**Universidad Nacional de La Matanza**

**Departamento de Ingeniería e Investigaciones Tecnológicas**

**Bases de Datos Aplicada**

**Grupo: [3]**

**TP Integrador**

**Fecha de Presentación: [01/07/2025]**

**Integrantes:**

|  |  |  |  |
| --- | --- | --- | --- |
| **DNI** | **Apellido** | **Nombre** | **Email** |
| 44.391.352 | Codina | Santiago Iván | Scodina352@alumno.unlam.edu.ar |
| 45.175.053 | Santillán | Lautaro | lsantillan053@alumno.unlam.edu.ar |
|  |  |  |  |
|  |  |  |  |

Contenido

[**Entrega 1** 3](#_Toc203168232)

[Informe I 3](#_Toc203168233)

[SQL Server 4](#_Toc203168234)

[Requisitos técnicos por cubrir 4](#_Toc203168235)

[Perfiles técnicos 5](#_Toc203168236)

[Seguridad Informática 6](#_Toc203168237)

[Costos 6](#_Toc203168238)

[Detalle de costos de perfiles técnicos 6](#_Toc203168239)

[Costo del soporte técnico del motor 7](#_Toc203168240)

[Costo de licencia 7](#_Toc203168241)

[Costos totales 7](#_Toc203168242)

[Conclusiones 7](#_Toc203168243)

[**Entrega 2** 9](#_Toc203168244)

[Informe II 9](#_Toc203168245)

[Servicios de Cloud Computing 9](#_Toc203168246)

[Bibliografía 10](#_Toc203168247)

[**Entrega 3** 11](#_Toc203168248)

[Versión 1: 11](#_Toc203168249)

[Versión 2: 12](#_Toc203168250)

[Versión 3: 14](#_Toc203168251)

[**Entrega 4:** 18](#_Toc203168252)

[Cómo instalar SQL Server: 18](#_Toc203168253)

[**CONSIDERACIÓN ADICIONAL:** 18](#_Toc203168254)

[Glosario 19](#_Toc203168255)

[Funciones: 23](#_Toc203168256)

[**Entrega 7** 24](#_Toc203168257)

[Políticas de Respaldo y Encriptación de Datos 24](#_Toc203168258)

# **Entrega 1**

## Informe I

Proyecto

Sol Norte: Digitalización

Estimación de costos

Fecha de presentación: 16/06/2025

Grupo: 03

Cliente: Sol Norte

Motor: SQL Server

Contenido

[Entrega 1 3](#_Toc201002056)

[Informe I 3](#_Toc201002057)

[SQL Server 4](#_Toc201002058)

[Requisitos técnicos por cubrir 4](#_Toc201002059)

[Perfiles técnicos 5](#_Toc201002060)

[Seguridad Informática 6](#_Toc201002061)

[Costos 6](#_Toc201002062)

[Detalle de costos de perfiles técnicos 6](#_Toc201002063)

[Costo del soporte técnico del motor 7](#_Toc201002064)

[Costo de licencia 7](#_Toc201002065)

## SQL Server

## Requisitos técnicos por cubrir

* La máquina principal (aquella descrita en la consigna) podrá correr el motor sin problemas, pues usaremos la aplicación de SQL Server Management Studio (SMSS) para ejecutarlo y trabajar nuestra base de datos. Este es un entorno integrado para manejar interfaces con SQL junto con otros servicios como Azure SQL Database y similares. Lo necesario para poder manejarlo es:
  + Cómo sistema operativo, se recomienda tanto un Windows 10 1607 o mayor como una instancia de Windows Server 2016 o mayor.
    - Para SQL Server en sí, es necesaria una versión del Framework .NET 4.7.2.
  + Un procesador de arquitectura x64 de cuatro núcleos. Específicamente, no se soportan aquellos de arquitectura Arm64 y Arm32.
  + 4 Gigabytes de RAM mínimo. Para soluciones profesionales como la que se tratará, dieciséis es recomendable.
  + 4~50 Gigabytes de espacio en Disco. Con respecto a los datos almacenados, el disco que se va a usar contará con más del espacio necesario para almacenar todos los datos necesarios según las especificaciones dadas en el informe del trabajo. ○ A pesar de ser más rígido en su aplicación (es un entorno pensado para S.O Windows en SQL Server), cuenta con el constante soporte de Windows en sí, con tres años más de actualizaciones de Calidad de Vida para la versión de SQL Server 2022 y cinco años posteriores de soporte extendido (ósea, de seguridad) antes de que se corte completamente.
* Otra opción como entorno de desarrollo para manejar SQL Server sería Devart, que permite a tanto desarrolladores como administradores de Bases de Datos acceder a varias herramientas para realizar sus labores con SQL Server.
  + Un punto a favor sobre SMSS es qué puede emplearse en varios sistemas operativos, desde Windows 7-11, Windows Server 2008-2022, varias distribuciones de Linux e incluso algunos modelos de macOS a través de Crossover.
  + También puede correr en procesadores ARM64. Todos los demás requisitos de hardware descritos para SMSS aplican para Devart.
  + Las licencias anuales para Devart son bastante más baratas que las de SMSS, siendo solo cientos de dólares.
* DBeaver es un entorno pensado para DBAs y desarrolladores con un gran foco en usabilidad. ○ Es multiplataforma, similar a Devart. Donde SQL Server opera principalmente en Windows, DBeaver funciona en Windows, macOS y Linux.
  + También da flexibilidad al desarrollador al admitir una gran variedad de lenguajes, mientras que SMSS tiene como foco principal SQL.
    - Sumado a esto, las conexiones admitidas por este entorno son mayores. SMSS está pensado principalmente para SQL Server, por lo que se tiene menos flexibilidad que con entornos menos dedicados.
  + El mayor beneficio de DBeaver, al ser Open-Source (de Código Abierto), es ser gratuito. Donde SMSS y Devart se compran con licencia (que puede costar varios miles de dólares en un entorno empresarial), DBeaver es gratis y sin comprometer ninguna función esencial de desarrollo para nuestro proyecto.

