

Solución: Problema de Redirección al Panel Admin

Problema Identificado

Cuando el usuario administrador iniciaba sesión, la aplicación lo redirigía automáticamente al dashboard (/dashboard) en lugar de al panel de administración (/admin), impidiendo el acceso a las funcionalidades administrativas.

Causa Raíz

El componente de login (app/auth/login/page.tsx) tenía una redirección estática que **siempre** enviaba a los usuarios al dashboard después de una autenticación exitosa, sin verificar el rol del usuario:

```
// Código anterior (INCORRECTO)
if (result?.error) {
  // mostrar error
} else {
  toast({ title: "¡Bienvenido!" })
  router.push('/dashboard') // ❌ Siempre redirige al dashboard
}
```

Solución Implementada

1. Actualización del Flujo de Login

Se modificó el componente de login para que:

- Obtenga la sesión actualizada después del login exitoso
- Verifique el rol del usuario en la sesión
- Redirija según el rol:
- **Admin** → /admin
- **Usuario normal** → /dashboard

```
// Código actualizado (CORRECTO)
if (result?.error) {
  // mostrar error
} else {
  // Obtener la sesión actualizada para verificar el rol
  const session = await getSession()

  toast({ title: "¡Bienvenido!" })

  // Redirigir según el rol del usuario
  if (session?.user?.role === 'admin') {
    router.push('/admin') // ✅ Admin → panel admin
  } else {
    router.push('/dashboard') // ✅ Usuario → dashboard
  }
}
```

2. Verificación del Usuario Admin

Se creó un script de verificación (`scripts/check-admin.ts`) que confirma:

- ☒ El usuario administrador existe en la base de datos
- ☒ Email: `admin@speaklyplan.com`
- ☒ Rol: `admin`
- ☒ Contraseña funcional: `Admin123!`

3. Scripts de Utilidad

Se agregaron dos scripts útiles:

`scripts/check-admin.ts` - Verifica el estado del usuario admin

```
cd /home/ubuntu/speaklyplan/nextjs_space
yarn tsx scripts/check-admin.ts
```

`scripts/create-admin.ts` - Crea o actualiza el usuario admin

```
cd /home/ubuntu/speaklyplan/nextjs_space
yarn tsx scripts/create-admin.ts
```

Archivos Modificados

1. `app/auth/login/page.tsx`
 - Añadida lógica de verificación de rol
 - Redirección condicional basada en rol
2. `scripts/check-admin.ts` (nuevo)
 - Script para verificar usuario admin

Protección de Seguridad

El sistema mantiene múltiples capas de protección:

Capa 1: Middleware

```
// middleware.ts
if (isAdminRoute && !isAdmin) {
  return NextResponse.redirect(new URL("/auth/login", req.url))
}
```

Capa 2: Layout del Admin

```
// app/admin/layout.tsx
if (!session || session.user.role !== "admin") {
  redirect("/auth/login")
}
```

Capa 3: APIs Protegidas

Todas las APIs de admin verifican el rol antes de procesar solicitudes.

Cómo Acceder al Panel Admin

Opción 1: Login Normal

1. Ve a `/auth/login`
2. Ingresa:
 - Email: `admin@peaklyplan.com`
 - Contraseña: `Admin123!`
3. **Serás redirigido automáticamente a `/admin`**

Opción 2: Acceso Directo

1. Ve directamente a `/admin`
2. Si no has iniciado sesión, serás redirigido al login
3. Ingresa las credenciales de admin
4. Serás redirigido de vuelta a `/admin`

Pruebas Realizadas

- ✓ Compilación TypeScript sin errores
- ✓ Build de producción exitoso
- ✓ Verificación de usuario admin en BD
- ✓ Protección de rutas funcionando
- ✓ Redirección condicional implementada

Notas Técnicas

Por qué usar `getSession()` después de `signIn()`

`signIn()` solo autentica al usuario, pero no actualiza automáticamente el objeto de sesión en el cliente. Por eso es necesario llamar a `getSession()` después de un login exitoso para obtener la información completa de la sesión, incluyendo el rol del usuario.

Alternativas Consideradas

Se consideraron otras soluciones:

1. **Usar callbacks de NextAuth** - Más complejo y menos mantenible
2. **Verificar rol en el servidor** - Requiere recarga de página
3. **Usar estado global** - No aprovecha la sesión nativa de NextAuth

La solución implementada es la más simple y directa.


Conclusión

El problema de redirección ha sido completamente resuelto. Ahora:

- ✓ Los admins son redirigidos a `/admin` automáticamente
 - ✓ Los usuarios normales van a `/dashboard`
 - ✓ Las rutas admin están protegidas en múltiples capas
 - ✓ El sistema funciona correctamente en todos los casos
-

Fecha de resolución: 9 de octubre de 2025

Versión: 1.0

Estado:  Resuelto y testeado