



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Integración SOAP para Carga Masiva de
Contratos en un ERP Corporativo**

INFORME DE ACTIVIDADES PROFESIONALES

Que para obtener el título de
Ingeniero en Computación

P R E S E N T A

José Manuel García Gómez

ASESOR DE INFORME

Ing. Jorge Luis López García



Ciudad Universitaria, Cd. Mx., 2026

Índice

1. Introducción.....	1
2. Objetivo.....	1
3. Descripción de la empresa.....	2
3.1. La empresa.....	2
3.2. Descripción del puesto de trabajo.....	2
4. Marco Teórico.....	3
4.1. Servicios Web.....	3
4.2. SOAP.....	6
4.3. WSDL	8
5. Antecedentes del Proyecto.....	9
5.1. Funcionamiento del ERP (<i>Enterprise Resource Planning</i>)	9
5.1.1. ¿Qué es un ERP?.....	9
5.1.2. ERP de la Empresa.....	9
5.2. Aplicación de un Contrato.....	10
5.3. Gestión de Contratos.....	10
6. Definición del problema o contexto de la participación profesional.....	11
7. Metodología utilizada.....	12
7.1. Etapas del modelo aplicado	13
7.1.1. Análisis.....	13
7.1.2. Diseño	14
7.1.3. Desarrollo	15
7.1.4. Pruebas	19
7.1.5. Despliegue	20
7.1.6. Mantenimiento	21
8. Participación profesional.....	21
9. Resultados y aportaciones.....	22
9.1. Resultados obtenidos	22
9.2. Resultados de las pruebas de rendimiento	24
9.3. Análisis detallado del servicio SOAP	25

9.4. Resultados del procesamiento integral de la aplicación	25
9.5. Incidentes e inconsistencias	26
10. Conclusión.....	28
11. Bibliografía.....	31

1. Introducción

Este escrito tiene la finalidad de informar el sistema que desarrollé e implementé para la empresa que laboro actualmente, dicha empresa brinda productos de enseñanza en idiomas, donde el objetivo principal es mejorar la productividad en los usuarios, ya que, actualmente su ERP (*Enterprise Resource Planning*) cuenta con diversas limitantes, lo que provoca deficiencia operativa y pérdida de tiempo para los clientes que llegan a contratar algún producto de enseñanza. Por ende, es necesaria una nueva herramienta que pueda remediar dichas limitantes.

Este proyecto fue desarrollado con base en los conocimientos adquiridos durante mi formación académica en la carrera de Ingeniería en Computación, integrando metodología de análisis de sistemas e ingeniería de software, para el diseño de la nueva herramienta, así como experiencia adquirida en el empleo, al percatarme de la problemática que presentaban los usuarios del ERP a la hora de gestionar un proceso en la plataforma.

Finalmente, el propósito principal de mi trabajo es evidenciar las competencias profesionales y técnicas adquiridas durante la formación académica, así como demostrar la capacidad para identificar problemas reales dentro de un entorno empresarial y proponer soluciones tecnológicas que aporten valor y mejora a la experiencia del usuario final.

2. Objetivo

Desarrollar e implementar una herramienta de software integrada con el ERP corporativo que permita automatizar la finalización masiva de contratos mediante el consumo de servicios SOAP y el acceso a la información requerida de las bases de datos, con el propósito de ejecutar la operación de manera controlada, considerando las condiciones asociadas a cada contrato y mejorando la eficiencia del procesamiento.

3. Descripción de la empresa

3.1. La Empresa

El corporativo es líder en el área de enseñanza de idiomas. Ésta lleva más de 145 años impartiendo clases de habla a nivel mundial. A lo largo de su historia ha creado diversos métodos de enseñanza, desde traducción gramatical, hasta métodos dinámicos e interactivos. Dichos métodos reciben su nombre gracias a un profesor políglota de origen alemán que impartió clases en Estados Unidos de América, dando origen a un método de trabajo práctico y funcional.

Este método consiste en tres puntos clave:

1. **Inmersivo:** Donde el profesor se dirige al alumno, en modo verbal y escrito, únicamente en el idioma que se está aprendiendo.
2. **Orientado hacia lo objetivo:** Aplicado a aprender el idioma a través de simulación de escenarios de la vida real.
3. **Presentación, práctica y performance:** En cada sesión se presentan nuevas estructuras, luego, el profesor acompaña al alumno en la práctica de dicho contenido y, finalmente, el alumno lo practica de manera individual.

Hoy en día la escuela cuenta con más de 50 centros de enseñanza de idiomas en Latinoamérica, cada uno de ellos necesita Soporte de Aplicativos de Negocios (ERP) con tareas específicas para un funcionamiento correcto.

3.2. Descripción del puesto de trabajo

Las actividades desarrolladas en el puesto de trabajo son:

1. Soporte y análisis a sus Aplicativos de Negocios, los cuales, son encargados de vender y facturar productos de enseñanza, tanto para usuarios individuales como grupales o empresariales, e incluso, servicios de traducción.
2. Detectar inconsistencias en la base de datos (SQL Server 2014 SP2 CU11) o posibles bugs de sus aplicaciones web.
3. Mejorar tareas de demanda masiva para su sistema.

4. Otorgar permisos a sus sistemas de negocio.
5. Configurar la infraestructura de red para nuevos centros que lleguen a crearse a lo largo del tiempo.
6. Configuración de ODBC's para conectar un ERP al sistema de base de datos (Sybase Central 6.0.0.4739)

4. Marco Teórico

4.1. Servicios Web

El internet y su manera de intercomunicarse entre diversos dispositivos han ido actualizándose a lo largo de la historia para un mejor manejo de los datos tanto de transmisión como de recepción. Una manera común de comunicarse es mediante Servicios Web. Pero ¿qué es un servicio Web? Los servicios Web son aplicaciones autónomas modulares que se pueden describir, publicar, localizar e invocar a través de una red. (IBM, 2023).

Los WS (en inglés, *Web Services*)¹ utilizan diversas tecnologías basadas en protocolos y estándares que permiten establecer comunicación entre diferentes aplicaciones. Entre estas tecnologías se encuentra SOAP (en inglés, *Simple Object Access Protocol*)², un protocolo basado en XML utilizado para el intercambio de información estructurada. SOAP y WSDL fueron utilizados ampliamente como tecnologías para la integración de aplicaciones, particularmente en entornos empresariales (W3C, 2007).

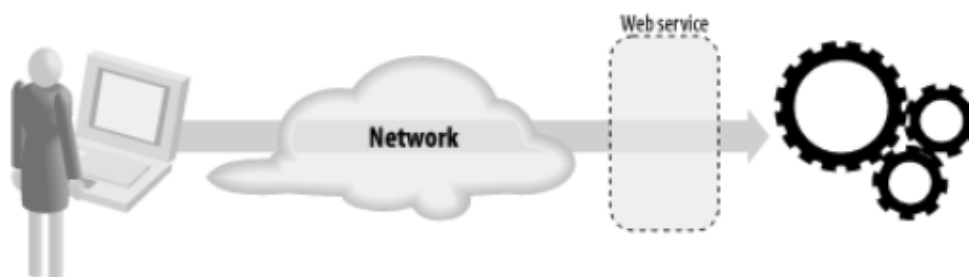
Cuando una aplicación puede acceder al código fuente de otra aplicación, a través de una red, empleando una combinación de protocolos estandarizados como HTTP, XML, SOAP, o algún otro, se habla, entonces, de un WS.

