**Sistema de Administración de Seguridad en ASP.NET C#**

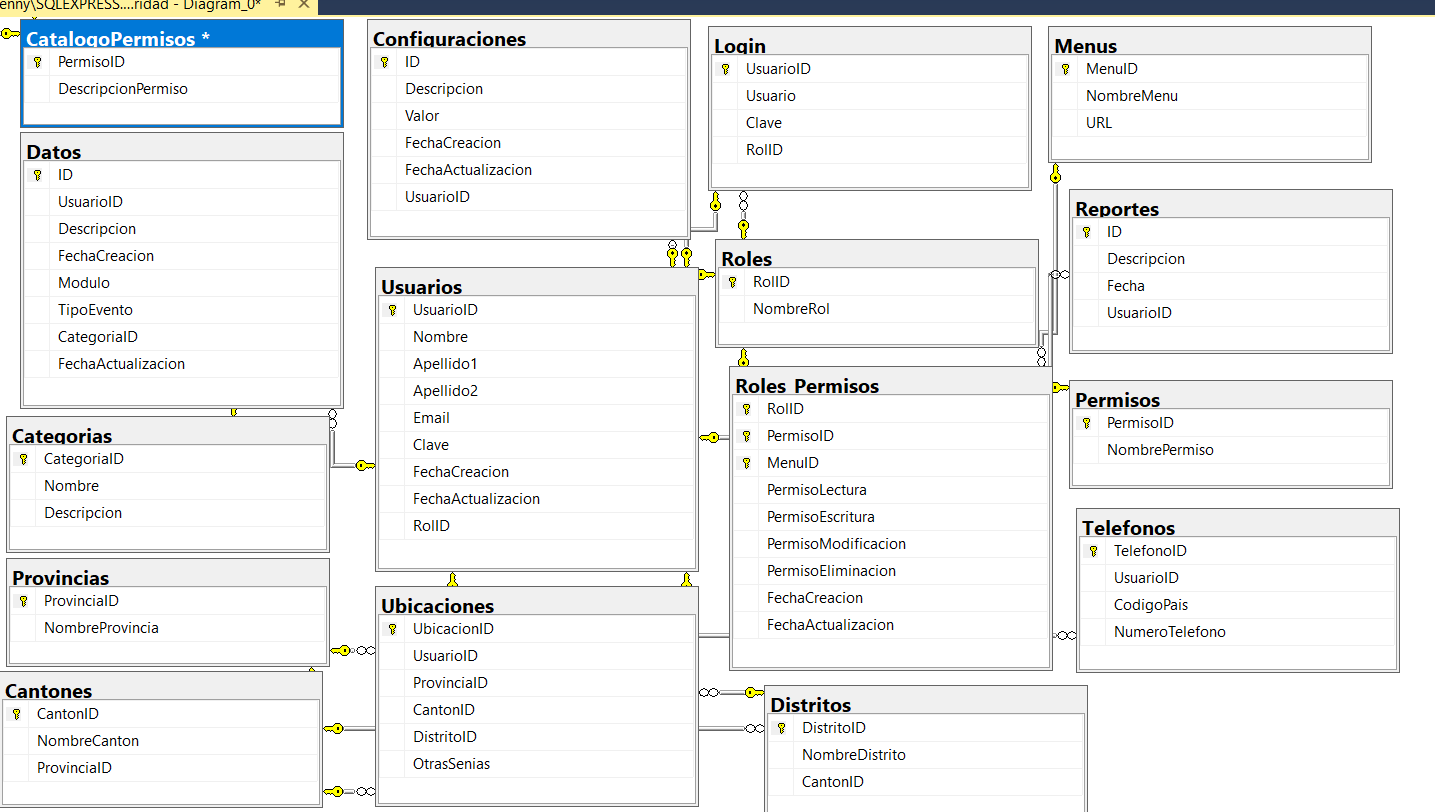
