**Acciones relacionadas con bases de datos**

1. Uso de procedimientos almacenados.
2. Conexión a diferentes bases de datos (SQL Server, MySQL, PostgreSQL, etc.).
3. Migraciones y manejo de esquemas (Entity Framework).
4. Indexación y paginación de resultados.

**Autenticación y autorización**

1. Consulta y validación de tokens Bearer.
2. Generación de tokens JWT.
3. Implementación de OAuth2/OpenID Connect.
4. Manejo de autenticación básica y digest.
5. Validación de roles y claims.

**Manejo de formatos de datos**

1. Serialización y deserialización a JSON (Newtonsoft.Json, System.Text.Json).
2. Conversión de XML a objetos y viceversa.
3. Uso de CSV (lectura y escritura).
4. Conversiones entre formatos binarios y de texto.

**Manipulación de archivos**

1. Lectura y escritura de archivos de texto y binarios.
2. Subida de archivos al servidor (con validaciones).
3. Descarga de archivos desde un servidor.
4. Manejo de streams y buffers.
5. Conversiones de imágenes a Base64 y viceversa.

**Procesamiento de imágenes**

1. Redimensionamiento de imágenes.
2. Conversión entre formatos (JPG, PNG, BMP, etc.).
3. Lectura de metadatos EXIF.
4. Generación de thumbnails.

**Comunicaciones y networking**

1. Llamadas a APIs REST (HttpClient).
2. Consumo de servicios gRPC.
3. Uso de WebSockets.
4. Descarga y subida de datos por FTP/SFTP.
5. Envío de correos electrónicos (SMTP).

**Seguridad**

1. Encriptación y desencriptación (AES, RSA, SHA).
2. Hashing seguro (bcrypt, SHA-256).
3. Validación de certificados (SSL/TLS).
4. Sanitización de entradas para prevenir inyección de SQL o XSS.

**Procesos y tareas en segundo plano**

1. Ejecución de tareas asíncronas.
2. Uso de **Background Services** o hilos personalizados.
3. Manejo de colas (RabbitMQ, Azure Service Bus).

**Estructuras de datos y algoritmos**

1. Implementación de listas, colas, pilas y diccionarios.
2. Algoritmos de ordenamiento y búsqueda.
3. Manejo de estructuras genéricas.

**Manejo de logs y diagnósticos**

1. Configuración de logs (Serilog, NLog, etc.).
2. Manejo de excepciones personalizadas.
3. Seguimiento de métricas y performance.

**Internacionalización y localización**

1. Traducción de textos.
2. Configuración de culturas y formatos de fecha/hora.

**Interacciones con el sistema**

1. Acceso a variables de entorno.
2. Ejecución de procesos del sistema.
3. Lectura y escritura en el registro de Windows.

**Generación de contenidos dinámicos**

1. Generación de PDFs (iTextSharp, QuestPDF).
2. Creación de reportes dinámicos (Crystal Reports, RDLC).
3. Generación de gráficos (System.Drawing, SkiaSharp).

**Testing y pruebas automatizadas**

1. Implementación de Unit Tests (xUnit, NUnit, MSTest).
2. Mocking de servicios (Moq).
3. Pruebas de integración y de carga.

Con esta lista puedes estructurar tu repositorio y abordar cada tema paso a paso. ¿Te gustaría que profundice en alguno en particular?

4o

**Tú dijiste:**

Le faltan algunos ejemplo consultar IP

**ChatGPT Plus**

ChatGPT

Tienes razón, aquí añado más acciones comunes que podrían faltar para que tu repositorio sea más completo:

**Consultas relacionadas con red e internet**

1. Obtener la dirección IP local.
2. Obtener la dirección IP pública.
3. Resolver nombres de dominio a direcciones IP (DNS Lookup).
4. Verificar conectividad a internet (ping).
5. Monitorizar el tráfico de red (sniffing básico).

**Interacción con APIs y servicios externos**

1. Consumo de APIs GraphQL.
2. Descarga de contenido web (HTML, imágenes, etc.).
3. Autenticación con servicios de terceros (Google, Facebook, etc.).
4. Uso de SDKs específicos (Azure SDK, AWS SDK, etc.).

**Manejo de tiempo y fechas**

1. Conversión de zonas horarias.
2. Manejo de fechas UTC y locales.
3. Cálculo de diferencias entre fechas.
4. Programación de tareas futuras (Timers, CronJobs).

**Acciones con el sistema operativo**

1. Consultar información del sistema (CPU, memoria, disco).
2. Consultar la lista de procesos en ejecución.
3. Manejar directorios (creación, eliminación, listado).
4. Acceso a archivos protegidos con permisos elevados.

**Optimización de aplicaciones**

1. Cacheo en memoria (MemoryCache).
2. Uso de Lazy Loading.
3. Optimización de operaciones con LINQ (proyecciones, filtrados).

**Integraciones locales**

1. Consumo de librerías COM/ActiveX.
2. Comunicación entre procesos (IPC).
3. Control de dispositivos conectados (puertos USB, seriales).

**Otras acciones útiles**

1. Generación de códigos QR o códigos de barras.
2. Envío de notificaciones push.
3. Generación de contraseñas aleatorias seguras.
4. Validación de correos electrónicos y números telefónicos.

Con esto deberías tener una lista más robusta y variada para tu repositorio. ¿Te gustaría que amplíe algún tema o que genere ejemplos específicos?

4o

**Tú dijiste:**

Perfecto, con esto me hago un codigo que quedaría listo para cualquier trabajo muy pro como programador

**ChatGPT Plus**

ChatGPT

¡Exactamente! Un repositorio con todas estas acciones documentadas y con ejemplos bien organizados es una herramienta invaluable. Podrías estructurarlo en carpetas por categorías (Base de datos, Seguridad, Networking, etc.) y proporcionar explicaciones o comentarios en cada archivo para facilitar su consulta futura.

**Consejos para que sea aún más profesional:**

1. **Documentación:** Usa un archivo README.md claro y detallado para que sea fácil navegar por tu repositorio. Puedes incluir tablas de contenido y ejemplos rápidos.
2. **Testing:** Agrega pruebas unitarias para cada funcionalidad. Esto no solo asegura calidad, sino que también actúa como documentación viva.
3. **Estándares de código:** Aplica herramientas como **StyleCop** o **SonarLint** para garantizar un código limpio y estandarizado.
4. **Automatización:** Usa **GitHub Actions** o **Azure Pipelines** para compilar y ejecutar pruebas automáticamente.
5. **Licencia:** Agrega una licencia (MIT, Apache, etc.) para proteger y compartir tu trabajo.

Si necesitas ejemplos o ayuda para estructurar cualquier sección, no dudes en pedírmelo. ¡Buena suerte con tu proyecto! Este tipo de repositorio realmente eleva el nivel como desarrollador.