En la imagen 1 se describe la interacción del usuario y el código fuente:

¹ A partir de aquí me referiré a *Web Services* con las siglas WS.

² En adelante las siglas SOAP harán referencia a *Simple Object Access Protocol*.

Imagen 1. Consumo de un WS.



Nota. Tomado de Programming Web Services with SOAP, por J. Snell, D. Tidwell y P. Kulchenko, 2001, O'Reilly Media, p. 6.

Los WS proporcionan una capa de abstracción entre el cliente de la aplicación y el código de la aplicación como se muestra en la imagen 2.

Imagen 2. Comunicación entre aplicaciones mediante WS.



Nota. Tomado de Programming Web Services with SOAP, por J. Snell, D. Tidwell y P. Kulchenko, 2001, O'Reilly Media, p. 7.

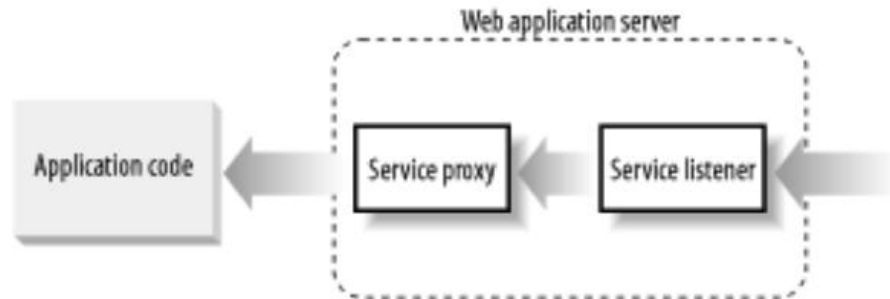
Los WS no son algo nuevo. Más bien, representan la evolución de principios que han guiado el desarrollo de la Internet durante años. Es un marco de mensajería, donde el único requisito para un WS es que pueda enviar y recibir mensajes usando alguna combinación de protocolos estandarizados. Su forma más común es una llamada a procedimientos en un servidor.

El servidor está integrado de diversos componentes. Uno de ellos es el *Service Listener*, éste maneja el protocolo de transporte (HTTP, SOAP, Jabber, etc.) y recibe solicitudes entrantes.

Otro es el *Service Proxy* (Proxy de servicio), el cual decodifica esas solicitudes y realiza las llamadas al código de aplicación. Luego, puede codificar una respuesta

para que el *Service Listener* envíe una respuesta de vuelta, esto es, una respuesta de acuerdo con una petición del cliente, como se aprecia en la imagen 3.

Imagen 3. Arquitectura de un servidor de aplicaciones Web.



Nota. Tomado de Programming Web Services with SOAP, por J. Snell, D. Tidwell y P. Kulchenko, 2001, O'Reilly Media, p. 8.

La arquitectura de un WS se implementa a través de una pila de tecnologías en capas que se construyen unas sobre otras. Esta pila está diseñada con la siguiente estructura:

1. Descubrimiento (*Discovery*). Permite a los consumidores obtener descripciones de servicios.
2. Descripción (*Description*). Un WS debe describir qué protocolo soporta y cómo debe conectarse y utilizarse.
3. Empaquetado (*Packaging*). Para mover datos a través de la red deben empaquetarse en un formato compartido.
4. Transporte (*Transport*). Es la tecnología por medio de la cual permite comunicación directa entre aplicaciones (TCP, HTTP, SMTP y Jabber).
5. Red (*Network*). Esta capa es igual a la del modelo TCP/IP. Se encarga de la comunicación, direccionamiento y enrutamiento.
6. Aplicación (*Application*). Es el código real que implementa la funcionalidad del WS, encontrado y accedido mediante las capas inferiores.

4.2. SOAP

Es un protocolo basado en XML diseñado para facilitar el intercambio de información entre aplicaciones distribuidas. Su funcionamiento se basa en el envío de mensajes estructurados entre un emisor y un receptor, lo que permite implementar mecanismos de comunicación como solicitudes y respuestas (W3C, 2000).

En la pila tecnológica el SOAP se encuentra en la capa de empaquetado para los mensajes que se comparten entre aplicaciones. Mediante un empaquetado simple basado en XML para la información que se transfiere y, un conjunto de reglas para traducir tipos de datos específicos de plataformas y aplicaciones en representaciones XML.

El diseño de SOAP lo hace adecuado para una gran variedad de patrones de integración y mensajería entre aplicaciones.

Como se mencionó anteriormente, SOAP utiliza XML como formato para estructurar los mensajes que intercambian las aplicaciones. Su especificación establece tres componentes principales relacionados con la estructura de los mensajes, las reglas para su procesamiento y la forma en que se representan los datos, los cuales se describen a continuación:

1. Estructura del mensaje. Define una infraestructura para describir qué hay en un mensaje y cómo procesarlo. Está conformada por los siguientes elementos:
 - a. Sobre (*Envelope*): Es la parte que identifica el mensaje SOAP como tal.
 - b. Cabecera (*Header*). Esta parte permite enviar información relativa para procesar el mensaje. Esta etiqueta nos servirá para poder enviar información como más convenga para el aplicativo. Se usa para datos de contexto, autenticación, enrutamiento, transacción, etcétera.
 - c. Cuerpo (*Body*). Contiene la identificación del mensaje de llamada y respuesta, así como sus parámetros.
 - d. *Fault*. Contiene información sobre los posibles errores que puede contener durante el proceso.

Ejemplo de la estructura de un mensaje SOAP. Imagen 4

Imagen 4. Mensaje SOAP

```
<s:Envelope
  xmlns:s="http://www.w3.org/2001/06/soap-envelope">
  <s:Header>
    <m:transaction xmlns:m="soap-transaction"
      s:mustUnderstand="true">
      <transactionID>1234</transactionID>
    </m:transaction>
  </s:Header>
  <s:Body>
    <n:purchaseOrder xmlns:n="urn:OrderService">
      <from><person>Christopher Robin</person>
        <dept>Accounting</dept></from>
      <to><person>Pooh Bear</person>
        <dept>Honey</dept></to>
      <order><quantity>1</quantity>
        <item>Pooh Stick</item></order>
    </n:purchaseOrder>
  </s:Body>
</s:Envelope>
```

Nota. Tomado de Programming Web Services with SOAP, por J. Snell, D. Tidwell y P. Kulchenko, 2001, O'Reilly Media, p. 17.