Aclaración: Todos los datos puestos en esta sección fueron sacados de la documentación oficial de los entornos de desarrollo.

## Perfiles técnicos

* Un Administrador de Base de Datos será necesario para gestionar los requisitos, privilegios y la seguridad de esta. Necesitaremos que sea Semi-Senior, cuanto menos, pues tendrá que estar aclimatado al entorno del desarrollo de software y ser capaz de lidiar con el entorno cambiante de trabajo.
* Un Diseñador de Bases de Datos para poder relevar las necesidades relevadas por el DBA en el entorno de trabajo elegido. Este también deberá ser Semi-Senior, pues su trabajo será una piedra angular que guiará el modelaje del proyecto y sus bases de datos. Desde sus diseños, los desarrolladores harán el trabajo y el Tester llevará a cabo las pruebas de estrés pertinentes.
* Dos Desarrolladores de Bases de Datos para poder llevar a cabo y testear los diseños planteados por el DBD en el entorno elegido, además de dar soporte y mantenimiento al sistema una vez implementado. Su seniority puede ser junior, pues contamos con gente capaz arriba suyo que será capaz de guiarlos acorde según la necesidad del trabajo a realizar.
* Un Supervisor de Seguridad Informática, que tomará el rol de asegurarse de que no haya tráfico malicioso en la red de Sol Norte, además de que la información de los usuarios esté siendo protegida correctamente.
* Un Tester, que tomará el trabajo de asegurarse de la calidad producida por los desarrolladores. También tomará el rol de hacer mantenimiento al código, dando correcciones exactas a los miembros del proyecto sobre las consecuencias de dichos errores y las consecuencias que pueden llegar a tener.

|  |  |  |  |
| --- | --- | --- | --- |
| Perfil | Cantidad de personas | Valor hora de trabajo | Total |
| Administrador de Base de Datos | 1 | USD $11 | US$1900 mensuales. |
| Diseñador de Bases de Datos. | 1 | USD $10 | US$1700 mensuales. |
| Desarrollador de Bases de Datos | 2 | USD $8,5 | US$1500 mensuales por individuo. |
| Tester | 1 | USD$ 8,5 | US$1500 mensuales. |
| Supervisor de Seguridad Informática | 1 | USD$ 14.42 | US$ 2500 mensuales |

Referencia:<https://sueldos.openqube.io/encuesta-sueldos-2025.01/>

## Seguridad Informática

SQL Server ofrece un conjunto completo de mecanismos de seguridad para proteger la información en distintos niveles. Entre sus principales características se incluyen:

1. Seguridad en la plataforma: Limita el acceso físico y lógico al servidor, asegurando el sistema operativo y reduciendo los servicios expuestos para minimizar posibles vulnerabilidades.
2. Autenticación segura: Permite controlar quién accede al sistema mediante autenticación de Windows o mixta, recomendando el uso de credenciales del sistema operativo por su mayor robustez.
3. Autorización y protección de datos: Utiliza permisos, roles y cifrado para proteger los objetos de base de datos (tablas, vistas, procedimientos) y la información almacenada. También permite enmascarar datos sensibles y aplicar seguridad a nivel de filas.
4. Seguridad en las aplicaciones: Refuerza la protección mediante validación de entradas, conexiones cifradas y prácticas de desarrollo seguro para evitar ataques como inyecciones SQL.

## Costos

Los costos que se necesitan estimar son los que se encuentran en la tabla.

## Detalle de costos de perfiles técnicos

* Un Administrador de Base de Datos será necesario para gestionar los requisitos, privilegios y la seguridad de esta.
* Un Diseñador de Bases de Datos para poder relevar las necesidades relevadas por el DBA en el entorno de trabajo elegido.
* Dos Desarrolladores de Bases de Datos para poder llevar a cabo y testear los diseños planteados por el DBD en el entorno elegido, además de dar soporte y mantenimiento al sistema una vez implementado.

|  |  |  |
| --- | --- | --- |
| Perfil | Cantidad de personas | Sueldo anualizado ($USD) |
| Administrador de Base de Datos | 1 | $22.800usd ($1900usd mensuales) |
| Diseñador de Bases de Datos. | 1 | $20.400usd ($1700usd mensuales) |
| Desarrollador de Bases de Datos | 2 | $36.000usd ($1500usd mensuales c/u) |

## Costo del soporte técnico del motor

SQL Server en su versión estándar ofrece soporte en línea, documentación en línea, recursos de aprendizaje y foros de la comunidad de manera gratuita. Aunque también ofrece soporte adicional pago, el cual sirve para problemas específicos, con un valor de $499 por cada incidente, y soporte técnico premier para socios.

## Costo de licencia

Para nuestro proyecto decidimos utilizar SQL Server versión standard: servidor + CAL, ya que esta licencia sirve para proyectos chicos donde se sabe cuántos usuarios van a acceder a la base de datos, en nuestro caso compraremos una licencia servidor ($989) y 10 licencias CAL para cada uno de los puestos de acceso ($230c/u), dándonos un total de $3289. Esta opción es mucho más económica que la versión por núcleos, la cual se utiliza para proyectos grandes donde no es posible saber la cantidad de usuarios de la base de datos, se trata de un modelo de paquete de 2 núcleos, al contar nosotros con un procesador de 4 núcleos, se necesitan 2 licencias de $3945c/u, dando un total de $7890.

