Paquete de Seguridad de DoltSharp

1. Autenticación de Usuarios:

* Implementar la autenticación utilizando el framework ASP.NET Identity para manejar el registro, login y autenticación de usuarios de manera segura. Esto incluye:
  + Almacenamiento de contraseñas de manera segura con hashing (por ejemplo, bcrypt) utilizando las clases disponibles en ASP.NET Core Identity.
  + Configuración de autenticación de múltiples factores (MFA) mediante la integración de Google Authenticator o servicios como Twilio para la verificación por SMS.

1. Control de Acceso Basado en Roles (RBAC):

* Definir roles y permisos utilizando ASP.NET Core Authorization. Esto permite restringir el acceso a diferentes partes de la aplicación según el rol del usuario.

1. Cifrado de Datos:

* Usar AES para cifrar y descifrar información sensible. Se puede utilizar el espacio de nombres System.Security.Cryptography para implementar este cifrado.

1. Protección contra Inyección SQL:

* Utilizar Entity Framework Core o Dapper para ejecutar consultas parametrizadas, lo que ayuda a prevenir ataques de inyección SQL.

1. Prevención de Ataques de Fuerza Bruta:

* Implementar un sistema de recuento de intentos de login fallidos y bloquear el acceso tras un número determinado de intentos, utilizando ASP.NET Core Identity.

1. Registro y Monitoreo de Actividades (Logging):

* Implementar el logging utilizando el sistema de ASP.NET Core Logging. Se puede configurar para guardar los logs en archivos o bases de datos.

1. Seguridad Física (Acceso a Servidores):

* Aunque la seguridad física no se implementa directamente en C#, es importante que la aplicación esté desplegada en servidores seguros y se monitoreen los accesos a estos servidores.
* En C#, podrías usar servicios como Azure Security Center o AWS CloudWatch para monitorizar la seguridad de la infraestructura.

1. Pruebas de Penetración y Auditoría de Seguridad:

* Usar herramientas como OWASP ZAP para hacer pruebas de seguridad en aplicaciones web ASP.NET.
* En C#, las pruebas de seguridad se pueden integrar en pipelines de CI/CD mediante herramientas como SonarQube, que analiza vulnerabilidades de código.

1. Recuperación ante Desastres (Disaster Recovery):

* Los backups de la base de datos pueden automatizarse con Entity Framework o mediante scripts SQL. Para restaurar bases de datos, puedes usar la funcionalidad integrada en Azure SQL o SQL Server.