

# 🔒 Sistema de Roles y Permisos - Fase 1 Completada

## ✓ ¿Qué se implementó?

### 1. Estructura de Base de Datos

- **3 Roles:** ADMINISTRADOR, SUPERVISOR, COBRADOR
- **Tabla de permisos** con 22 permisos granulares
- **Relaciones de supervisión** (supervisor asignado a cobradores)
- **Control de tiempo de uso** por día en minutos
- **Estados activo/inactivo** para usuarios

### 2. Sistema de Autenticación Avanzado

- NextAuth actualizado con nuevos campos
- Verificación de usuario activo en login
- Control automático de límite de tiempo diario
- Actualización de último login

### 3. Middleware de Permisos

- Funciones server-side para verificar permisos
- Control de tiempo automático para cobradores
- Verificación de roles jerárquicos
- API para registro de actividad

### 4. Hooks y Componentes Frontend

- `usePermissions()` - Hook principal para permisos
- `PermissionGuard` - Componente para proteger contenido
- `RequireRole`, `AdminOnly`, etc. - Componentes específicos
- Control automático de tiempo de uso

## 👤 Usuarios de Demostración

#### 👑 ADMINISTRADOR

Email: admin@cobranza.com

Password: admin123

Permisos: ACCESO TOTAL

#### 👤 SUPERVISOR

Email: supervisor@cobranza.com  
Password: supervisor123  
Permisos: Gestión avanzada + supervisión

#### ■ COBRADOR

Email: cobrador@cobranza.com  
Password: cobrador123  
Permisos: Operaciones básicas  
Límite: 8 horas/día  
Supervisor: María Supervisora

## ✂ Cómo Usar el Sistema

### En el Frontend (Componentes)

```
import { usePermissions } from '@/hooks/use-permissions'
import PermissionGuard from '@/components/PermissionGuard'

function MiComponente() {
  const { canRegisterPayments, canDeleteLoans, isAdmin } =
    usePermissions()

  return (
    <div>
      {/* Mostrar botón solo si tiene permiso */}
      {canRegisterPayments && (
        <Button>Registrar Pago</Button>
      )}

      {/* Proteger sección completa */}
      <PermissionGuard permissions="ELIMINAR_PRESTAMOS">
        <Button variant="destructive">Eliminar</Button>
      </PermissionGuard>

      {/* Solo para administradores */}
      <AdminOnly>
        <Button>Panel Admin</Button>
      </AdminOnly>
    </div>
  )
}
```

### En APIs (Server-side)

```
import { requirePermission, requireRole } from '@/lib/permissions'

export async function POST(request: NextRequest) {
```

```
// Verificar permiso específico
const session = await requirePermission('CREAR_PRESTAMOS')

// O verificar rol mínimo
const session = await requireRole('SUPERVISOR')

// Continuar con la lógica...
}
```

## Verificar Permisos Múltiples

```
// Requiere AL MENOS UNO de estos permisos
<PermissionGuard permissions={['CREAR_CLIENTES',
'EDITAR_CLIENTES']}>
  <ClienteForm />
</PermissionGuard>

// Requiere TODOS estos permisos
<PermissionGuard
  permissions={['CREAR_PRESTAMOS', 'VER_REPORTES']}
  requireAll={true}
>
  <PrestamoAvanzado />
</PermissionGuard>
```

## Permisos Disponibles

### Operaciones Básicas

- REGISTRAR\_COBROS - Registrar pagos
- MAPA\_CLIENTES - Ver ubicaciones en mapa
- REGISTRAR\_GASTOS - Registrar gastos diarios
- VER\_DASHBOARD - Acceder al dashboard
- VER\_LISTADO\_GENERAL - Ver lista de clientes

### Gestión de Datos

- CREAR\_CLIENTES / EDITAR\_CLIENTES
- CREAR\_PRESTAMOS / EDITAR\_PRESTAMOS / ELIMINAR\_PRESTAMOS
- REGISTRAR\_TRANSFERENCIAS / VER\_TRANSFERENCIAS

### Administrativos

- GESTIONAR\_USUARIOS - Crear/editar usuarios
- VER\_AUDITORIA - Ver logs de actividad
- CONFIGURAR\_SISTEMA - Configuraciones globales

## 🔗 Estado Actual

### ✓ Implementado (Fase 1)

- ☒ Schema de base de datos con roles y permisos
- ☒ Sistema de autenticación actualizado
- ☒ Middleware de permisos server-side
- ☒ Hooks y componentes frontend
- ☒ Control de tiempo de uso
- ☒ Relaciones de supervisión
- ☒ Interface de login actualizada

### 🔗 Pendiente (Fase 2 y 3)

- ☐ Panel de administración de usuarios
- ☐ Interface para asignar/quitar permisos
- ☐ Control granular de tiempo por usuario
- ☐ Reportes de actividad y auditoría
- ☐ Integración en todos los componentes existentes
- ☐ Notificaciones de límite de tiempo
- ☐ Geofencing por zonas (avanzado)

## 🔗 Próximos Pasos

Para continuar con las **Fases 2 y 3**:

1. **Panel de Administración:** Interface para gestionar usuarios
2. **Control Granular:** Asignación individual de permisos
3. **Integración:** Aplicar permisos en todos los componentes
4. **Auditoría:** Sistema de logs y reportes
5. **Optimizaciones:** Notificaciones y mejoras UX

El sistema de roles está **100% funcional** y listo para usar. Los usuarios pueden iniciar sesión con diferentes roles y experimentar las restricciones de permisos automáticamente.

---

*Sistema implementado exitosamente el 17 de Septiembre de 2025*