En este mensaje se aprecia un ejemplo de un aplicativo, el cual, simula una compra, pero, se puede apreciar en su contenido las tres etiquetas fundamentales que conforma un mensaje SOAP. En ocasiones se agrega la etiqueta *Fault*, como se mencionó anteriormente, es para detallar posibles errores que puede presentar el mensaje, cabe mencionar que, no es obligatorio colocarla para enviar un mensaje. Solo se utiliza para dar más detalle si se llega a presentar algún error a la hora de consumir el WS.

2. Reglas de codificación. Definen un mecanismo de serialización que se puede utilizar para intercambiar instancias de tipos de datos definidos por la aplicación. SOAP define un esquema de tipo de datos independiente del lenguaje de programación basado en XSD (*XML Schema Definition*) y normas de codificación para todos los tipos de datos que estén definidos a este modelo.

Las reglas de codificación permiten establecer la forma en que los tipos de datos de una aplicación o plataforma son representados dentro de un mensaje XML. Sin embargo, las diferencias entre las implementaciones de SOAP pueden ocasionar problemas de interoperabilidad cuando los datos son representados o interpretados de manera diferente. El uso de esquemas XML permite establecer una estructura común para la representación de los datos y facilitar su interpretación entre diferentes plataformas (Tidwell, Snell y Kulchenko, 2001, p. 32).

3. Estilos de comunicación. Las comunicaciones pueden seguir el formato RPC (llamada a procedimiento remoto) u orientado a mensajes (Documento). Estos se describen a continuación:

Como XML no está ligado a ningún sistema operativo o lenguaje de programación, los mensajes pueden usarse en cualquier entorno. Por ejemplo: un programa *Perl* en *Windows* puede generar un mensaje XML y enviarlo a una aplicación *Java* en *Unix*.

La idea fundamental es que dos aplicaciones, independientemente de la tecnología, puedan compartir información usando solo un mensaje estructurado que ambas entienden.

4.3. WSDL

WSDL (*Web Services Description Language*) es un formato basado en XML que permite describir las características de un WS. Su definición establece información relacionada con los tipos de datos, mensajes, operaciones, enlaces (*bindings*) y servicios, de manera que un consumidor pueda conocer las funcionalidades proporcionadas por el servicio y la forma en que puede comunicarse con él (W3C, 2001).

La descripción mediante WSDL permite establecer una separación entre las operaciones que proporciona un servicio y los detalles necesarios para realizar la comunicación. De esta manera, el consumidor puede identificar las operaciones disponibles, los mensajes requeridos y el protocolo utilizado para establecer la comunicación con el proveedor (W3C, 2001).

En el presente proyecto, WSDL fue relevante para establecer la integración entre la herramienta desarrollada y el ERP, ya que permitió identificar las operaciones disponibles en el WS que fueron consumidos por la aplicación. En conjunto con SOAP 1.1, esta definición proporcionó la información necesaria para establecer la comunicación entre la herramienta desarrollada y el ERP.

5. Antecedentes del Proyecto

5.1. Funcionamiento del ERP (*Enterprise Resource Planning*)

5.1.1. ¿Qué es un ERP?

Es un sistema de software que ayuda a las organizaciones a optimizar sus procesos de negocio centrales, incluyendo finanzas, RR. HH., fabricación, cadena de suministro, ventas y abastecimiento, compra, recepción e inspección de todos los bienes y servicios que ofrece el negocio, con una visión unificada de la actividad y una única fuente de verdad. (SAP, s. f.).

5.1.2. ERP de la Empresa

Este aplicativo es el encargado de dar seguimiento a la venta de productos de enseñanza. Ofrece la facturación electrónica de los productos adquiridos por el cliente y da seguimiento a los posibles movimientos que puede llegar a tener un contrato, los cuales, pueden ser lecciones contratadas, tomadas o impartidas por un instructor. A su vez, da un panorama de datos que van desde lo financiero hasta tener una bitácora informativa de las lecciones tomadas al día, el nivel que tiene cada estudiante, horarios y las futuras lecciones a impartir, todo esto, a nivel centro. Se entiende por centro todo el conjunto de registros que se encuentran en una escuela de la compañía. Se dividen por países y cada país se engloba en una base de datos, éstas contienen toda la información de contratación de los productos de enseñanza que realicen los clientes.

Este ERP debe ser eficiente, ya que, todos los días se realizan contrataciones de servicios educativos, a su vez, a los productos ya contratados se les hacen modificaciones preliminares denominadas programaciones, tales como asignaciones de instructores, salón de clases, cambios de horario o qué tipo de clase será,

presencial o en línea. Todo esto le da estructura a lo que se denomina lección o servicio pedido por el cliente.

Los posteos son la confirmación de los datos que pueden llegar a modificar el cliente o el instructor, cuyos datos fueron elegidos en el proceso anterior, así como ingresos a cartera, es decir, todos los pagos que puedan llegar a existir por el producto de enseñanza contratado por el cliente.

5.2. Aplicación de un Contrato

La importancia de una buena gestión de un contrato para el ERP es que éste nos dará una vista rápida del idioma, el nivel, el número de lecciones contratadas y tomadas, además del dinero que se abone por el curso comprado, para obtener un reporte con información coherente en tiempo real y lo más realista posible para la empresa. También para tener un panorama efectivo sobre los productos más vendidos, si el cliente está satisfecho o buscar mejoras que puedan optimizar el servicio adquirido.

5.3. Gestión de Contratos

Actualmente el ERP de la empresa se basa en la venta de productos de enseñanza, al cual se le da un seguimiento mediante contratos, éstos pueden ser para un *Customer* o Corporativo. En dichos contratos se dictará el número de lecciones que tendrá el estudiante, ya sea en un curso privado o grupal, dependiendo de las características que pida el cliente.

Diariamente se modifican las lecciones de dichos contratos y el Departamento de Posteo se encarga de programarlas, pero en ocasiones, el estudiante por razones externas pide que se pause su contrato y para evitar que se sigan asignando lecciones se solicita el completado del mismo.

En ocasiones, los grupos finalizan clases y, con el fin de prevenir posteos posteriores a estos contratos, se solicita su completado. Estas tareas se realizaban una a una, generando retrasos en el proceso de posteo e inconsistencia en la información de los contratos finalizados.

6. Definición del problema o contexto de la participación profesional.

Actualmente, el ERP no es capaz de realizar procesos masivos; por consiguiente, la operación realiza este proceso desde el aplicativo Web, donde se busca al cliente por la sede a la que pertenece. Una vez localizado, se identifican los contratos activos y aquellos que finalizaron o deben ser terminados, lo que genera una demora significativa al realizar dicho proceso. Esta tarea se incrementa al final de cada mes, debido a que, para llevar un mejor control de los productos vendidos o por vender, se opta por depurar los contratos que ya no son necesarios o que no se encuentran activos por el desuso del cliente. Esto tiene como finalidad obtener un inventario actualizado de la cantidad de productos al finalizar el mes.

