

ESCUELA SUPERIOR DE ECONOMÍA Y NEGOCIOS (ESEN)

**Entregable 1: Especificación de Software para la aplicación de SIGER (Sistema de
Generación y Calidad de Energía, S.A. de C.V)**

Elaborado por:



Integrantes:

Kathleen Abigail Argueta Gómez

Alejandro Javier Durón Rodríguez

Kevin Elías Luna Palacios

Lorena Esmeralda Mejía Ramos

Sheyla Alexandra Sarmiento Aragón

Catedrático:

Licenciado Alvin Portillo

Santa Tecla, 20 de junio del 2025

Entregable 1: Especificación de Software para la aplicación de SIGER (Sistema de Generación y Calidad de Energía, S.A. de C.V)

1. Introducción

El siguiente documento trata sobre el análisis y revisión preliminar del proyecto asignado por parte de la empresa SIGER. La compañía antes de nuestra intervención podía prestarse a problemáticas por la forma en cómo se guardaban, procesaban y manejaban los datos de esta, pues hacía uso de un Excel para esta tarea. La situación daba lugar a posibles filtraciones, perdidas y uso ineficiente de los datos.

El objetivo de la siguiente entrega del proyecto es asegurar la comprensión de la problemática, identificación clara y precisa de los dolores de la empresa, esquematización de la solución y explicación de esta. Además de verificar que se logre solventar todas las necesidades antes, durante y después de la iniciación del proyecto.

2. Revisión del Problema y Solución Propuesta (Aplicación PyQt6)

2.1. Descripción del problema original

La empresa SIGER enfrenta problemas con la optimización y el almacenamiento de información en el área de registro general de clientes y facturación. Actualmente, para el registro de datos utiliza archivos de Excel, lo que dificulta la búsqueda y gestión de datos. Además, daba lugar a posibles filtraciones, perdidas y uso ineficiente de los mismos. Entre los principales problemas generados por este sistema están:

- **Dificultad en búsquedas y pérdidas de tiempo:** Se invierten mucho tiempo en la organización de los datos, ya que no hay filtros eficientes para localizar registros específicos.
- **Errores manuales e ineficiencia en reportes:** La introducción manual de datos aumenta el riesgo de duplicados o información incorrecta. Además, generar informes se dificulta debido a la repetición de este tipo de procesos.

- **Datos desactualizados:** La información no se actualiza en tiempo real, lo que genera inconsistencias.
- **Falta de seguridad:** Los archivos pueden ser modificados o eliminados sin control, poniendo en riesgo la integridad de los datos.
- **Problemas de colaboración:** Dificultad para trabajar en equipo, ya que los archivos no permiten acceso simultáneo.

2.2. Propuesta de solución usando una aplicación con PyQt6

La forma en cómo se abordará lo anteriormente expuesto es mediante el uso de bases de datos relacionales, manejadas por scripts de SQL y la plataforma DBeaver. Además, conllevará una representación gráfica de la aplicación, donde el uso de PyQt6, una biblioteca de Python para crear programas con interfaz gráfica será vital para dar la solución adecuada. La aplicación solucionará los problemas de la siguiente manera:

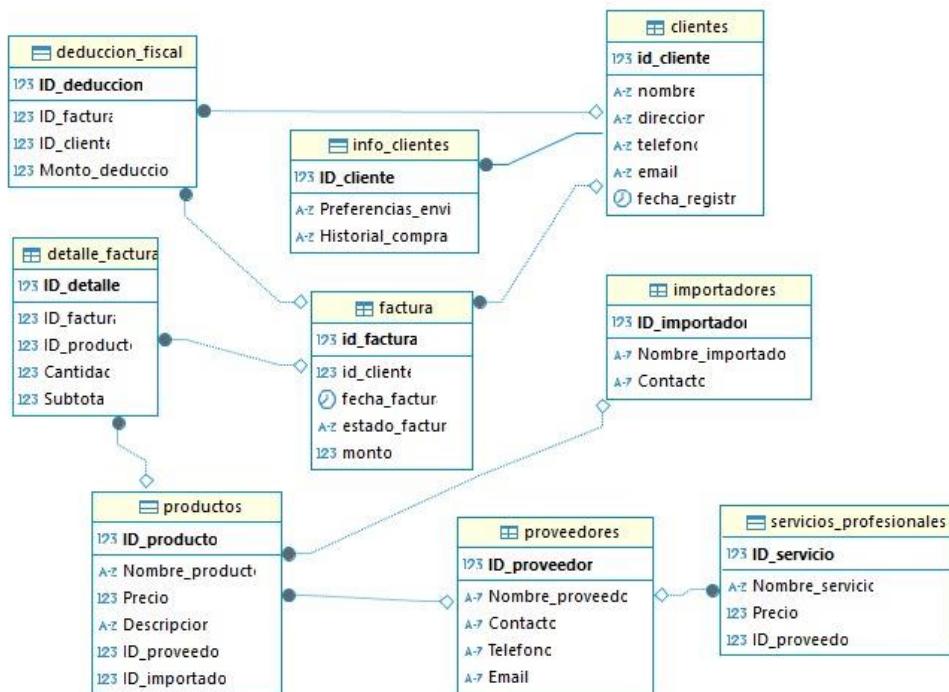
- **División en tablas:** La información será dividida dentro de tablas según el tipo de registro. Estas tablas serán Cliente, Info_Cliente, Deduccion_Fiscal, Factura, Detalle_Factura, Producto, Importadores, Proveedores.
- **Agrupación de datos:** Permitirá agrupar los datos según el tipo de información. Por ejemplo, dentro de Cliente, los campos serán nombre, dirección, teléfono, email y fecha_registro. En las otras tablas, los tipos de datos que se podrán ingresar serán diferentes.
- **Interfaz intuitiva (PyQt6):** Permitirá a la empresa llenar los datos necesarios de forma eficiente y amigable.
- **Búsqueda avanzada:** Permitirá filtrar por campos (nombre, fecha, ID), y mostrará los resultados en tiempo real.
- **Generación de reportes:** Permitirá exportar las facturas, los listados de clientes o los productos en un archivo de texto (*.txt).

- **Validación automática:** Tendrá campos obligatorios y formatos predefinidos (por ejemplo, fechas).

3. Evaluación del Diseño de la Base de Datos (del Ciclo Anterior)

3.1. Esquema de la Base de Datos Propuesto Originalmente

Durante el ciclo anterior, se diseñó una base de datos para SIGER, con el objetivo de mejorar el control sobre los clientes, las ventas de la empresa, y los proveedores. A continuación, se presentan las principales tablas diseñadas:



Estas tablas buscan representar las operaciones comerciales de SIGER, incluyendo los datos de los clientes, los productos y los proveedores.

3.2. Adecuación y Completitud para la Aplicación PyQt6

El diseño de base de datos resulta adecuado para los objetivos actuales de la aplicación que se está desarrollando con PyQt6. Esta estará enfocada en facilitar el registro de clientes, las compras, y la gestión de los proveedores.

La estructura en tablas separadas permite que sea más efectivo al momento de observar y analizar sus secciones dentro de la aplicación. Asimismo, los datos estarán relacionados para facilitar búsquedas y la toma de decisiones.

El diseño de base de datos resulta adecuado para los objetivos actuales de la aplicación que se está desarrollando con PyQt6. Esta estará enfocada en facilitar el registro de clientes, las compras que estos realizan, y la gestión con los proveedores.

3.3. Ajustes y Mejoras Necesarias en el Diseño de la Base de Datos

Considerando los requisitos de la aplicación PyQt6, se propone agregar una tabla a la base de datos creada durante el ciclo pasado.

Nueva Tabla: Usuarios

- ID_Usuario (PK): INT
- NombreUsuario: VARCHAR(50) unique
- contrasenna: VARCHAR(255)
- Rol: VARCHAR(20)

Justificación: Sería útil para tener idea de cuándo fue la última vez que se cambió algún dato, sobre todo en productos, proveedores y servicios profesionales. Además, permite mayor seguridad en la información.

4. Especificaciones de software

4.1. Requisitos Funcionales

La aplicación deberá cumplir con los siguientes requisitos:

1. Ingreso y gestión de datos
 - Crear, leer y actualizar (CRUD) registros de compras, información de clientes y proveedores, así como ingresos e información de servicios prestados y sus respectivas fechas.
 - La aplicación deberá permitir ingresar datos de los clientes, el servicio o producto adquirido y la fecha del movimiento.

- Se deberá incluir la función de registrar y administrar los productos y servicios disponibles que ofrece la empresa.
- Debe ser posible ingresar la información de los clientes y proveedores, asociando esta con los productos, facturas y deducciones fiscales.