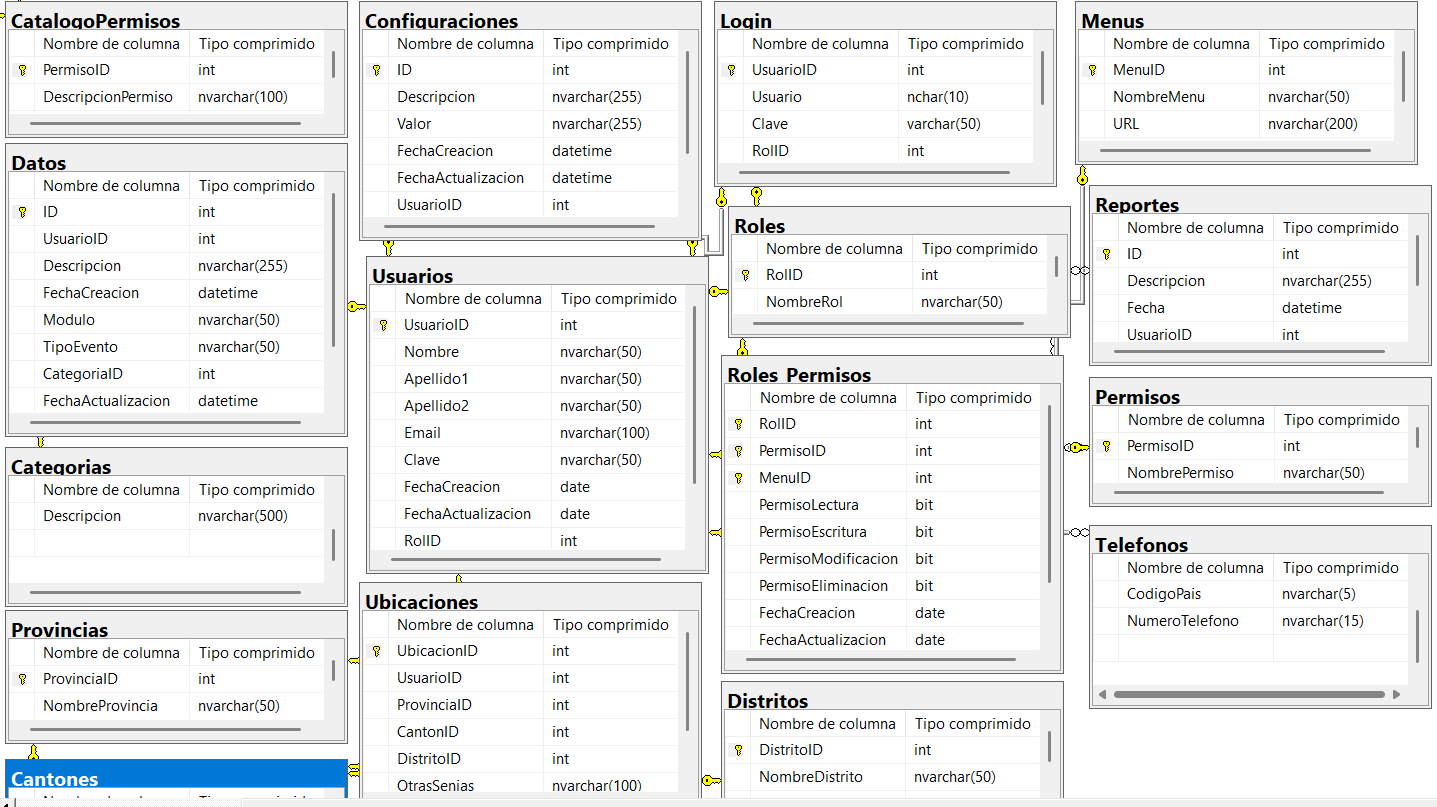
Jennifer Rebeca García Moncada