Al principio, este proceso se realizaba mediante un script en SQL (*Structured Query Language*), el cual solo modificaba el estado del contrato directamente en la base de datos. Si bien este procedimiento permitía realizar cambios sobre un gran número de registros, no ejecutaba el flujo funcional utilizado por el ERP para completar un contrato mediante el WS. Como consecuencia, se identificaron fallas e inconsistencias tanto en la información mostrada en el Front End (ERP) y en algunos reportes de inventarios, como en la información que posteriormente se sincronizaba con el sitio Web utilizado para el seguimiento de las actividades y bitácoras de los estudiantes. Esto evidenció que la modificación directa del estado del contrato no era suficiente para garantizar que la información relacionada se actualizara correctamente en los diferentes componentes que dependían de ella.

Además de las inconsistencias identificadas, el volumen de contratos representaba una limitación para realizar la operación de manera individual desde el ERP. Debido a que mensualmente se podían superar los 20,000 contratos, surgió la necesidad de desarrollar una herramienta capaz de realizar el procesamiento masivo, pero que, a diferencia del procedimiento anterior (mediante *queries*), ejecutara la operación mediante los mecanismos de integración proporcionados por el ERP. De esta manera, la solución no se planteó únicamente como una alternativa para reducir el tiempo de procesamiento, sino como un mecanismo para realizar la operación masiva de forma

controlada, considerando las condiciones asociadas a cada contrato y reduciendo el riesgo de generar inconsistencias en el ERP y en los demás aplicativos que dependen de su información.

Como los WS están desarrollados en C# (*C Sharp .NET Framework*) y se comunican a través del protocolo SOAP 1.1, se consideró conveniente utilizar el servicio del ERP para ejecutar la operación de finalización de los contratos. De esta manera, la operación se realiza mediante los mecanismos de integración proporcionados por el ERP, en lugar de modificar directamente la información desde un gestor de base de datos. Esto era importante debido a que el software fue desarrollado por proveedores externos y no se tenía claridad sobre el funcionamiento del código fuente; no obstante, sí se contaba con una idea de cómo funcionaba el Aplicativo Web y de las operaciones que realizaba.

7. Metodología utilizada

Para poder comenzar con la solución propuesta a la problemática identificada en el ERP y, como responsable del desarrollo de una nueva herramienta capaz de realizar procesos masivos, se seleccionó el modelo en cascada (también conocido como Waterfall Model), debido a las características del proyecto y a la naturaleza del problema identificado.

El proyecto partió de un proceso operativo existente y de una problemática claramente identificada a partir de las solicitudes e incidentes relacionados con la finalización masiva de contratos. Esto permitió establecer los requerimientos principales de la solución a partir del análisis del proceso existente.

El modelo en cascada se consideró adecuado debido a que permitió organizar el desarrollo de manera secuencial, estableciendo una relación entre el análisis del problema, el diseño de la solución, el desarrollo de la herramienta, las pruebas, el despliegue y el mantenimiento. Esta estructura permitió avanzar de manera ordenada desde la identificación de las necesidades hasta la validación de la solución propuesta.

Aunque durante las pruebas fue necesario realizar correcciones y optimizaciones, principalmente relacionadas con el tratamiento de diferentes condiciones de los contratos y con el rendimiento del procesamiento masivo, estos ajustes no modificaron el objetivo principal del proyecto. Se realizaron como parte de las actividades de corrección y mejora derivadas de los resultados obtenidos durante la validación.

Por estas características, el modelo en cascada proporcionó una estructura ordenada para el desarrollo de la herramienta, permitiendo establecer las actividades correspondientes a cada etapa y mantener una relación entre el problema identificado, los requerimientos, el diseño, la implementación y las pruebas realizadas para validar su funcionamiento.

7.1. Etapas del modelo aplicado.

7.1.1. Análisis

En esta fase del proyecto, se realizaron reuniones en conjunto con las personas de operaciones que estaban encargadas de completar contratos desde el ERP. Ahí compartieron un desglose de los permisos que se tenían, ya que para habilitar la opción de terminar contratos se requería un perfil en concreto.

En el Aplicativo Web se explicó paso a paso el procedimiento que se ejecutaba para finalizar un contrato. Al mismo tiempo, la duración de la gestión, la cual se realizaba en varias etapas y validaciones para poder culminar un contrato. Además, se mostraron las diferentes advertencias que podía arrojar el sistema por no cumplir las validaciones pertinentes, tales como; la falta de claridad de saldo a favor o en contra del cliente, las lecciones por postear y/o facturas pendientes de pago.

Al poseer certeza de todos los pasos que se debían realizar y, teniendo conocimiento del funcionamiento del ERP para realizar este proceso, con la ayuda de un software externo y gratuito llamado *Fiddler Classic v5.0.20242.10753 for .NET 4.6.2*, que es capaz de captar e inspeccionar el tráfico de red en HTTP y HTTPS, entre el ERP y WS que consumía, nos permitió captar el servicio que se usaba en el proceso descrito por la operación. Concluyendo que, solo era necesario utilizar un servicio del Aplicativo Web, dicho WS está desarrollado en C# con *.NET Framework 3.5*.

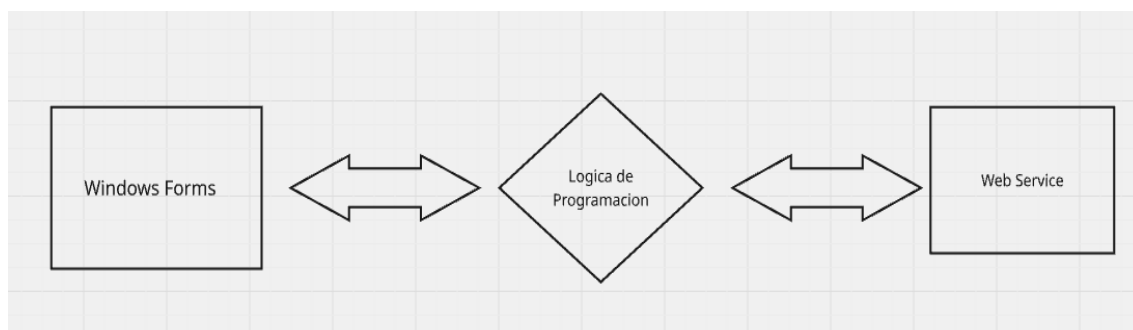
7.1.2. Diseño

Una vez conociendo el proceso manual y sabiendo cómo funcionaba el WS, se comenzó a diseñar la posible herramienta a desarrollar, priorizando que fuera capaz de consumir los servicios de una forma nativa. Por esta razón, se utilizó C# *.NET Framework v4.7.2*, pues sus estructuras coinciden con el mismo lenguaje de programación.

Para agilizar el entorno de interacción con el usuario se optó por usar *Windows Forms*, estas ventanas están diseñadas para agregar componentes sencillos, por ejemplo: botones, etiquetas, cajas de texto, cajas desplegables, etcétera. Todos estos complementos ayudan a crear una interfaz rápida e interactiva. De esta manera, se economiza tiempo en la interfaz y se enfoca el trabajo en la lógica de programación de la herramienta.