2. Información de clientes

- La información de los clientes deberá incluir información de contacto
- Cada tipo de registro debe contar con un formulario adecuado para ingresar los datos relevantes (nombre, dirección, teléfono, etc.).

3. Visualización de datos

- La aplicación debe permitir consultar y visualizar la información ingresada en las tablas.
- Debe permitir realizar búsquedas y filtrados según parámetros específicos (por ejemplo, mostrar todos los proveedores en una región determinada, o todas las facturas de un cliente).
- La interfaz gráfica debe ser intuitiva, con tablas y formularios visibles y fáciles de manejar.

4. Agrupación de datos:

- Los datos deben estar organizados en tablas relacionadas entre sí, siguiendo el modelo relacional (por ejemplo, los clientes deben estar asociados con sus facturas, los proveedores con los productos, etc.).
- La aplicación debe permitir agrupar y relacionar datos de manera eficiente y clara en la interfaz gráfica.

5. Modificación y eliminación de datos:

- Los usuarios deben poder modificar y eliminar registros en las tablas, con confirmación antes de realizar estas acciones para evitar accidentes.

6. Generación de reportes:

- La aplicación debe permitir generar reportes en formato de lista, tabla o texto para mostrar las facturas, productos, deducciones fiscales, entre otros.
- Los reportes deben ser exportables a formatos como CSV, TXT, PDF o Excel.

7. Validación de datos:

- Deben realizarse validaciones de datos al momento de ingresar información para asegurar que los campos obligatorios estén completos y que los datos sean del tipo adecuado (por ejemplo, asegurarse de que los correos electrónicos tengan un formato válido, que los números de teléfono contengan solo números, etc.).

8. Acceso y gestión de roles:

- La aplicación debe permitir gestionar diferentes tipos de usuarios (por ejemplo, administradores, usuarios normales) y asignar permisos específicos a cada uno (por ejemplo, solo los administradores pueden eliminar registros).

9. Integración con SQL:

- La aplicación debe estar integrada con una base de datos SQL, donde las tablas y relaciones estén bien definidas y que estas sean relevantes para la obtención y administración de datos de la empresa.
- Los scripts SQL deben ser capaces de manejar el ingreso, actualización, consulta y eliminación de datos correctamente.

4.2. Requisitos no funcionales

1. Rendimiento:

- La aplicación debe ser capaz de manejar grandes cantidades de datos sin afectar su rendimiento (por ejemplo, consultas rápidas de datos masivos sin tiempos de espera excesivos).
- El tiempo de respuesta en la interfaz gráfica debe ser rápido.

2. Escalabilidad:

- La aplicación debe ser escalable para permitir la expansión en términos de volumen de datos y número de usuarios sin que afecte el rendimiento.
- Debería ser fácil agregar nuevas tablas y relaciones a la base de datos a medida que el sistema crezca.

3. Seguridad:

- Los datos sensibles (como información de contacto de los clientes y proveedores) deben estar protegidos mediante encriptación y acceso restringido.
- Se deben implementar medidas de seguridad para evitar inyecciones de SQL y otros tipos de ataques comunes.

4. Usabilidad:

- La interfaz gráfica debe ser fácil de usar e intuitiva, con un diseño atractivo y accesible.
- Debe haber ayuda contextual disponible dentro de la aplicación (por ejemplo, explicaciones emergentes sobre qué datos ingresar en cada campo).

5. Mantenibilidad:

- El código de la aplicación debe ser modular y fácil de mantener, con documentación adecuada para los desarrolladores.
- Las actualizaciones y correcciones deben ser fáciles de implementar.

6. Interoperabilidad:

- La aplicación debe ser capaz de integrarse con otros sistemas, como plataformas de contabilidad o gestión de inventarios, si fuera necesario.

4.3. Tecnologías a Utilizar

- Lenguaje de Programación: Python 3.x
- Framework de GUI: PyQt6
- Base de Datos: SQLite (para facilidad de despliegue en esta fase)

5. Conclusiones y Próximos Pasos

El proyecto de software para la empresa SIGER representará una solución bien estructurada abordando todas las problemáticas anteriormente expuestas, principalmente las que están ligadas al uso de un documento de Excel para llevar el registro de las operaciones de la compañía. Nuestra propuesta consiste en el uso de bases de datos relaciones, con uso de consultas SQL y una interfaz gráfica, con ayuda de PyQt6, ofreciendo mayor seguridad, mejor control y organización de datos y además una disminución de errores manuales.