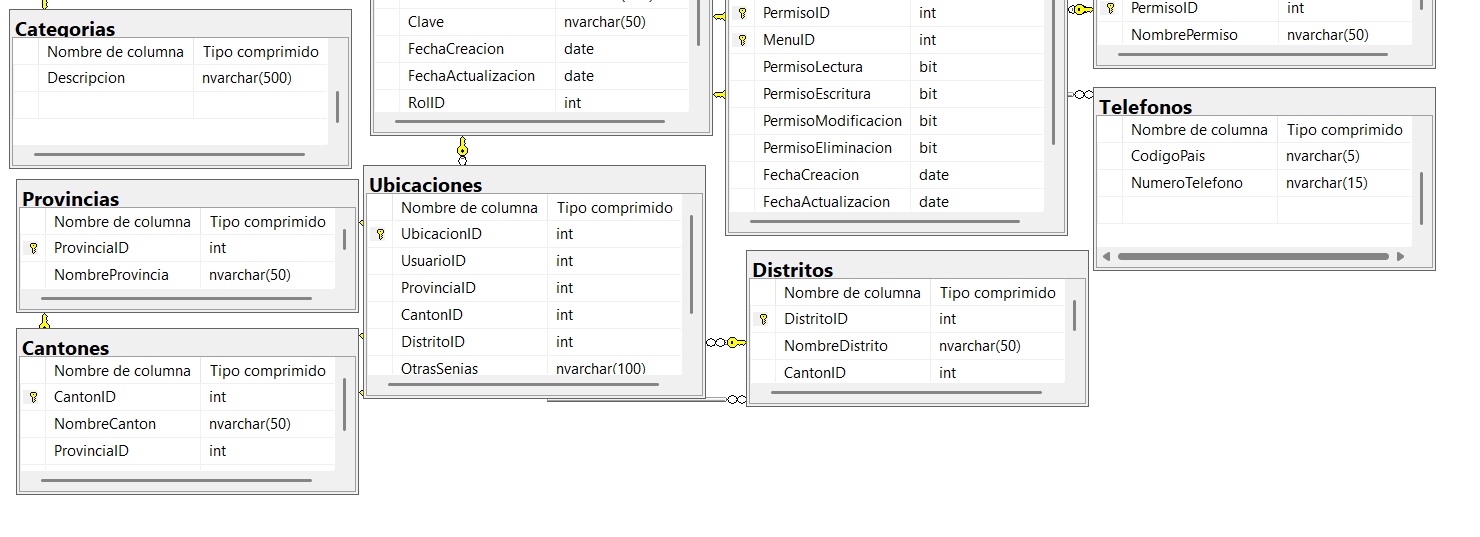
Universidad Politécnica Internacional

Programación III

Profesor: Alexander Benjamin Curling

Noviembre, 2024

Diagrama de la Base de Datos Normalizada



# Descripción del diseño

El sistema está diseñado con una arquitectura por capas, que separa la lógica de negocio, la presentación y el acceso a datos. Esto facilita la escalabilidad, el mantenimiento y la aplicación de los principios SOLID.

## Capa de Presentación: Se utilizan páginas ASP.NET Web Forms para la interfaz de usuario. Estas páginas contienen controles como TextBox, GridView, y DropDownList para capturar y mostrar información.

Capa de Lógica de Negocio: Implementa las reglas de negocio y valida los datos antes de interactuar con la base de datos. Aquí se encuentran métodos como GuardarUsuario y VerificarPermisos.

Capa de Acceso a Datos: Maneja las interacciones directas con la base de datos utilizando consultas SQL y procedimientos almacenados. Las clases como UsuarioDataAccess encapsulan esta lógica.

## Lógica de negocio:

El sistema incluye:

Gestión de Usuarios: Creación, lectura, actualización y eliminación (CRUD) de usuarios, con validación de datos.

Control de Permisos: Basado en roles definidos en la base de datos, como Administrador, Usuario, Supervisor y Auditor.

Gestión de Ubicaciones: Los usuarios se asocian con una provincia, cantón y distrito, pudiendo introducir estos últimos manualmente.