## Costos totales

|  |  |
| --- | --- |
|  | Importe total $UDS anual (expresado en moneda Dólar americano) |
| Costos del personal necesario para la implementación | $79.200 |
| Costo del soporte técnico del motor. | Gratuito. |
| Cantidad de horas Soporte técnico si las ofrece y el costo de estas | 0 |
| Costo de Licencia | $3.289 |
| Total $UDS | $82.489usd |

## Conclusiones

Luego de analizar los requerimientos funcionales y técnicos del sistema de gestión administrativa para la institución deportiva Sol Norte, se concluye que el equipo utilizará SQL Server como motor de base de datos para la implementación del sistema. La elección se justifica en base a las siguientes razones:

* Seguridad robusta: SQL Server ofrece mecanismos avanzados de control de acceso, cifrado de datos sensibles y auditoría de acciones, garantizando la protección de la información de los socios y las operaciones financieras.
* Estructuras para reglas de negocio: Las funcionalidades como descuentos automáticos, gestión de morosidad, reintegros por condiciones climáticas y generación de facturación personalizada pueden implementarse eficientemente mediante procedimientos almacenados y funciones dentro del motor.
* Integración con plataformas externas: Facilita la conexión con sistemas de facturación electrónica (AFIP) y pasarelas de pago como MercadoPago, aspecto clave para eliminar el uso de efectivo.
* Herramientas de administración y monitoreo: SQL Server incluye entornos como SQL Server Management Studio que permiten una gestión eficiente del sistema por parte del equipo técnico.

Nosotros como equipo coincidimos en que SQL Server es una buena herramienta para que la institución Sol Norte lo use. Pero también podríamos sugerir como motor alternativo a **PostgreSQL**, una base de datos relacional de código abierto y de alta disponibilidad que se ha posicionado como una de las mejores opciones en la industria del software. Esta elección se basa en los siguientes fundamentos clave:

##### 1. Costo cero de licencias y escalabilidad económica

PostgreSQL es un motor completamente gratuito, incluso para uso comercial. Esto significa que: no hay costos por instalación, por número de usuarios ni por núcleos del servidor. Para un proyecto como el de Sol Norte, que podría experimentar crecimiento en cantidad de usuarios o servicios en el futuro, este beneficio es estratégico, ya que elimina barreras presupuestarias para escalar.

##### 2. Multiplataforma y adaptable a cualquier entorno

PostgreSQL se ejecuta de manera nativa en múltiples sistemas operativos (Linux, Windows, MacOS), y se integra perfectamente en entornos on-premise y cloud (AWS, Azure, GCP).

##### 3. Robustez y soporte de características avanzadas

PostgreSQL cuenta con un motor muy maduro que incluye funcionalidades que incluso no están presentes en versiones estándar de SQL Server como soporte avanzado para JSON y almacenamiento híbrido relacional/no relacional, y funciones personalizadas, triggers, vistas materializadas, índices avanzados (GIN, GiST, SP-GiST).

##### 4. Seguridad y respaldo de una gran comunidad

PostgreSQL permite una gestión segura de usuarios y privilegios y tiene un ecosistema robusto, con actualizaciones regulares y soporte profesional a través de múltiples empresas proveedoras.

Por todo lo anterior, PostgreSQL no solo representa una alternativa viable, sino una elección estratégica, moderna, segura y sostenible a largo plazo para el desarrollo del sistema de gestión administrativa requerido por Sol Norte.

# **Entrega 2**

## Informe II

## Servicios de Cloud Computing

**Amazon Web Services (AWS):** Servicio de Alta Disponibilidad (Contamos con una localización de puerto en CABA; Cirion, Buenos Aires).

Esto significa que podemos contar con infraestructuras cerca de los usuarios finales.

* Buenos protocolos de seguridad, indispensables para la información sensible que vamos a manejar.
* Amplio portfolio de servicios, que también viene de la mano con su modelo de pago “Pay-as-you-go”, que optimiza el uso del presupuesto y evita desperdicio de capacidades.
* Los servicios dados por Amazon tienen una fiabilidad de 99.9% o superiores, dicho por ellos mismos. Con más de treinta zonas distribuidas alrededor del mundo, fallos locales podrían ser mitigados, implementando la misma arquitectura en otra zona.
* Con los servicios de Amazon Direct Connect y el paquete de Amazon EC2, el total de mantenimiento de AWS agregaría unos USD$2620 mensuales, aproximadamente.
  + Especificaciones del Amazon Direct Connect: Number of ports (1), Location (Cirion BNARAGMS, Buenos Aires), Port type (Dedicated), Port capacity (10G), Data transfer out (10 GB), Data transfer in (gratis) (10 GB)
  + Especificaciones de Amazon EC2: Tenancy (Shared Instances), Operating system (Linux), Workload (Consistent, Number of instances: 10), Advance EC2 instance (t3.xlarge), Pricing strategy (EC2 Instance Savings Plans 3yr).
* Usamos 10 instancias, una para cada puesto, y diez gigas de capacidad de transferencia para poder lidiar tanto con actualizaciones de software como con tráfico convencional.

**Microsoft Azure:** Azure Hybrid Benefit significa que, nosotros al trabajar con SQL Server, tendremos beneficios a la hora de presupuestar el software de Azure.

* Azure cuenta con muchísimas más localidades que AWS (un total de 170), con tiempos de SLA que rondan el 99.9% también, incluso presentando mayor rendimiento que la anterior.
* Además de contar con modelos “Pay-as-you-go”, Azure permite pagar por lo que se usa y hacer “reservas” del software, que también reduce el precio por mes.
* Más allá de eso, Azure viene con soporte incluido, mientras que AWS no. o Los servicios contratados serán: 10 Equipos D2ds4 (2 nucleos virtuales, 8 gigas de RAM y una licencia SQL Enterprise, pay-as-you-go) y una Base de Datos Única de 1 tera, reservada por un año y con copias diarias, mensuales y anuales de los datos
  + En nuestro caso, el tema de precios (diez instancias para los equipos y una base de datos de 1 TB) es 100% apoyado por las licencias de SQL Server Híbridas con Azure, pues las diez instancias de VMs costarán USD$1700 y USD$1800 mensuales respectivamente.

**Google Cloud:**

Para este servicio en la nube utilizamos la instancia db-standar-2, la cual cuenta con 2 vCPUs y 7.5GiB de memoria. La edición de la nube de SQL es la Enterprise, con ubicación del servidor en Santiago. Por último el almacenamiento calculado para la base de datos es de 100GB con un almacenamiento para las copias de seguridad de 25GB. Estas características tienen un costo de 702,62 USD (8431,44 USD Anuales) por instancia, que en nuestro caso sería suficiente solo una, estando 24/7 operativa.

