PERMISOS DETALLADOS POR ROL

|  |  |  |  |
| --- | --- | --- | --- |
| ROL | DESCRIPCIÓN | PERMISOS | PÁGINA PRINCIPAL |
| 🔴 Admin | Administrador del Sistema | ✅ ACCESO TOTAL - Todos los permisos | /Home |
| 🔵 Manager | Gerente de Sucursal | ✅ Casi completo ❌ admin\_only | /Home |
| 🟠 Supervisor | Supervisor | ✅ orders kitchen payments tables | /Order |
| 🟢 Waiter | Mesero | ✅ orders tables customers | /Order |
| 🟣 Cashier | Cajero | ✅ orders payments customers | /Order |
| 🟤 Chef | Cocinero | ✅ kitchen orders | /StationOrders |
| 🟢 Bartender | Bartender | ✅ orders kitchen | /StationOrders |
| ⚫ Inventory | Inventarista | ✅ inventory products | /Product |
| ⚪ Accountant | Contador | ✅ payments reports | /Home |
| 🔴 Support | Soporte Técnico | ✅ orders users | /Home |

🎯 ACCIONES ESPECÍFICAS DEL SISTEMA  
Apply to \_LoginLayout...

🛠️ CONTROLADORES CON AUTORIZACIÓN

|  |  |  |
| --- | --- | --- |
| CONTROLADOR | NIVEL DE ACCESO | MÉTODOS PROTEGIDOS |
| AuthController | Mixto | Profile, CurrentUser, CheckPermission |
| OrderController | 🔒 Completo | TODOS los métodos requieren autenticación |
| Otros | Sin restricciones | Acceso libre |

📊 ESTADO ACTUAL DE IMPLEMENTACIÓN

* ✅ IMPLEMENTADO:
* Sistema de roles con enum UserRole
* Autenticación por cookies (8 horas)
* Validación de permisos con HasPermissionAsync()
* Claims personalizados (UserId, UserRole, BranchId, etc.)
* Redirección automática por rol
* Usuario admin por defecto
* ❌ PENDIENTE DE IMPLEMENTAR:
* Atributos [Authorize(Roles = "admin")] en controladores específicos
* Middleware de autorización personalizado
* Interface para asignar roles a usuarios
* Control de acceso granular por vistas

🚀 PRÓXIMO PASO RECOMENDADO

1. Crear interfaz de gestión de usuarios para asignar roles
2. Implementar atributos de autorización en controladores específicos
3. Crear middleware personalizado para validación de permisos
4. Implementar control de acceso en vistas usando el rol del usuario