# Aplicación de los principios SOLID

**Responsabilidad Única (SRP):** Cada clase tiene una única responsabilidad. Por ejemplo, UsuarioDataAccess solo maneja interacciones con la base de datos para los usuarios.

Abierto/Cerrado (OCP): Las clases están abiertas para extensión, pero cerradas para modificación. Por ejemplo, se pueden agregar nuevos roles sin modificar la lógica principal.

Sustitución de Liskov (LSP): Las subclases pueden sustituir a sus superclases. Si se utiliza herencia, como en Usuario y Administrador, mantienen coherencia con la clase base.

Segregación de Interfaces (ISP): Las interfaces dividen funcionalidades específicas, evitando la sobrecarga. Por ejemplo, métodos específicos para permisos y usuarios.

Inversión de Dependencia (DIP): Las clases de negocio no dependen directamente de la base de datos, sino que interactúan a través de interfaces o clases de acceso a datos.

# Instrucciones de Configuración y Ejecución

## Requisitos Previos: Se requiere de software necesario:

1. Visual Studio (2019 o posterior) con ASP.NET habilitado.
2. SQL Server Management Studio (SSMS) para la gestión de la base de datos.
3. Navegador web moderno (Chrome, Edge o Firefox).

## Base de Datos:

Se debe de asegurar que la base de datos AdminSeguridad se encuentre creada.

Se debe de importar las tablas necesarias (Roles, Usuarios, Login, etc.) y llena los datos iniciales (roles como Administrador, Auditor, etc.).

## Configuración del Proyecto:

1. **Configura la cadena de conexión:**

En el archivo Web.config o en el código de la capa de acceso a datos, ajustar la cadena de conexión para que apunte a tu servidor SQL:

<connectionStrings>

<add name="AdminSeguridadDB" connectionString="Data Source=TU\_SERVIDOR;Initial Catalog=AdminSeguridad;Integrated Security=True" providerName="System.Data.SqlClient" />

</connectionStrings>

1. **Revisar las dependencias:**

Se debe de asegurar que las referencias a System.Data.SqlClient y otros paquetes necesarios estén configuradas correctamente.

1. **Carga inicial de datos:**

Usa scripts SQL para llenar tablas base como Roles y Permisos:

INSERT INTO Roles (NombreRol) VALUES ('Administrador'), ('Auditor'), ('Supervisor'), ('Usuario');

1. **Configura las sesiones:**

Se debe de asegurar de que las páginas ASP.NET verifican la sesión:

if (Session["Rol"] == null)

{

Response.Redirect("Login.aspx");

}

## Ejecución:

1. **Compilar y ejecutar el proyecto:**

Abre el proyecto en Visual Studio.

Configura la página Login.aspx como la página de inicio:

* Haz clic derecho sobre Login.aspx en el Explorador de Soluciones.
* Selecciona Establecer como página de inicio.

Presiona F5 o selecciona Iniciar sin depuración para compilar y ejecutar el proyecto.

1. **Iniciar sesión:**

Ingresa con las credenciales de un usuario registrado (por ejemplo, admin).

Dependiendo del rol del usuario, serás redirigido automáticamente a las páginas correspondientes según sus permisos:

* + **Administrador:** Acceso completo a todas las páginas.
  + **Usuario:** Acceso limitado según el rol y permisos asignados.

1. **Validación de permisos:**

Si intentas acceder a una página sin los permisos adecuados, el sistema debe redirigirte a una página de acceso denegado, como **SinPermiso.aspx**.

1. **Prueba de funcionalidades:**

Prueba las acciones CRUD en la página de gestión de usuarios.

Cambia los roles y verifica los permisos.

## Resolución de Problemas:

Si no carga la base de datos:

* + Revisa la conexión en el archivo Web.config o el string en el código.

Si los permisos no se aplican correctamente:

* + Verifica que las tablas Permisos y Roles están correctamente llenas.