## Bibliografía

[https://learn.microsoft.com/en-us/sql/sql-server/install/hardware-and-software-requirements-for-install ing-sql-server-2022?view=sql-server-ver16](https://learn.microsoft.com/en-us/sql/sql-server/install/hardware-and-software-requirements-for-installing-sql-server-2022?view=sql-server-ver16) <https://docs.devart.com/studio-for-mysql/getting-started/requirements.html><https://dbeaver.io/download/>

[https://www.microsoft.com/es-ar/sql-server/sql-server-2022-pricing#x356c8eb5743c4651874e1870114 ab204](https://www.microsoft.com/es-ar/sql-server/sql-server-2022-pricing#x356c8eb5743c4651874e1870114ab204) <https://partner.microsoft.com/es-ar/support/microsoft-services-premier-support>

[https://learn.microsoft.com/es-es/troubleshoot/sql/database-engine/install/windows/support-policy-sql](https://learn.microsoft.com/es-es/troubleshoot/sql/database-engine/install/windows/support-policy-sql-server)

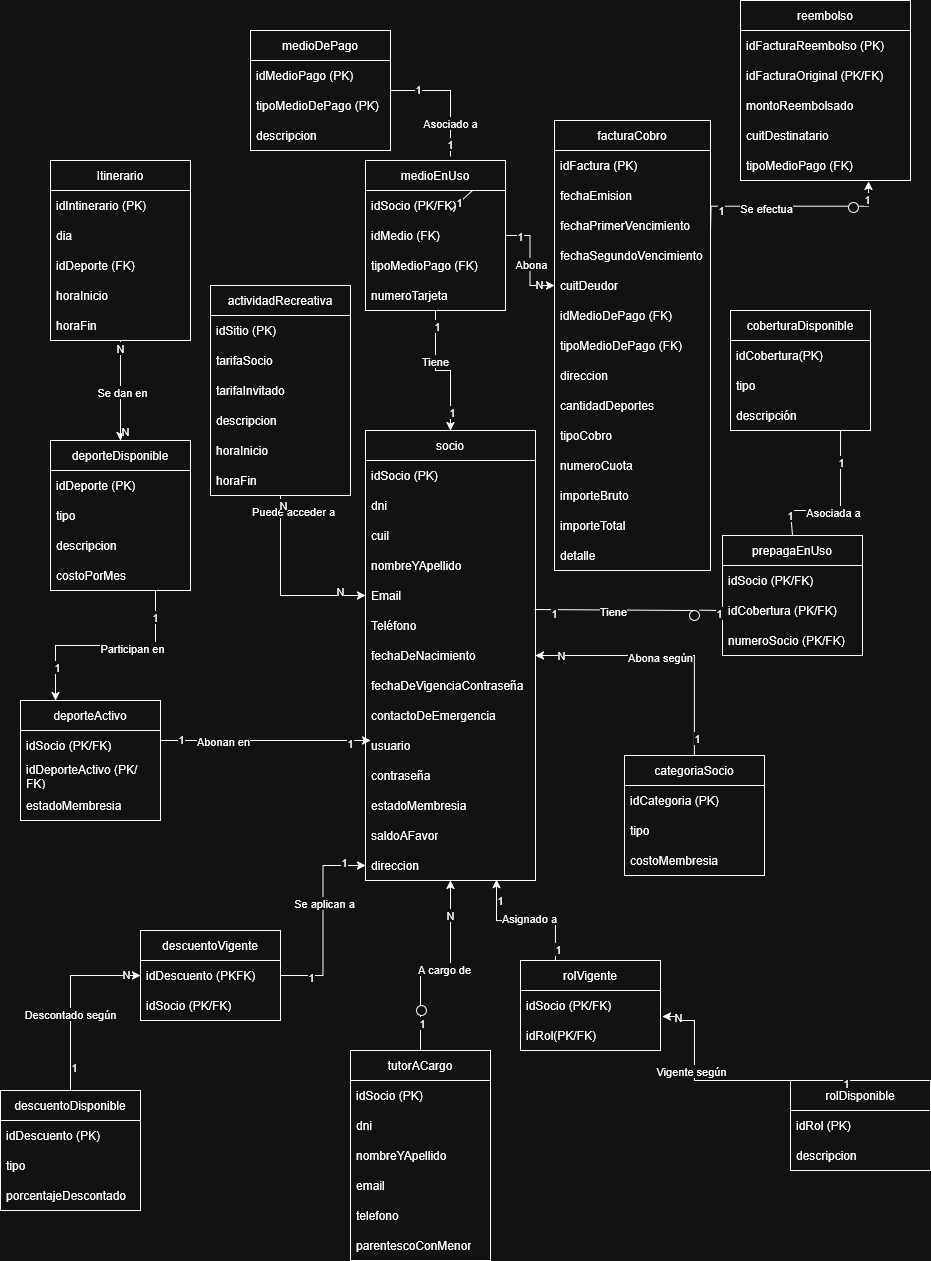
[-server](https://learn.microsoft.com/es-es/troubleshoot/sql/database-engine/install/windows/support-policy-sql-server)

[https://learn.microsoft.com/es-es/sql/relational-databases/security/securing-sql-server?view=sql-server -ver16](https://learn.microsoft.com/es-es/sql/relational-databases/security/securing-sql-server?view=sql-server-ver16)

