

## Laboratorio 19

### **Capacidad del modelo RBAC**

#### **Controlador centralizado de permisos:**

estos son los roles y los privilegios.

#### **Facilidad de administración:**

asignación de roles y gestión de roles.

#### **Seguridad mejorada:**

Acceso Basado en Necesidad

Auditoría y Cumplimiento:

### **Soporte del modelo en la base de datos (sql):**

-- Crear tablas para el modelo RBAC

CREATE TABLE Roles (

Id\_Rol INT PRIMARY KEY,

Descripcion VARCHAR(255)

);

CREATE TABLE Privilegios (

Id\_Privilegio INT PRIMARY KEY,

Descripcion VARCHAR(255),

Accion VARCHAR(255)

);

CREATE TABLE Usuarios (

Id\_Usuario INT PRIMARY KEY,

Nombre VARCHAR(255),

Apellidos VARCHAR(255),

Fecha\_Nacim DATE,

Fecha\_Crea DATE,

```
Balance DECIMAL(10, 2),  
Contraseña VARCHAR(255),  
Habilitado BOOLEAN  
);
```

```
CREATE TABLE Trabajadores (  
    Id_Usuario INT PRIMARY KEY,  
    RFC VARCHAR(13),  
    Sueldo DECIMAL(10, 2),  
    Seguro VARCHAR(255),  
    Telefono VARCHAR(10),  
    FOREIGN KEY (Id_Usuario) REFERENCES Usuarios(Id_Usuario)  
);
```

```
CREATE TABLE Areas_de_Trabajo (  
    Id_Area INT PRIMARY KEY,  
    Descripcion VARCHAR(255)  
);
```

```
CREATE TABLE Secciones (  
    Id_Seccion INT PRIMARY KEY,  
    Ubicacion VARCHAR(255)  
);
```

```
CREATE TABLE Transacciones (  
    Id_Trans INT PRIMARY KEY,  
    Tipo VARCHAR(255),  
    Descripcion VARCHAR(255)  
);
```

```
CREATE TABLE Generar (  
    Id_Usuario INT,  
    Id_Trans INT,  
    Fecha DATE,  
    Tipo VARCHAR(255),  
    Monto DECIMAL(10, 2),  
    PRIMARY KEY (Id_Usuario, Id_Trans),  
    FOREIGN KEY (Id_Usuario) REFERENCES Usuarios(Id_Usuario),  
    FOREIGN KEY (Id_Trans) REFERENCES Transacciones(Id_Trans)  
);
```

```
CREATE TABLE Tiene (  
    Id_Rol INT,  
    Id_Privilegio INT,  
    PRIMARY KEY (Id_Rol, Id_Privilegio),  
    FOREIGN KEY (Id_Rol) REFERENCES Roles(Id_Rol),  
    FOREIGN KEY (Id_Privilegio) REFERENCES Privilegios(Id_Privilegio)  
);
```

```
CREATE TABLE Posee (  
    Id_Usuario INT,  
    Id_Rol INT,  
    PRIMARY KEY (Id_Usuario, Id_Rol),  
    FOREIGN KEY (Id_Usuario) REFERENCES Usuarios(Id_Usuario),  
    FOREIGN KEY (Id_Rol) REFERENCES Roles(Id_Rol)  
);
```

**Obtener los permisos del usuario al autenticarse (javascript):**

```

const obtenerPermisos = async (usuarioid) => {
  const roles = await db.query(`
    SELECT R.Descripcion AS rol, P.Descripcion AS privilegio, P.Accion
    FROM Usuarios U
    JOIN Posee PO ON U.Id_Usuario = PO.Id_Usuario
    JOIN Roles R ON PO.Id_Rol = R.Id_Rol
    JOIN Tiene T ON R.Id_Rol = T.Id_Rol
    JOIN Privilegios P ON T.Id_Privilegio = P.Id_Privilegio
    WHERE U.Id_Usuario = ?`, [usuarioid]);

  return roles;
};

```

#### **Generar la interfaz gráfica de manera dinámica (archivo javascript):**

```

// Suponiendo que tienes los permisos en una variable `permisos`
if (permisos.includes('ver_dashboard')) {
  document.getElementById('dashboard').style.display = 'block';
} else {
  document.getElementById('dashboard').style.display = 'none';
}

```

#### **Validar permisos en cada ruta (archivo javascript):**

```

const verificarPermiso = (permiso) => {
  return (req, res, next) => {
    if (req.usuario.permisos.includes(permiso)) {
      next();
    } else {

```

```
        res.status(403).send('Acceso denegado');
    }
};

};

// Usar el middleware en una ruta
app.get('/dashboard', verificarPermiso('ver_dashboard'), (req, res) => {
    res.render('dashboard');
});
```

### **Crear interfaz de gestión de roles y permisos (archivo HTML):**

```
<form action="/asignar-rol" method="post">
    <label for="usuario">Usuario:</label>
    <select name="usuario" id="usuario">
        <!-- opciones de usuarios -->
    </select>
    <label for="rol">Rol:</label>
    <select name="rol" id="rol">
        <!-- opciones de roles -->
    </select>
    <button type="submit">Asignar Rol</button>
</form>
```