El parámetro utilizado para realizar esta configuración se representa en el diagrama de la imagen 5, en el cual, se muestra de manera visual el flujo y los elementos que intervienen para una correcta comunicación.

Imagen 5. Flujo de comunicación entre la aplicación y el WS.



Como se muestra en la imagen 6, éste sería el prototipo básico realizado en *Windows Forms* para la interacción gráfica de la herramienta y así completar contratos de manera masiva:

Imagen 6. Prototipo de interfaz.

Terminación de Contratos

SQL Server:

Database:

Filename:

Registros a procesar: 0 Contratos Completados: 0 Errores: 0

User:

Password:

7.1.3. Desarrollo

En esta etapa de la herramienta se comenzó a dar funcionalidad a los botones y apartados que se mostraron en la imagen 6, los cuales son los siguientes:

En la etiqueta de *SQL Server* se implementó la conexión a la base de datos, como era la fase piloto, todo se realizó para el servidor de pruebas, pero contemplando que éste apartado fuera dinámico y que, con solo meter el dominio del sistema fuera capaz de realizar conexión a producción. Por esta razón se colocó un “*ComboBox*” que desplegará el menú de pruebas y producción.

El apartado de *Database* fue diseñado únicamente para que tuviera conexión con los países que solicitaron la herramienta, en este caso, son: Colombia, México, Chile, Brasil y Ecuador.

Para facilitar el proceso y disminuir las consultas a la base de datos se optó por crear una plantilla sencilla en Excel que estuviera conformada por dos columnas, una donde se coloca el número de contrato y, la otra, donde se guarda una descripción cuando éste se encuentre en gestión. La plantilla proporciona información relevante sucedida

durante el proceso, tales como su correcta terminación o si presentó alguna advertencia que impidiera su finalización (véase la imagen 7).

Imagen 7. Plantilla de Excel

	A	B
1	Contrato	Descripción
2		
3		
4		

El documento en formato Excel es solicitado en la etiqueta *Filename* del prototipo. Para comenzar el proceso se debe guardar el número del contrato en la columna “A”. Una vez realizado el paso anterior se deberá cerrar el archivo mencionado.

Por temas de seguridad es necesario colocar el usuario que la empresa proporciona a cada trabajador para tener acceso al ERP y su respectiva contraseña. Al ser usados los mismos permisos que en el Aplicativo Web solo podrán realizar el proceso los usuarios autorizados por el Departamento encargado de finalizar los contratos.

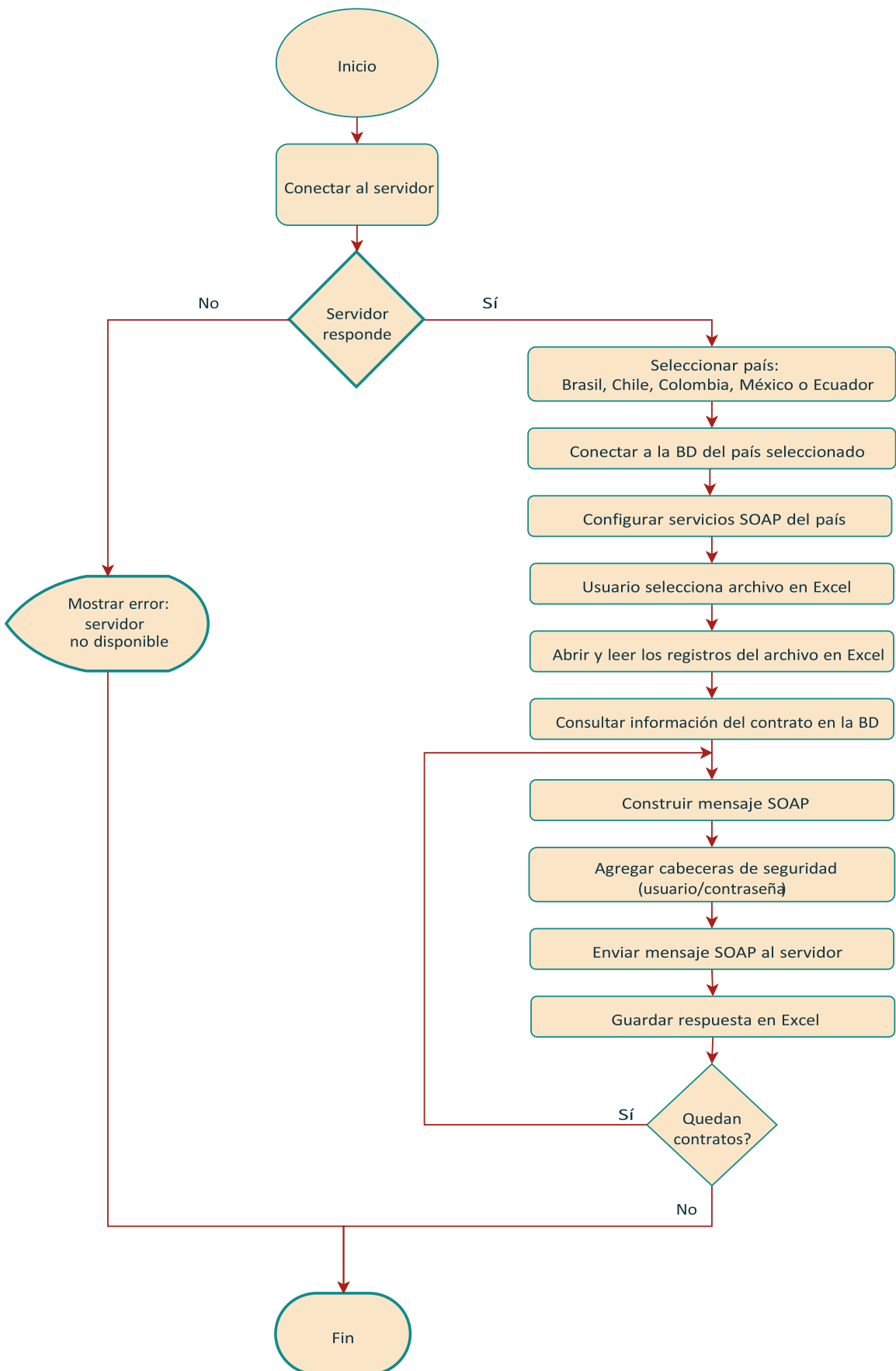
Como se había mencionado anteriormente, estos servicios funcionan mediante el protocolo SOAP 1.1, por esta razón, se implementó un código que logra enviar mensajes en XML, empleando la estructura SOAP 1.1 y que, a su vez, sea capaz de utilizar los WS-Security 1.0 del ERP. Para entender y analizar los datos que solicita el WS del Aplicativo Web, y, con el fin de realizar una petición de forma correcta y construir el mensaje en XML, me apoyé de una herramienta en su versión gratuita, *Fiddler Classic v5.0.20242.10753 for .NET 4.6.2*, que previamente mencioné para identificar el WS.

