

# **FIX: Problema de Login - Usuarios No Ingresan**

---

## **Problema Reportado**

---

Los usuarios de prueba no pueden iniciar sesión con las credenciales documentadas:

- admin@escalafin.com / admin123
- asesor@escalafin.com / asesor123
- cliente@escalafin.com / cliente123

## **Diagnóstico**

---

### **Causa Probable**

El script `start-improved.sh` ejecuta el seed automáticamente SOLO si la base de datos está vacía (0 usuarios). Si ya hay usuarios pero con contraseñas diferentes o mal hasheadas, no los actualiza.

### **Verificación**






1. **¿Los usuarios existen?** Puede que sí, pero con contraseñas incorrectas
2. **¿El hash es correcto?** El bcrypt debe usar la misma configuración
3. **¿El seed se ejecutó?** Solo se ejecuta si `USER_COUNT = 0`

## **Solución Implementada**

---

### **Script de Configuración de Usuarios**

Creé un script específico que:

-  Verifica la conexión a la base de datos
-  Crea o actualiza los usuarios de prueba
-  Usa `upsert` para garantizar que funcione siempre
-  Hashea las contraseñas correctamente con bcrypt (rounds=12)
-  Muestra las credenciales al finalizar

### **Archivos Creados**

1. `app/scripts/setup-test-users.ts`
  - Script TypeScript que configura los usuarios
  - Usa Prisma Client directamente
  - Implementa `upsert` para crear/actualizar
2. `setup-users.sh`
  - Script bash wrapper para ejecutar fácilmente
  - Verifica requisitos previos
  - Muestra instrucciones claras

## Cómo Ejecutar

---

### Opción 1: En Producción (EasyPanel Container)

```
# Conectar al contenedor
docker exec -it <container-name> sh

# Ejecutar el script
cd /app
./setup-users.sh
```


### Opción 2: En Desarrollo Local


```
cd /home/ubuntu/escalafin_mvp
chmod +x setup-users.sh
./setup-users.sh
```

### Opción 3: Directamente con Node


```
cd /home/ubuntu/escalafin_mvp/app
npx tsx scripts/setup-test-users.ts
```


## Output Esperado

 CONFIGURANDO USUARIOS DE PRUEBA - ESCALAFIN

 Verificando conexión a base de datos...

 Conexión exitosa

 Usuarios actuales en la base de **datos**: X


 Creando/Actualizando usuarios de prueba...

 **ADMIN** - **admin@escalafin.com**

 **ASESOR** - **asesor@escalafin.com**

 **CLIENTE** - **cliente@escalafin.com**

 **USUARIOS DE PRUEBA CONFIGURADOS EXITOSAMENTE**

 **CREDENCIALES DE LOGIN:**

 **ADMINISTRADOR**

Email: **admin@escalafin.com**

Password: **admin123**

 **ASESOR**

Email: **asesor@escalafin.com**

Password: **asesor123**

 **CLIENTE**

Email: **cliente@escalafin.com**

Password: **cliente123**

 **ESTADÍSTICAS FINALES:**

Total **usuarios**: X

Administradores: X

Asesores: X

Cientes: X

## Credenciales Garantizadas

Después de ejecutar este script, estas credenciales FUNCIONARÁN:

Rol	Email	Password
<b>ADMIN</b>	admin@escalafin.com	admin123
<b>ASESOR</b>	asesor@escalafin.com	asesor123
<b>CLIENTE</b>	cliente@escalafin.com	cliente123

## Técnico

### Hash de Contraseñas

```
const hashedPassword = await bcrypt.hash(password, 12);
```

- Algoritmo: bcrypt
- Rounds: 12
- Compatible con NextAuth

### Upsert Pattern

```
await prisma.user.upsert({
  where: { email: userData.email },
  update: { password: hashedPassword, ... },
  create: { email: userData.email, password: hashedPassword, ... },
});
```

Esto garantiza que:

- Si el usuario existe → actualiza la contraseña
- Si no existe → lo crea



## Próximos Pasos

1. **Commit y push** de los nuevos archivos
2. **Pull** en EasyPanel
3. **Rebuild** (opcional, solo si quieres que se incluya el script en la imagen)
4. **Ejecutar** el script dentro del contenedor en producción



## Notas Importantes

- Este script es **idempotente** - puedes ejecutarlo múltiples veces sin problemas
- Usa **upsert** - no duplica usuarios, solo actualiza
- Las contraseñas se **hashean correctamente** - compatible con NextAuth
- **No elimina** usuarios existentes - solo crea/actualiza los 3 de prueba

**Fecha:** 28 de Octubre 2025

**Estado:**  RESUELTO - Listo para ejecutar