself)
- ☒ Usa `process.cwd()` para portabilidad
- ☒ Mantiene compatibilidad con variable de entorno

2. Eliminación de yarn.lock

Acción:

```
rm -f app/yarn.lock
```

Resultado:

- ☒ Solo queda `package-lock.json` (correcto para proyecto NPM)
- ☒ Builds más consistentes
- ☒ Sin conflictos de package managers

3. Corrección de Prisma Schema




ANTES:

```
generator client {
  provider = "prisma-client-js"
  binaryTargets = ["native", "linux-musl-arm64-openssl-3.0.x"]
  output = "/home/ubuntu/escalafin_mvp/app/node_modules/.prisma/client"
}
```

DESPUÉS:

```
generator client {
  provider = "prisma-client-js"
  binaryTargets = ["native", "linux-musl-arm64-openssl-3.0.x"]
}
```

Beneficios:

-  Prisma usa la ruta por defecto (portátil)
-  Funciona en desarrollo, Docker y producción
-  Cliente generado en la ubicación estándar

Post-fix:

```
npx prisma generate
```


Cliente regenerado exitosamente en ubicación correcta.



Scripts de Validación Ejecutados

1. fix-yarn-lock-symlink.sh

Resultado:

```
! ADVERTENCIA: yarn.lock es un symlink
! Convirtiendo a archivo real...
!  yarn.lock convertido a archivo real
```

Acción tomada:


- Convertido de symlink a archivo real
- Posteriormente eliminado (ya que el proyecto usa NPM)

2. validate-absolute-paths.sh

Primera ejecución:

```
✗ ERROR: Ruta absoluta encontrada en: app/app/api/admin/storage/test/route.ts
✗ ERROR: Ruta absoluta encontrada en: app/app/api/admin/storage/config/route.ts
✗ ERROR: Ruta absoluta encontrada en: app/app/api/files/[...path]/route.ts
```

Después de correcciones:

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

3. pre-push-check.sh

Resultado:

- ✓ Proyecto usa NPM (**package-lock.json** detectado)
- ✓ **package-lock.json** es un archivo regular
- ✓ Sin rutas absolutas problemáticas
- ✓ Verificaciones completadas - OK para push

4. revision-fix.sh ✓

Resultado:

- ✓ OK: schema.prisma no contiene rutas absolutas
- ✓ OK: Proyecto usa NPM (**package-lock.json** encontrado)
- ✓ OK: Script encontrado: app/scripts/setup-users-production.js
- ✓ OK: Todos los scripts necesarios presentes



Resumen de Cambios

Archivos Modificados

Archivo	Tipo de Cambio	Descripción
app/api/admin/storage/test/route.ts	🔧 Fix	Ruta absoluta → process.cwd()
app/api/admin/storage/config/route.ts	🔧 Fix	Ruta absoluta → Variable de entorno
app/api/files/[...path]/route.ts	🔧 Fix	Ruta absoluta → process.cwd()
app/prisma/schema.prisma	🔧 Fix	Remover output path absoluto
app/yarn.lock	🗑 Delete	Eliminado (proyecto usa NPM)

Estadísticas Git

```
git diff --stat 221622f..bf32ecf
.abacus.donotdelete           | 2 +-
app/app/api/admin/storage/config/route.ts | 2 +-
app/app/api/admin/storage/test/route.ts   | 2 +-
app/app/api/files/[...path]/route.ts     | 4 ++-
app/prisma/schema.prisma          | 3 +-
app/yarn.lock                    | 1 -
6 files changed, 6 insertions(+), 7 deletions(-)
```

Instrucciones de Deploy

Estado Actual

✓ Código sincronizado en ambos repositorios:

- `ghosting/escalafin` → commit `bf32ecf`
- `ghosting/escalafinmx` → commit `bf32ecf`

Para Hacer Rebuild en EasyPanel/Coolify

1. Limpiar caché de build:

- En EasyPanel: Settings → Build → ☒ Clear Build Cache
- En Coolify: Settings → Build → Clear build cache

2. Hacer rebuild:

- Click en “Rebuild” o “Deploy”
- Esperar a que termine (2-5 minutos)

3. Verificar en logs de build:

Commit: `bf32ecf`

✓ Compiled successfully

4. Verificar que la app inicia:

✓ Ready in X ms

▲ Next.js 14.2.28

- Local: `http://0.0.0.0:3000`

Variables de Entorno Recomendadas

Para producción, configurar:

```
# Opcional: Personalizar directorio de uploads
LOCAL_UPLOAD_DIR=/app/uploads

# Variables críticas (ya deberían estar configuradas)
DATABASE_URL=postgresql://...
NEXTAUTH_URL=https://tu-dominio.com
NEXTAUTH_SECRET=...
```