# **Entrega 3**

**Versiones de DER:**

## Versión 1:



## Versión 2:



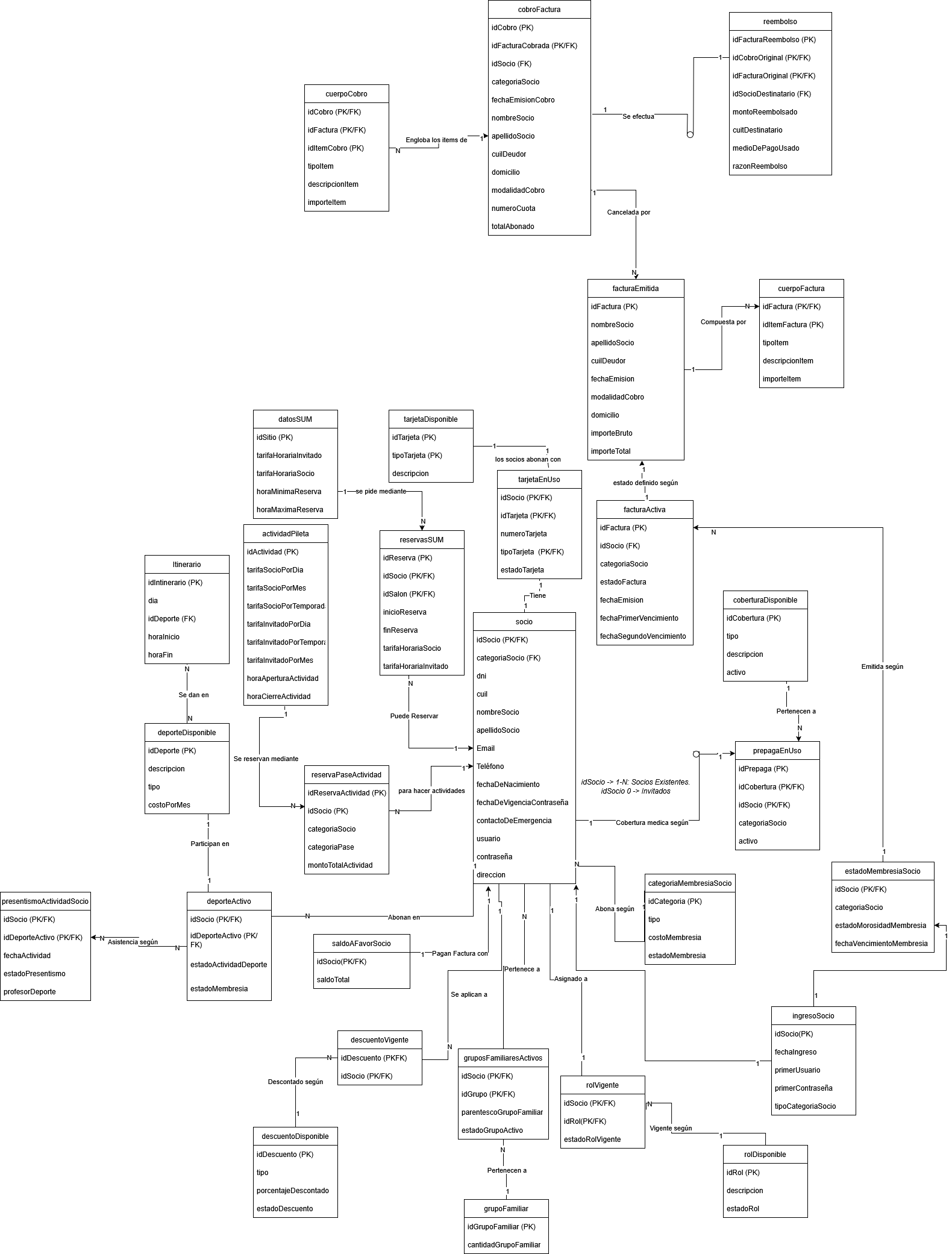
**Historial de Cambios:**

* tabla "tutorACargo" reemplazada con "grupoFamiliar". También se usa una tabla intermedia "gruposFamiliaresActivos" para poder relacionar socios con sus respectivos grupos y su parentesco.
* tabla "socios" tiene "estadoMembresia" y "fechaDeVencimientoMembresia". Se considera que habrá un Stored Procedure para actualizar todos los días aquellos socios con vencimientos.
* medioDePago y medioEnUso reemplazadas por tarjetaDisponible y tarjetasEnUso, respectivamente.
* facturaCobro estará relacionado con los socios, no con sus tarjetas, para así puede ser más fácil hacer la distinción entre Tarjetas, PagoFacil, etc.
* cambiada la cardinalidad de la relación de deporteActivo (de 1-1 a N-1)
* removido por completo la constraint relacionada a la tarjeta en facturaCobro, reemplazado por el campo "medioDePagoUsado".
* idem para reembolso.

## Versión 3:



**Historial de cambios 2:**

* Agregada la entidad "cobroFactura" para poder rastrear los cobros específicos para cada factura y para cada servicio. Con este esquema, también podremos rastrear las cuotas pagadas para cada factura (descrito en el campo "modalidadCobro")
* Agregados los campos "categoriaSocio" junto con "idServicioFacturado" y "descripcionServicioFacturado", junto con datos personales del Socio.
* Agregados también los campos "fechaPrimer/SegundoVencimiento", junto con el domicilio desde donde se emite la factura y su estado (Pendiente / Pagada / Nulificada).
* La entidad reembolso estará relacionada con cobroFactura, pues tiene sentido que los reembolsos se den en relación a los cobros enviados al club.
* Removida la entidad "prepagaDisponible", dejando solo "prepagaEnUso" para poder rastrear los datos del individuo.
* Agregados los campos "estadoActividadDeporte" y "estadoMembresia" para poder rastrear si el deporte se encuentra activo y el socio sigue en la institución, respectivamente.
* Idem para el campo "estadoDescuento", "estadoRolVigente", "estadoGrupoActivo", "estadoRol", "estadoCategoriaSocio", "activo" (para prepagaEnUso) y "estadoFactura". Todos estos atributos ayudaran para mantener un seguimiento de los servicios prestados o personas involucradas, además de permitirnos el borrado lógico de sus tuplas en caso de necesitarlo.
* Reemplazada la entidad "actividadRecreativa" con las tablas "datosSUM", "reservasSUM" para el Salón de usos multimedios y "actividadPileta" y "reservaPaseActividad" para tanto la Colonia como la Pileta Recreativa.
* Tomamos la consideración de reservar la idSocio = 0 para los invitados, así podemos emitir facturas y cobros en base a ese número de identificación.