El proceso realizado en conjunto con el Departamento de Finalización de Contratos, se basó en solicitar la finalización de un contrato de forma manual y, gracias a la herramienta mencionada para la depuración de tráfico de red, se mostró el servicio que consumía para finalizar los contratos, la estructura del mensaje que se debe enviar en formato XML (teniendo en cuenta la estandarización SOAP 1.1) y la forma

en cómo respondía el servicio (la seguridad que tenía los encabezados WS-Security 1.0). Teniendo claro los datos a manejar y el servicio a consumir, se comenzó a diseñar un algoritmo que fuera capaz de utilizar el servicio de manera repetitiva y así lograr un proceso masivo a la hora de completar contratos.

Como la operación siempre proporcionaba el número de contrato, el diseño se redujo a utilizar únicamente el número del contrato y, con él obtener la información necesaria desde la base de datos y construir el mensaje XML para, posteriormente, enviarlo al WS. En este punto se implementó una expresión regular que fuera capaz de distinguir si se trataba de un contrato privado, grupal o corporativo (*Customer* o Corporativo), ya que, actualmente, los datos de dichos contratos se encuentran en diferentes tablas de la base de datos.

Una vez terminado el diseño, se implementó la funcionalidad de las cajas desplegadas, los botones y las etiquetas; todo esto se llevó a cabo desde el lenguaje de programación C#. Para este punto, primero se generó un diagrama de flujo para facilitar el diseño de la nueva herramienta y los elementos que debería tener en cuenta para que el WS se consumiera de manera correcta. A continuación, se muestra dicho diagrama (la información contenida está sujeta a un acuerdo de confidencialidad con la empresa, por ende, se muestra de manera general).



La lógica secuencial del programa se estructura en las siguientes etapas generales:

1. Conexión a la base de datos
2. Extracción de datos
3. Creación del mensaje XML, empleando la estandarización SOAP 1.1 y WS-Security 1.0
4. Utilizar el WS para completar contratos
5. Hacer visible la respuesta del WS
6. Creación de la ventana de Windows Forms asíncrona. En el diseño actual, la ventana no era capaz de mostrar avances sobre el proceso ejecutado, sino que quedaba bloqueada hasta terminar el proceso. Con esta implementación, el usuario puede ver qué cantidad de contratos va a procesar, cuántos lleva procesados y cuántos han tenido alguna advertencia o restricción por parte del sistema que impidió que se completaran de forma correcta.
7. Por último, todas las posibles advertencias que regresa el WS son guardados en el Excel proporcionado, de forma que, en la columna "B" de descripción esté la información de cada uno de los contratos, con el fin de que el usuario, empleando la herramienta, vea qué contratos se completaron o los que, en su defecto, no, e indique el por qué.

7.1.4. Pruebas

La empresa cuenta con un servidor totalmente aislado de producción, por lo que, desde la implementación, se utilizaron los WS y las bases de datos de dicho entorno para realizar las pruebas de la herramienta. Para ello, se solicitó a la operación que proporcionara varios contratos correspondientes a los diferentes países que utilizarían la herramienta, con la finalidad de validar su funcionamiento bajo distintas configuraciones y condiciones. Durante esta etapa se realizaron pruebas funcionales y de procesamiento masivo, en las cuales se identificaron y corrigieron bugs, así como situaciones relacionadas con el consumo de los WS y el acceso a la base de datos.

En las primeras pruebas se identificó un problema de rendimiento, debido a que el nuevo Aplicativo tardaba un minuto o más en completar un contrato. A partir de este

resultado, se analizaron las operaciones realizadas durante el procesamiento y se implementaron mejoras orientadas a optimizar las consultas a la base de datos y el método responsable de realizar las llamadas al WS encargado de finalizar los contratos. Posteriormente, se realizaron nuevas pruebas para comprobar el comportamiento de la herramienta después de las optimizaciones.

Asimismo, se analizaron las advertencias (warnings) devueltas por el servicio SOAP cuando un contrato no cumplía con las condiciones requeridas para su finalización. A partir de estas respuestas, se revisaron los diferentes escenarios y se incorporó a la herramienta el tratamiento correspondiente. Por ejemplo, cuando un contrato presentaba facturación pendiente, se definió un procedimiento controlado para permitir su finalización conforme a las condiciones establecidas por la operación; mientras que, cuando presentaba un saldo a favor o en contra, la herramienta identificaba esta condición e indicaba que era necesario que el contrato se encontrara en cero para poder completarlo. También se verificó el funcionamiento de la herramienta con contratos correspondientes a diferentes países y bajo distintas condiciones de procesamiento. De esta manera, las pruebas permitieron validar tanto el funcionamiento de la solución como su comportamiento ante diferentes escenarios y volúmenes de registros, además de obtener las mediciones de tiempo posteriores a las mejoras realizadas.

7.1.5. Despliegue

El objetivo principal de esta etapa fue poner la herramienta a disposición del Departamento de Finalización de Contratos para atender las solicitudes de procesamiento masivo. Debido a las condiciones de seguridad establecidas por la empresa y a que la herramienta se ejecuta desde un servidor interno, se optó por centralizar su ejecución en el equipo de IT. De esta manera, el Departamento de Finalización de Contratos genera la solicitud mediante el sistema de mesa de ayuda y proporciona los contratos que deben ser procesados, mientras que IT ejecuta la herramienta y gestiona los resultados obtenidos.

Una vez puesta en uso, la herramienta permitió realizar el procesamiento masivo mediante los WS SOAP del ERP, considerando las condiciones asociadas a cada

contrato. Las pruebas de procesamiento integral permitieron obtener un tiempo promedio aproximado de 3.88 segundos por registro, además de comprobar su funcionamiento con las diferentes configuraciones regionales consideradas en el proyecto.

7.1.6. Mantenimiento

Como todo software, siempre existe una nueva manera de optimizar la lógica de programación, por ejemplo: en el consumo del WS, ya que, en ocasiones mandan más de 20,000 contratos a completar y, el servidor donde se encuentra alojado el WS lo toma como una carga excesiva y rechaza las peticiones, saturando el sistema. Actualmente se han corregido estos pequeños detalles y ha mostrado una gran estabilidad, por eso, se ha propuesto como idea final migrar esta herramienta de escritorio a un Aplicativo Web, donde el usuario sea autónomo del software y, con ello, no dependa del Departamento de IT. Esto necesariamente debe ejecutarse así, porque no se cuenta con el código fuente del ERP. Lo ideal sería que se ejecutara desde el aplicativo Web y evitar tener dos herramientas que hagan el mismo proceso.

8. Participación profesional

Mi participación profesional consistió en realizar el análisis, diseño y desarrollo de una herramienta orientada a optimizar el proceso de finalización masiva de contratos realizado en el ERP, el cual se encuentra integrado mediante WS SOAP. Como parte del análisis, revisé el procedimiento utilizado para atender las solicitudes de finalización de contratos e identifiqué las condiciones e inconsistencias que debían considerarse para ejecutar correctamente la operación.