✓ Verificaciones Post-Deploy

Después del rebuild, verificar:

1. App Funciona

- [] La aplicación carga sin errores
- [] Login funciona correctamente
- [] Dashboard se muestra

2. Sin Errores en Logs

- [] No hay “Cannot find module” errors
- [] No hay “ENOENT: no such file or directory” errors
- [] Prisma Client se carga correctamente

3. Funcionalidad de Archivos

- [] Subir archivos funciona
- [] Descargar archivos funciona
- [] Rutas de archivos se resuelven correctamente

Beneficios de Estos Fixes

Portabilidad

- El código funciona en cualquier entorno
- Sin dependencias de rutas específicas del sistema
- Compatible con Docker, Kubernetes, etc.

Mantenibilidad

- Configuración mediante variables de entorno
- Sin hardcoding de valores específicos
- Más fácil de mantener y actualizar

Robustez

- Menos errores en producción
- Build más consistente
- Sin conflictos de package managers

Best Practices

- Uso correcto de `process.cwd()`
- Respeto por variables de entorno
- Configuración por entorno

Prevención Futura

Antes de Cada Deploy

1. Ejecutar scripts de validación:

```
bash
bash scripts/fix-yarn-lock-symlink.sh
bash scripts/validate-absolute-paths.sh
bash scripts/pre-push-check.sh
```







2. Verificar que no hay rutas absolutas:

```
bash
grep -r "/home/ubuntu" app/ | grep -v node_modules | grep -v .build
```

3. Verificar package manager consistente:

```
bash
# Solo debe existir uno:
ls -l app/package-lock.json # Para NPM
ls -l app/yarn.lock          # Para Yarn (no debe existir si usas NPM)
```

Durante Desarrollo

-  NO usar rutas como `/home/ubuntu/...`
-  Sí usar `process.cwd()`, `path.join()`, `path.resolve()`
-  NO hardcodear paths en el código
-  Sí usar variables de entorno para configuración
-  NO mezclar NPM y Yarn
-  Sí mantener un solo package manager



Checklist de Deploy

Antes de cada deploy, verificar:

- ☐ Sin rutas absolutas en código
- ☐ Solo un tipo de lockfile (`package-lock.json` O `yarn.lock`)
- ☐ Prisma schema sin output path absoluto
- ☐ Scripts de inicio presentes en Dockerfile
- ☐ Variables de entorno documentadas
- ☐ Tests pasan localmente
- ☐ Build local exitoso (`npm run build`)
- ☐ Commits pusheados a GitHub
- ☐ Ambos repos sincronizados (si aplica)



Troubleshooting

Error: Cannot find directory '/home/ubuntu/...'

Causa: Ruta absoluta hardcodeada en código

Solución:

1. Buscar la ruta: `grep -r "/home/ubuntu" app/`
2. Reemplazar con `process.cwd()` o variable de entorno
3. Commit y push
4. Rebuild

Error: Conflictos de package manager

Causa: Existen tanto `package-lock.json` como `yarn.lock`

Solución:

```
# Si usas NPM:
rm app/yarn.lock

# Si usas Yarn:
rm app/package-lock.json
```


Error: @prisma/client not found

Causa: Output path incorrecto en schema.prisma

Solución:

1. Remover línea `output = "..."` de schema.prisma
2. Regenerar: `npx prisma generate`
3. Commit y push
4. Rebuild

Información Adicional

Documentación Relacionada

- `FIX_DEPLOY_SYNC_29_OCT_2025.md` - Fix de sincronización deploy
- `INSTRUCCIONES_REBUILD_EASYPANEL.md` - Instrucciones de rebuild
- `DEPLOYMENT_GUIDE.md` - Guía completa de deployment

Commits Importantes

- `bf32ecf` - Fix de rutas absolutas y yarn.lock (ESTE COMMIT)
- `221622f` - Documentación de fix de sincronización
- `4635923` - Fix anterior de Prisma schema

Repositorios

- Origin: `https://github.com/qhosting/escalafin`
- EscalafinMX: `https://github.com/qhosting/escalafinmx`
- Ambos sincronizados en commit `bf32ecf`

Última actualización: 30 de Octubre de 2025

Commit: `bf32ecf`

Estado:  Listo para deploy en producción