Historial de Cambios:

- Separada la entidad facturaEmitida en dos campos: la entidad original (que es la que contendrá los datos como el nombre/apellido del deudor junto con su fecha de emisión) y cuerpoFactura (que contendrá todos y cada uno de los ítems correspondientes a la misma) junto con el tipo de los mismos y su importe individual (para así poder identificarlos con mayor facilidad).

- Agregada la entidad facturaActiva, en donde guardaremos el estado de la factura (Pendiente, Pagada o Nulificada) junto con sus fechas de vencimiento (así también podremos identificar que importe deberá abonar el socio a la hora de hacer el cobro).

- Agregada la entidad “presentismoActividadSocio” para poder rastrear los socios, las actividades a las que asisten, en QUE fecha asisten y bajo que profesor estuvieron a cargo.

- Separamos el campo “saldoAFavorSocio” acorde a las condiciones marcadas, pues al ser un dato que será visto y modificado constantemente, conviene tenerlo en una entidad aparte para así poder seguirlo mejor y no generar más complicaciones. Además, el atributo en sí se llamará “saldoTotal”, para evitar mayores confusiones a la hora de trabajarlo.

- Agregada la entidad “estadoMembresiaSocio” por las mismas razones que “saldoAFavorSocio”. Es algo que se estará modificando constantemente y hace falta tenerlo a mano siempre.

-Agregada la entidad "cuerpoCobro" para poder también rastrear los datos del cobro (básicamente, para que sea simétrico con la factura)

# **Entrega 4:**

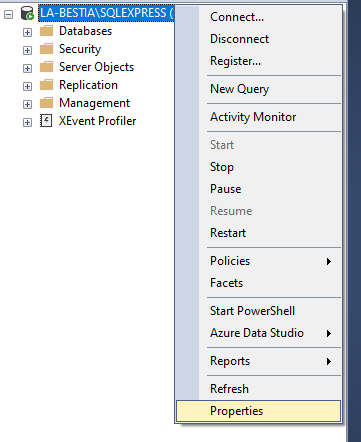
## Cómo instalar SQL Server:

1. Decidir en qué máquina se hará la instalación (nosotros ya tenemos un equipo dedicado a esa labor).
2. Descargar SQL Server 2022 desde la [página oficial de Microsoft](https://www.microsoft.com/es-es/download/details.aspx?id=104781).
3. Una vez descargado, ejecutar el archivo que descargamos (debería llamarse SQL2022-SSEI-Expr.exe o algo similar.)
4. Una vez abierto el ejecutable, elegiremos la instalación “básica” para nuestro dispositivo.
   1. Al optar por la instalación básica, usaremos las configuraciones predeterminadas (pues no necesitamos modificar nada de ellas para llevar a cabo nuestro proyecto).
   2. Aceptando el acuerdo, dejamos la ruta de instalación tal y como está y le damos clic al botón “Instalar”.
5. Al terminar con la instalación, le daremos clic al botón de “Instalar SMSS”. El SQL Server Management Studio es la herramienta con la que vamos a trabajar para llevar a cabo nuestro proyecto, por lo que es imperativo que terminemos de instalarlo una vez terminemos de instalar el SQL Server.
6. Elegiremos la ubicación predeterminada para el SQL Management Studio y le daremos clic a “Instalar”. Cuando termine, nos aparecerá una pantalla que diga “Instalación Completada”.

**CONSIDERACIÓN ADICIONAL:**

Para poder hacer importaciones masivas desde archivos .CSV a futuro, es importante que dichas acciones/importaciones se hagan desde el rol de SA (sysadmin). Para hacer esto, hace falta:

* **Hacer clic derecho sobre nuestro servidor desde el centro de conexiones y seleccionar la opción propiedades.**



* En la pestaña de “Seguridad” / “Security”, habilitar la autentificación de SQL Y Microsoft para acceder a la BDD.
* Ejecutar desde el archivo “01\_importDatosCSV.sql” el comando “ALTER LOGIN sa ENABLE” y “ALTER LOGIN sa WITH PASSWORD = 'M1x3dMode!2025';”. Esto habilitará nuestra cuenta de administrador y nos dará permiso para iniciar sesión como la misma, siempre usando esa contraseña antes declarada.
* Buscar en services.msc (el administrador de servicios de Windows) la cuenta asociada al Servicio SQLEXPRESS (en nuestro caso fue NT Service\MSSQL$SQLEXPRESS) y anotarlo.
* Seleccionar la tabla donde estarán guardado los CSV (es recomendado guardar todo en una carpeta local), darle clic derecho y acceder a propiedades.
  + Una vez en el menú de propiedades, darle clic a la solapa de Seguridad, editar, agregar e ingrese el nombre que anotó en el paso anterior.
* Darle clic a aceptar.
* Asignarle todos los permisos de lectura/escritura a la cuenta de SQL EXPRESS.
* Iniciar el SQL Server de nuevo.
* En la pestaña de “Authentication”, seleccionar “SQL Server Authentication”.
* Ingresar los datos del login para sa.
* En la pestaña de “Encryption”, seleccionar “Mandatory” y habilitar la opción directamente abajo (Always Trust).

**¡Con esto, ya está!** Se inicio sesión como sysadmin y se tienen habilitados los permisos para ingresar datos por bulkInsert (la parte de los certificados para los permisos ya fueron gestionados por nosotros).

**Configuración:**

1. Hasta que tengamos la encriptación y seguridad preparada, entraremos al entorno de desarrollo de la siguiente manera:
   1. En la sección de “Autentificación / Authentication” (depende de si lo tienen en inglés o en español), seleccionaremos “Windows Authentification”.
   2. En la sección de “Encryption”, le daremos clic a “Optional”, y le daremos clic a conectar.