Posteriormente, participé en el diseño de la solución, definiendo el flujo de procesamiento, la comunicación con los WS SOAP del ERP y las consultas necesarias a las bases de datos. También diseñé un mecanismo de parametrización que permitiera trabajar con diferentes configuraciones regionales sin requerir una implementación independiente para cada país.

Durante la etapa de desarrollo, realicé la programación de la herramienta y la integración con los WS SOAP proporcionados por el ERP. Implementé los mecanismos necesarios para procesar los contratos, consultar la información requerida, identificar las condiciones asociadas a cada registro, gestionar las respuestas de los servicios y registrar los resultados obtenidos durante la ejecución.

También participé directamente en la realización de las pruebas funcionales, de procesamiento masivo y de rendimiento. Durante esta etapa analicé las respuestas obtenidas de los WS, incluyendo las advertencias (*warnings*) generadas bajo determinadas condiciones de los contratos, y realicé las correcciones necesarias para atender los escenarios identificados. Asimismo, analicé los factores que afectaban el rendimiento de la herramienta y realicé ajustes en las consultas a la base de datos y en el método responsable de realizar las llamadas al WS.

De esta manera, mi participación comprendió las actividades de análisis del problema, identificación de requerimientos, diseño de la solución, programación, integración con WS SOAP, integración con bases de datos, parametrización, realización de pruebas, corrección de errores y optimización del rendimiento. Estas actividades permitieron aplicar conocimientos de Ingeniería en Computación en el desarrollo de una solución orientada a atender una problemática real dentro de un entorno empresarial.

9. Resultados y aportaciones

9.1. Resultados obtenidos

Evidentemente, el proceso de trabajo se modificó. Actualmente, el Departamento de Finalización de Contratos, encargado de realizar este proceso, genera un *ticket* en el sistema de mesa de ayuda de la empresa, en el cual adjunta un archivo de Excel que contiene los contratos que deben ser terminados. El departamento de IT se encarga de gestionar estas solicitudes mediante la nueva herramienta desarrollada, permitiendo un trabajo conjunto entre ambas áreas y reduciendo los tiempos de gestión de la solicitud y del proceso de finalización.

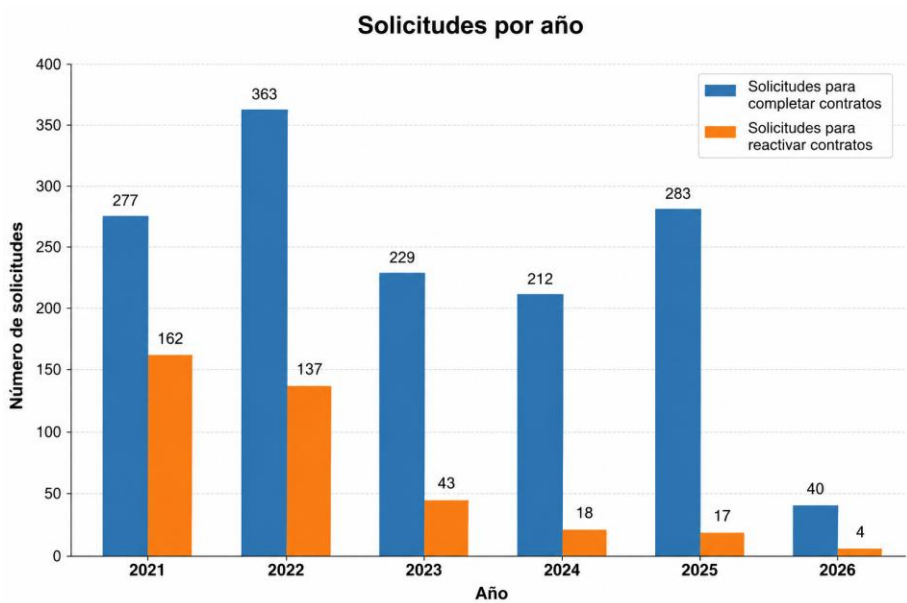
La implementación de la herramienta permitió realizar la operación mediante los WS proporcionados por el ERP, en lugar de depender de modificaciones directas sobre la base de datos. Esto permitió considerar las condiciones asociadas a cada contrato y ejecutar la operación mediante el mecanismo de integración establecido por el ERP, reduciendo el riesgo de generar inconsistencias en la información de los contratos y en los demás aplicativos que dependen de ella.

Como parte de la evaluación de los resultados, se analizaron los registros disponibles en la base de datos del sistema de mesa de ayuda de la empresa, con la finalidad de observar el comportamiento de las solicitudes relacionadas con la finalización y reactivación de contratos durante los años considerados.

Imagen 8. Tickets relacionados con solicitudes de contratos.

Año	Solicitudes para completar contratos	Solicitudes para reactivar contratos
2021	277	162
2022	363	137
2023	229	43
2024	212	18
2025	283	17
2026	40	4

Imagen 9. Gráfica de tickets relacionados con solicitudes de contratos.



Las imágenes 8 y 9 muestran la cantidad de *tickets* cerrados relacionados con solicitudes para completar y reactivar contratos. En los años 2021 y 2022 se registró una mayor cantidad de solicitudes para reactivar contratos. El desarrollo de la herramienta comenzó en febrero de 2022 y pasó a producción a finales de ese mismo año. A partir de 2023, periodo en el que la herramienta ya se encontraba en uso, se observa una disminución en este tipo de solicitudes.

Este comportamiento representa un cambio en la forma de atender la operación, debido a que la reactivación de contratos dejó de depender exclusivamente de modificaciones directas mediante *queries* en la base de datos. Con la herramienta desarrollada, esta operación puede realizarse mediante los mecanismos funcionales proporcionados por el ERP.

9.2. Resultados de las pruebas de rendimiento

Además de la validación funcional, se realizaron pruebas de rendimiento para evaluar el comportamiento de la comunicación entre la herramienta y los servicios SOAP del ERP. Estas mediciones corresponden exclusivamente al tiempo de respuesta del servicio SOAP y, por lo tanto, no representan el tiempo total requerido por la aplicación para procesar cada contrato.

Las pruebas se realizaron con registros correspondientes a México, Colombia, Brasil, Ecuador y Chile, configuraciones para las cuales fue desarrollada la herramienta.

País	Registros	Tiempo promedio	Tiempo mínimo	Tiempo máximo
México	488	1,141.31 ms	596 ms	3,355 ms
Colombia	120	952.14 ms	605 ms	1,476 ms
Brasil	60	810.09 ms	595 ms	1,372 ms
Ecuador	100	1,133 ms	521 ms	2,102 ms
Chile	67	910.09 ms	637 ms	1,151 ms

Los resultados muestran que el tiempo promedio de respuesta del servicio SOAP se mantuvo alrededor de un segundo en las diferentes pruebas realizadas. El menor promedio se obtuvo en Brasil, con 810.09 ms, mientras que el mayor correspondió a

México, con 1,141.31 ms. El tiempo mínimo registrado entre las pruebas fue de 521 ms en Ecuador y el máximo fue de 3,355 ms en México.

