


FIX: Script de Configuración de Usuarios de Prueba


Fecha: 29 de Octubre de 2025

Commit: d179cbc

Estado:  Completado y Pusheado

Problema Identificado

Al intentar configurar usuarios de prueba, se mostraba el mensaje:

 scripts/setup-users-production.js no encontrado, continuando...

Aunque el archivo **sí existía** en el repositorio y en la imagen Docker, el script de inicio no lo estaba encontrando correctamente.

Análisis del Problema

Causas Identificadas:

1. **Falta de diagnóstico:** El script no mostraba información sobre el directorio actual ni verificaba múltiples rutas
2. **Búsqueda simple:** Solo se buscaba en una ruta sin verificar alternativas
3. **Falta de logging:** No se registraba información útil para debugging
4. **start-easypanel.sh incompleto:** Este script no incluía la lógica de configuración de usuarios

Verificación del Estado:

```
# El archivo sí existe:
$ ls -la /home/ubuntu/escalafin_mvp/app/scripts/setup-users-production.js
-rw-r--r-- 1 ubuntu ubuntu 4741 Oct 28 19:18 setup-users-production.js

# El archivo está en Git:
$ git status app/scripts/setup-users-production.js
On branch main
nothing to commit, working tree clean

# El Dockerfile lo copia correctamente:
COPY --from=builder --chown=nextjs:nodejs /app/scripts ./scripts
```

✓ Solución Implementada

1. Mejora en `start-improved.sh`

Cambios aplicados:

- ✓ Diagnóstico extendido del directorio actual
- ✓ Búsqueda en múltiples rutas (relativa y absoluta)
- ✓ Listado del contenido del directorio scripts/ si no se encuentra
- ✓ Mensajes claros y orientadores para el usuario
- ✓ Configuración correcta de NODE_PATH

Código mejorado:

```
if [ "$USER_COUNT" = "0" ]; then
    echo " 🌱 Configurando usuarios de prueba..."
    echo " 📁 Directorio actual: $(pwd)"
    echo " 📁 Verificando existencia de archivos..."
    ls -la scripts/ 2>/dev/null || echo " ⚠️ Directorio scripts/ no encontrado"



    # Intentar con ruta relativa primero
    if [ -f "scripts/setup-users-production.js" ]; then
        echo " ✓ Script encontrado (ruta relativa)"
        SCRIPT_PATH="scripts/setup-users-production.js"
    # Intentar con ruta absoluta
    elif [ -f "/app/scripts/setup-users-production.js" ]; then
        echo " ✓ Script encontrado (ruta absoluta)"
        SCRIPT_PATH="/app/scripts/setup-users-production.js"
    else
        echo " ⚠️ setup-users-production.js no encontrado en:"
        echo "   - $(pwd)/scripts/setup-users-production.js"
        echo "   - /app/scripts/setup-users-production.js"
        echo " ⚠️ Listando contenido de directorios..."
        ls -la . 2>/dev/null || true
        ls -la scripts/ 2>/dev/null || true
        SCRIPT_PATH=""
    fi

    if [ -n "$SCRIPT_PATH" ]; then
        export NODE_PATH=/app/node_modules:$NODE_PATH
        echo " 📍 NODE_PATH configurado: $NODE_PATH"
        echo " 🚀 Ejecutando: node $SCRIPT_PATH"
        node "$SCRIPT_PATH" || echo " ⚠️ Error configurando usuarios, continuando..."
    else
        echo " ⚠️ No se puede configurar usuarios automáticamente"
        echo " 💡 Configura manualmente usando el panel de administración"
    fi
else
    echo " ✓ DB ya inicializada con usuarios"
fi
```

2. Actualización de `start-easypanel.sh`

Cambios aplicados:

- ✓ Añadida lógica completa de configuración de usuarios
- ✓ Verificación del conteo de usuarios en DB
- ✓ Búsqueda en múltiples rutas

-  Configuración de NODE_PATH
-  Logging mejorado

Nueva sección añadida:

```
# Configurar usuarios de prueba si la DB está vacía
echo ""
echo "🌱 Verificando usuarios de prueba..."
USER_COUNT=$(node -e "
  const { PrismaClient } = require('@prisma/client');
  const prisma = new PrismaClient();
  prisma.user.count()
    .then(count => { console.log(count); process.exit(0); })
    .catch(err => { console.error('0'); process.exit(0); })
    .finally(() => prisma.$disconnect());
" 2>/dev/null || echo "0")

echo "👤 Usuarios en DB: $USER_COUNT"

if [ "$USER_COUNT" = "0" ]; then
  echo "🌱 Configurando usuarios de prueba..."
  echo "📁 Directorio actual: $(pwd)"

  # Búsqueda inteligente del script...
  # [Código completo en el archivo]
fi
```



Archivos Modificados

```
start-improved.sh      - Mejorado con diagnóstico extendido
start-easypanel.sh     - Añadida configuración de usuarios
app/scripts/setup-users-production.js - Ya existía, verificado
```



Instrucciones de Despliegue en EasyPanel

Paso 1: Pull del Último Commit

En EasyPanel, ve a tu servicio y:

1. Haz clic en el botón **“Rebuild”** o **“Redeploy”**
2. Si no sincroniza automáticamente, ve a **Settings → Source** y haz clic en **“Pull Latest”**

Paso 2: Limpiar Cache de Build (Opcional pero Recomendado)

```
# En EasyPanel, ejecuta estos comandos en el terminal del servicio:
cd /app
rm -rf .next/cache
```

O desde la interfaz:

1. Ve a **Settings → Build Settings**

2. Activa “Clean Build Cache”








3. Guarda y reconstruye




Paso 3: Verificar en los Logs





Después del rebuild, los logs de inicio deberían mostrar:


Si la DB está vacía:

```

 Verificando usuarios de prueba...
 Usuarios en DB: 0
 Configurando usuarios de prueba...
 Directorio actual: /app
 Script encontrado (ruta relativa)
 NODE PATH configurado: /app/node_modules:...
 Ejecutando: node scripts/setup-users-production.js

 CONFIGURANDO USUARIOS DE PRUEBA - ESCALAFIN
=====
 Conexión exitosa
 Usuarios actuales en la base de datos: 0




 Creando/Actualizando usuarios de prueba...
 ADMIN - admin@escalafin.com
 ASESOR - asesor@escalafin.com
 CLIENTE - cliente@escalafin.com

 USUARIOS DE PRUEBA CONFIGURADOS EXITOSAMENTE

```

Si la DB ya tiene usuarios:

```

 Verificando usuarios de prueba...
 Usuarios en DB: 3
 DB ya tiene usuarios configurados

```

Paso 4: Verificar Acceso

Prueba el login con las credenciales de prueba:

Administrador:

- Email: admin@escalafin.com

- Password: admin123

Asesor:

- Email: asesor@escalafin.com

- Password: asesor123

Cliente:

- Email: cliente@escalafin.com

- Password: cliente123

Diagnóstico de Problemas

Si el script aún no se encuentra:

1. Verificar que el commit se haya sincronizado:

```
# En el terminal de EasyPanel:  
cd /app  
git log -1 --oneline  
# Debe mostrar: d179cbc fix: Mejorar detección...
```

2. Verificar que el directorio scripts existe:

```
ls -la /app/scripts/  
# Debe mostrar setup-users-production.js
```

3. Verificar permisos del archivo:

```
ls -la /app/scripts/setup-users-production.js  
# Debe ser legible (r--r--r--)
```

4. Verificar que bcryptjs está disponible:

```
ls -la /app/node_modules/bcryptjs/  
# Debe existir y contener index.js
```

Si el script falla al ejecutarse:

Error: Cannot find module '@prisma/client'

```
# Verificar que Prisma Client esté generado:  
ls -la /app/node_modules/.prisma/client/
```

Error: Cannot find module 'bcryptjs'

```
# Verificar que bcryptjs esté en runtime:  
ls -la /app/node_modules/bcryptjs/
```

Solución: Reconstruir la imagen desde cero:

1. En EasyPanel: **Settings** → **Build Settings**
 2. Activa **“Clean Build Cache”**
 3. Click en **“Rebuild”**
-



Cambios en el Repositorio

```
# Commit realizado:
$ git log --oneline -1
d179cbc fix: Mejorar detección y ejecución del script de usuarios de prueba

# Archivos modificados:
$ git show --name-only d179cbc
start-improved.sh
start-easypanel.sh

# Push exitoso:
To https://github.com/qhosting/escalafin.git
30254df..d179cbc main -> main
```



Checklist de Verificación

- [x] Script `setup-users-production.js` existe en el repositorio
- [x] Script está incluido en `.dockerignore` (no excluido)
- [x] Dockerfile copia el directorio `scripts/`
- [x] `start-improved.sh` actualizado con diagnóstico extendido
- [x] `start-easypanel.sh` actualizado con configuración de usuarios
- [x] Commit realizado y pusheado a GitHub
- [x] Documentación completa creada
- [] Pull realizado en EasyPanel
- [] Build exitoso con nuevos logs
- [] Usuarios de prueba creados automáticamente
- [] Login verificado con credenciales de prueba



Próximos Pasos

- 1. Pull del último commit en EasyPanel**
 - Ve a tu servicio en EasyPanel
 - Haz clic en “Rebuild” o “Pull Latest”
- 2. Verificar logs de inicio**
 - Revisa que el script sea encontrado y ejecutado
 - Verifica que los usuarios sean creados
- 3. Probar login**
 - Intenta acceder con las credenciales de prueba
 - Verifica que cada perfil funcione correctamente
- 4. Si hay problemas**
 - Consulta la sección “Diagnóstico de Problemas” arriba
 - Revisa los logs del servicio en EasyPanel
 - Verifica las variables de entorno (`DATABASE_URL`)

Soporte

Si después de aplicar este fix sigues teniendo problemas:

1. Copia los logs completos del inicio del servicio
 2. Verifica que `git log -1` muestre el commit `d179cbc`
 3. Comparte los logs para diagnóstico adicional
-

Documentación actualizada: 29 de Octubre de 2025

Versión del fix: 1.0

Estado: Listo para despliegue en producción 