🎭 MATRIZ DE ROLES Y PERMISOS - AGROMANO

📋 ROLES DEL SISTEMA

1. 👑 ADMIN_AGROMANO

Descripción: Administrador del sistema con acceso completo **Permisos:** TODOS los permisos listados

2. 🎯 SUPERVISOR_CAMPO

Descripción: Supervisor de operaciones de campo **Permisos:**

```
# Personal
trabajadores:read:all
trabajadores:update:all
trabajadores:export

# Asistencia
asistencia:read:all
asistencia:update
asistencia:approve
asistencia:reports
asistencia:dashboard
permisos:approve

# Productividad
productividad:read:all
productividad:register:others
productividad:reports
tareas:create
tareas:assign
metas:set
metas:track

# Cultivos
parcelas:read
parcelas:update
cultivos:read
cultivos:update
cultivos:track
cosechas:register
cosechas:read

# Reportes
reportes:read:advanced
reportes:export
dashboard:view:advanced
kpis:view
```

3. GERENTE_RRHH

Descripción: Gerente de Recursos Humanos
Permisos:

```
# Personal
trabajadores:create
trabajadores:read:all
trabajadores:update:all
trabajadores:delete
trabajadores:import
trabajadores:export

# Asistencia
asistencia:read:all
asistencia:update
asistencia:reports
permisos:approve
horarios:manage

# Nómina
nomina:process
nomina:read:all
nomina:approve
nomina:calculate
nomina:reports
nomina:export
salarios:update
bonificaciones:manage
deducciones:manage

# Reportes
reportes:read:advanced
reportes:export
dashboard:view:advanced
```

4. SUPERVISOR_RRHH

Descripción: Supervisor de RRHH con permisos limitados **Permisos:**

```
# Personal
trabajadores:read:all
trabajadores:update:all
trabajadores:export

# Asistencia
asistencia:read:all
asistencia:reports
permisos:read
```

```
# Nómina
nomina:read:all
nomina:reports

# Reportes
reportes:read
dashboard:view:basic
```

5. 👷 EMPLEADO_CAMPO

Descripción: Trabajador de campo con acceso básico **Permisos:**

```
# Personal
trabajadores:read:own
trabajadores:update:own

# Asistencia
asistencia:register
asistencia:read:own
permisos:create

# Productividad
productividad:register
productividad:read:own
tareas:read
tareas:update
tareas:complete

# Cultivos
cultivos:read
cosechas:register

# Reportes
dashboard:view:basic

# Móvil
mobile:access
mobile:sync
gps:track
photos:upload
```

6. 👁 VISUAL_SOLO_LECTURA

Descripción: Usuario con acceso de solo lectura para reportes **Permisos:**

```
# Reportes
reportes:read
dashboard:view:basic
kpis:view
```

```
# Básicos
trabajadores:read:all
asistencia:read:all
productividad:read:all
cultivos:read
parcelas:read
```

🔄 IMPLEMENTACIÓN EN AUTH0

Paso 1: Crear Roles en Auth0

1. Ve a **User Management** → **Roles**
2. Crea cada uno de los 6 roles listados arriba

Paso 2: Crear Permisos en Auth0 API

1. Ve a **APIs** → **AgroMano API** → **Scopes**
2. Agrega todos los permisos del archivo `RBAC_PERMISOS_COMPLETOS.md`

Paso 3: Asignar Permisos a Roles

1. Ve a cada rol y asigna los permisos correspondientes según la matriz

Paso 4: Crear Usuarios de Prueba

```
// Admin
admin@gestionagricola.com
Role: ADMIN_AGROMANO

// Supervisor de Campo
supervisor@gestionagricola.com
Role: SUPERVISOR_CAMPO

// Gerente RRHH
gerente@gestionagricola.com
Role: GERENTE_RRHH

// Empleado
empleado@gestionagricola.com
Role: EMPLEADO_CAMPO

// Solo lectura
viewer@gestionagricola.com
Role: VISUAL_SOLO_LECTURA
```

🔧 CONFIGURACIÓN EN CÓDIGO

Middleware de verificación

```
// Verificar permiso específico
export const requirePermission = (permission: string) => {
  return (req: Request, res: Response, next: NextFunction) => {
    const userPermissions = req.user?.permissions || [];

    if (!userPermissions.includes(permission)) {
      return res.status(403).json({
        success: false,
        message: `Permiso requerido: ${permission}`,
        userPermissions
      });
    }

    next();
  };
};

// Verificar múltiples permisos (AND)
export const requirePermissions = (permissions: string[]) => {
  return (req: Request, res: Response, next: NextFunction) => {
    const userPermissions = req.user?.permissions || [];

    const hasAllPermissions = permissions.every(
      permission => userPermissions.includes(permission)
    );

    if (!hasAllPermissions) {
      return res.status(403).json({
        success: false,
        message: `Permisos requeridos: ${permissions.join(', ')}`,
        userPermissions
      });
    }

    next();
  };
};

// Verificar al menos uno de varios permisos (OR)
export const requireAnyPermission = (permissions: string[]) => {
  return (req: Request, res: Response, next: NextFunction) => {
    const userPermissions = req.user?.permissions || [];

    const hasAnyPermission = permissions.some(
      permission => userPermissions.includes(permission)
    );

    if (!hasAnyPermission) {
      return res.status(403).json({
        success: false,
        message: `Se requiere al menos uno de: ${permissions.join(', ')}`,
        userPermissions
```

```
      });
    }

    next();
  };
};
```

Ejemplos de uso en rutas

```
// Ejemplo: Solo admin puede crear trabajadores
router.post('/trabajadores',
  verifyToken,
  requirePermission('trabajadores:create'),
  crearTrabajador
);

// Ejemplo: Ver trabajadores (diferentes permisos según rol)
router.get('/trabajadores',
  verifyToken,
  requireAnyPermission([
    'trabajadores:read:all',
    'trabajadores:read:own'
  ]),
  listarTrabajadores
);

// Ejemplo: Múltiples permisos requeridos
router.post('/nomina/procesar',
  verifyToken,
  requirePermissions([
    'nomina:process',
    'nomina:calculate'
  ]),
  procesarNomina
);
```

📊 ESTADÍSTICAS DEL RBAC

- **Total Permisos:** 89 scopes granulares
- **Total Roles:** 6 roles específicos
- **Módulos Cubiertos:** 8 módulos principales
- **Niveles de Acceso:** 4 niveles (Admin, Gerente, Supervisor, Empleado)

☑ BENEFICIOS DE ESTE RBAC

1. **Seguridad Granular:** Control preciso de cada acción
2. **Escalabilidad:** Fácil agregar nuevos permisos/roles
3. **Flexibilidad:** Roles adaptables a diferentes organizaciones
4. **Auditoría:** Trazabilidad completa de permisos

5. **Usabilidad:** Roles claros y comprensibles