Una vez cambiemos el entorno de desarrollo, habrá que aportar credenciales para acceder a la base de datos propiamente dicha.

## Glosario

* datosSUM:
  + idSitio: por el momento será 0. En caso de construir o tener la capacidad de usar más salones, se irán creando más tuplas.
  + tarifaHorarioInvitado/Socio
  + horaMinimaReserva: Desde CUANDO se toman las reservas.
  + horaMaximaReserva: Hasta CUANDO se toman las reservas.
* reservasSUM:
  + idReserva.
  + idSocio.
  + idSalon
  + inicioReserva: a qué hora arranca la reserva.
  + finReserva: hasta qué hora es la reserva.
  + tarifaFinal: idSocio>=1 : Tarifa de Socio. idSocio=0 : Tarifa de Invitado.
* actividadPileta:
  + idActividad: Por ahora solo contaremos con las actividades de Colonia y Pileta Recreativa.
  + tarifaSocio/InvitadoPorDia: Es el único dato que guardaremos, pues los demás son calculables.
  + horaApertura/horaFin: inicio / Fin de la actividad.
* reservaPaseActividad:
  + idReserva
  + idSocio
  + categoriaSocio
  + categoriaPase: Dia, Mes, Temporada.
  + montoTotal (Acá dejamos el calculable en base a tarifaPorDia).
* Itinerario:
  + idItinerario: Lo planteamos de tal manera que se puedan cambiar itinerarios según temporada. (En Verano, por ejemplo, se usaría la Pileta para Natación, Colonia, etc…en invierno, sin embargo, se usaría climatizada únicamente para natación).
  + dia
  + idDeporte
  + horaInicio
  + horaFin
* deporteDisponible:
  + idDeporte
  + Descripción.
  + Tipo
  + costoPorMes.
* deporteActivo: Entidad que relaciona y describe QUÉ socios hacen QUÉ deportes.
  + idSocio
  + idDeporte
  + estadoActividadDeporte (Activo - Inactivo)
  + estadoMembresia
* descuentoDisponible:
  + idDescuento
  + tipo
  + porcentajeDescontado
  + estadoDescuento (Activo - Inactivo)
* descuentoVigente:
  + idDescuento
  + IdSocio
* grupoFamiliar:
  + idGrupoFamiliar
  + cantidadGrupoFamiliar
* gruposFamiliaresActivos:
  + idSocio
  + idGrupo
  + parentescoGrupoFamiliar (Tutor - Menor)
  + estadoGrupoActivo (también usado para borrado lógico)
* categoriaSocio:
  + idCategoria
  + tipo (Cadete – Menor - Mayor)
  + costoMembresia
  + estadoCategoria
* rolDisponible:
  + idRol
  + descripcion
  + estadoRol
* rolVigente:
  + idSocio
  + idRol
  + estadoRolVigente
* tarjetaDisponible
  + idTarjeta
  + tipoTarjeta ('Credito', 'Debito', 'Prepaga', 'Virtual'))
  + descripción
* tarjetaEnUso:
  + idSocio
  + idTarjeta
  + tipoTarjeta
  + numeroTarjeta
  + estadoTarjeta
* prepagaEnUso:
  + idPrepaga
  + idCobertura (PK/FK): que Cobertura tiene el usuario
  + idSocio
  + categoriaSocio
  + activo (Para Borrado Lógico)
* coberturaDisponible:
  + idCobertura
  + tipo
  + descripción
  + activo (para borrado lógico)
* socio:
  + idSocio
  + categoriaSocio
  + dni
  + cuil (tanto el DNI como el CUIL fueron validados con funciones aparte).
  + nombreSocio
  + apellidoSocio
  + email
  + telefono
  + fechaDeNacimiento
  + fechaVencimientoContraseña (se tiene que renovar cada tres meses)
  + fechaVencimientoMembresia (se tiene que renovar cada mes)
  + fechaIngresoSocio
  + contactoDeEmergencia
  + usuario
  + contraseña
  + saldoAFavor
  + dirección
* facturaEmitida:
  + idFactura
  + idSocio
  + categoriaSocio
  + idServicioFacturado
  + descripcionServicioFacturado
  + nombreSocio
  + apellidoSocio
  + fechaEmision (desde está fecha se calcula el primer y segundo vencimiento)
  + cuilDeudor
  + domicilio
  + modalidadCobro (Cuotas: 1, por ejemplo. En la función de cobro, descompone la el varchar y saca el número para poder descubrir cuantas cuotas debe sacar.
  + importeBruto (Antes de Descuentos)
  + importeNeto (Luego del Descuento)
  + fechaPrimerVencimiento
  + fechaSegundoVencimiento
  + estadoFactura (Pendiente – Pagada - Nulificada)
* cobroFactura:
  + idCobro
  + idFacturaCobrada
  + idSocio
  + categoriaSocio
  + idServicioCobrado
  + descripcionServicioCobrado
  + fechaEmisionCobro
  + nombreSocio
  + apellidoSocio
  + cuilDeudor
  + domicilio
  + modalidadcobro
  + numeroCuota
  + totalAbonado
* reembolso:
  + idFacturaReembolso
  + idCobroOriginal (el REEMBOLSO debe generarse por un COBRO)
  + idFacturaOriginal (idem per con COBRO y FACTURA)
  + idSocioDestinatario
  + montoReembolso
  + cuitDestinatario
  + medioDePagoUsado
  + razonReembolso

Scripts para la Base de Datos encontrados en el ZIP entregable.

TODAS las entidades traen consigo tres SPs: insertarEntidad, actualizarEntidad y eliminar/desactivar/nulificarEntidad. Tomamos en cuenta también la necesidad del borrado lógico y lo aplicamos en donde lo creímos conveniente.

## Funciones:

validarCUIL: Valida un CUIL traído en formato de texto matemáticamente

validarDNI: Valida la longitud de un DNI para asegurarse de que estemos tratando con un dato correcto.

# **Entrega 7**

Asigne los roles correspondientes para poder cumplir con este requisito, según el área a la cual pertenece.



Por otra parte, se requiere que los datos de los empleados se encuentren encriptados, dado que los mismos contienen información personal.

La información de las cuotas pagadas y adeudadas es de vital importancia para el negocio, por ello se requiere que se establezcan políticas de respaldo tanto en las ventas diarias generadas como en los reportes generados.

Plantee una política de respaldo adecuada para cumplir con este requisito y justifique la misma. No es necesario que incluya el código de creación de los respaldos.

Debe documentar la programación (Schedule) de los backups por día/semana/mes (de acuerdo a lo que decidan) e indicar el RPO.

--

Todo lo que es asignación de Roles de Seguridad y Permisos están especificados y realizados en el SQL en el archivo “03\_**RequisitosDeSeguridad.sql**”

## Políticas de Respaldo y Encriptación de Datos

La información de las cuotas pagadas y adeudadas, así como los datos personales de socios y empleados, son críticos para el negocio. Por lo tanto, se establecerían políticas robustas de encriptación y respaldo.

1. **Encriptación de Datos Sensibles**

Requisito: Proteger la información personal contenida en la base de datos.

Política Propuesta: Se implementará una estrategia de encriptación de dos niveles para la base de datos Com2900G03:

1. **Transparent Data Encryption (TDE)**: Se aplicará TDE a toda la base de datos Com2900G03. Justificación: TDE encripta los archivos de datos (.mdf) y de registro (.ldf) en el disco. Esto protege los datos en reposo contra accesos no autorizados a nivel de sistema de archivos (ej. si alguien roba los discos o accede a las copias de seguridad). Es transparente para las aplicaciones, no requiriendo cambios en el código.
2. **Always Encrypted**: Se utilizará Always Encrypted para columnas específicas que contienen datos altamente sensibles dentro de las tablas la BD. Columnas a Encriptar:
   1. socios.socio: dni, cuil, email, telefono, fechaNacimiento, contrasenia.
   2. socios.grupoFamiliar: dni, emailPersonal, fechaNacimiento, telefonoContacto, telefonoContactoEmergencia, nombreObraSocial, nroSocioObraSocial, telefonoObraSocialEmergencia.
   3. pagos.tarjetaEnUso: numeroTarjeta.
   4. pagos.reembolso: cuilDestinatario.

Justificación: Always Encrypted garantiza que los datos sensibles permanezcan encriptados en la base de datos, incluso en memoria, y solo se descifren en la aplicación cliente. Esto protege contra amenazas internas (ej. administradores de base de datos con privilegios) y reduce el riesgo de exposición de datos durante el procesamiento de consultas.

1. **Políticas de Respaldo (Backup) y Recuperación (RPO)**

Requisito: Asegurar la disponibilidad y la integridad de los datos de cuotas, ventas diarias y reportes generados.

Política de Respaldo:

1. **Copias de Seguridad Completas (Full Backups):**

* **Frecuencia**: Semanalmente (por ej. cada Sábado a las 01:00 AM).
* **Justificación**: Proporciona una base completa para la recuperación de la base de datos.
* **Ubicación**: Almacenamiento de red seguro con redundancia, o un servicio de almacenamiento en la nube para durabilidad y accesibilidad en caso de desastre local.

1. **Copias de Seguridad Diferenciales (Differential Backups):**

* **Frecuencia**: Diariamente (por ej. de Domingo a Viernes a las 01:00 AM).
* **Justificación**: Capturan eficientemente los cambios realizados en la BD desde el último backup completo, acelerando el proceso de respaldo y reduciendo el tiempo de restauración en comparación con la aplicación de múltiples backups de registro.
* **Ubicación**: Misma ubicación que los backups completos.

1. **Copias de Seguridad del Registro de Transacciones (Transaction Log Backups):**

* **Frecuencia**: Cada 15 minutos.
* **Justificación**: Son fundamentales para la recuperación a un punto en el tiempo (Point-in-Time Recovery) y para minimizar la pérdida de datos. Capturan todas las transacciones realizadas en la BD desde el último backup de registro, asegurando que la información de cuotas y ventas (en tablas como pagos.facturaActiva, pagos.facturaEmitida, pagos.cobroFactura, socios.saldoAFavorSocio) esté protegida con la mínima pérdida.
* **Ubicación**: Misma ubicación que los backups completos.

**Programación Detallada (Schedule):**

* Lunes a Viernes:
  + 01:00 AM: Backup Diferencial de Com2900G03.
  + Cada 15 minutos (por ej. 01:15, 01:30, ..., 00:45 del día siguiente): Backup del Registro de Transacciones de Com2900G03.
* Sábado:
  + 01:00 AM: Backup Completo de Com2900G03.
  + Cada 15 minutos (a partir de las 01:15): Backup del Registro de Transacciones de Com2900G03.
* Domingo:
  + 01:00 AM: Backup Diferencial de Com2900G03.
  + Cada 15 minutos (a partir de las 01:15): Backup del Registro de Transacciones de Com2900G03.

**Retención de Backups:**

* Registros de Transacciones: Retener los backups de los últimos 7 días.
* Diferenciales: Retener los últimos 7 backups diferenciales
* Completos: Retener los últimos 4 backups completos (equivalente a 1 mes).
* Archivos de Largo Plazo: Considerar un backup completo mensual o trimestral archivado por 1 año o más, según requisitos de auditoría o normativos.

**RPO (Recovery Point Objective - Objetivo de Punto de Recuperación):**

* RPO Definido: 15 minutos.
* Justificación: Dada la criticidad de los datos de cuotas y ventas diarias, un RPO de 15 minutos asegura que, en el peor de los casos de un fallo inesperado de la base de datos, la pérdida máxima de datos transaccionales no superará los 15 minutos. Esto se logra mediante la aplicación del último backup completo, el último diferencial y todos los backups de registro de transacciones subsiguientes hasta el punto de fallo.