En conjunto, se analizaron 835 llamadas al servicio SOAP correspondientes a las cinco configuraciones regionales. Estos resultados permiten observar el comportamiento de la comunicación con el ERP bajo diferentes configuraciones, aunque las cantidades de registros utilizadas en cada país son diferentes, por lo que los resultados se presentan como una referencia descriptiva y no como una comparación estadística entre países.

9.3. Análisis detallado del servicio SOAP

Como parte de las pruebas de rendimiento se realizó un análisis detallado de 120 llamadas al WS de Colombia. El tiempo promedio obtenido fue de 952.14 ms (0.952 segundos) por llamada, con una mediana de 954 ms, un tiempo mínimo de 605 ms y un máximo de 1,476 ms.

De las 120 llamadas realizadas, 72 (60.0 %) finalizaron en un segundo o menos, mientras que 48 (40.0 %) superaron dicho tiempo. El tiempo acumulado de las llamadas fue de 114.26 segundos (1 minuto y 54.26 segundos).

Estos resultados permiten observar que el servicio presentó un comportamiento relativamente estable durante la prueba, al concentrarse la mayoría de las respuestas alrededor de un segundo. Es importante señalar que estos valores corresponden únicamente a la comunicación con el servicio SOAP y no incluyen las demás operaciones ejecutadas por la herramienta.

9.4. Resultados del procesamiento integral de la aplicación

Para evaluar el comportamiento real de la herramienta también se realizó una medición del procesamiento integral. A diferencia de las pruebas anteriores, esta medición considera las operaciones adicionales que intervienen durante la ejecución, como las llamadas a otros servicios, consultas a las bases de datos, validaciones, procesamiento de información y almacenamiento de los resultados.

En una prueba realizada con 577 registros, el procesamiento completo tuvo una duración de 37 minutos con 18.75 segundos, equivalente a un promedio aproximado de 3.88 segundos por registro.

Por lo tanto, se distinguen dos indicadores de rendimiento: el tiempo de respuesta del servicio SOAP y el tiempo integral de la aplicación. En las pruebas realizadas, el servicio SOAP presentó un promedio de 0.952 segundos por llamada en Colombia, mientras que el procesamiento integral presentó un promedio aproximado de 3.88 segundos por registro en México. La diferencia se debe a las operaciones adicionales que realiza la herramienta durante el procesamiento.

Esta medición permite evaluar el rendimiento de la solución desde una perspectiva más cercana a su utilización real para el procesamiento masivo.

9.5. Incidentes e inconsistencias

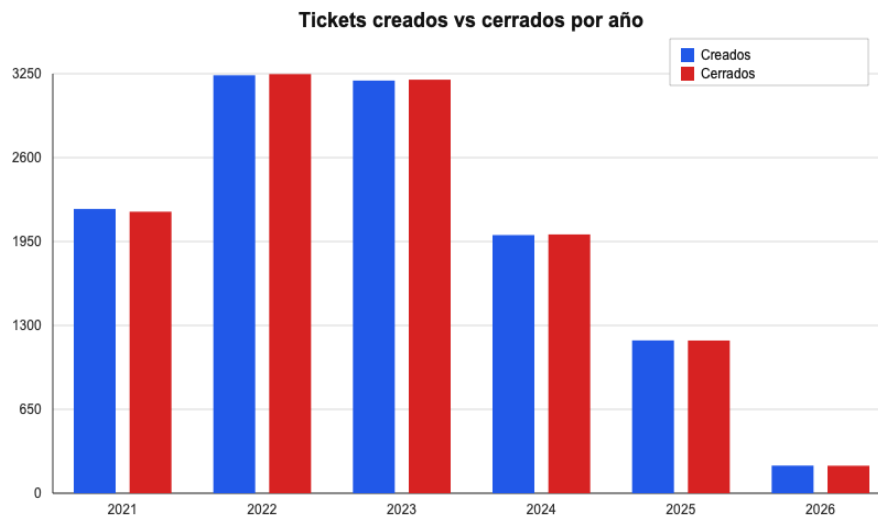
El desarrollo no solo tuvo impacto en la operación de finalización y reactivación de contratos, sino también en los incidentes relacionados con inconsistencias entre la información del ERP y los sistemas que dependen de ella.

Para analizar este comportamiento, se consultó la información disponible en la base de datos del sistema de mesa de ayuda de la empresa, tomando como punto inicial mayo de 2021.

Imagen 10. Incidentes creados vs. cerrados por año.

Año	Incidentes creados	Incidentes cerrados
2021	2202	2181
2022	3238	3247
2023	3197	3204
2024	2000	2005
2025	1184	1183
2026	214	213

Imagen 11. Gráfica de incidentes.



En la imagen 11 se observa el comportamiento de los incidentes creados y cerrados durante el periodo analizado. Los datos muestran una disminución en el número de incidentes registrados en los periodos posteriores a la implementación y adopción de la herramienta.

Esta información representa un indicador complementario del cambio realizado en el proceso. Al utilizar los WS proporcionados por el ERP en lugar de modificar directamente el estado de los contratos mediante queries, la operación se ejecuta mediante el flujo funcional establecido por el sistema. Esto contribuye a disminuir el riesgo de que la información relacionada con los contratos quede desactualizada y afecte a otros componentes que dependen de ella, como los reportes del ERP y el sistema utilizado para el seguimiento de las actividades y bitácoras de los estudiantes.

9.6. Aportación de la solución

La principal aportación del proyecto fue desarrollar una herramienta capaz de atender la necesidad de realizar la finalización masiva de contratos sin depender de modificaciones directas sobre la base de datos. La solución utiliza los WS SOAP proporcionados por el ERP y permite procesar grandes volúmenes de información considerando las condiciones asociadas a cada contrato.

Otra aportación importante fue la implementación de un mecanismo de parametrización que permite trabajar con las diferentes configuraciones regionales

consideradas en el proyecto. Las pruebas realizadas con México, Colombia, Brasil, Ecuador y Chile permitieron validar el funcionamiento de la herramienta bajo estas configuraciones.

Asimismo, el análisis de las respuestas del servicio SOAP permitió incorporar el tratamiento de diferentes condiciones de los contratos. La herramienta puede identificar situaciones como facturación pendiente o saldos diferentes de cero y proporcionar el tratamiento correspondiente de acuerdo con las condiciones establecidas para la operación.

La solución también permitió establecer un mecanismo de trabajo entre el departamento de Finalización de Contratos y el departamento de IT. Mediante este esquema, la operación proporciona los contratos que requieren procesamiento y IT ejecuta la herramienta para realizar la operación de manera controlada.

Finalmente, el desarrollo permitió atender la limitación relacionada con el procesamiento masivo y reducir la dependencia del procedimiento anterior basado en modificaciones directas mediante queries. De esta manera, se proporcionó una alternativa que aprovecha los mecanismos de integración existentes en el ERP y que contribuye a reducir el riesgo de inconsistencias en la información utilizada por el propio sistema y por los demás aplicativos que dependen de ella.

10. Conclusión

El desarrollo del presente proyecto permitió atender una problemática operativa relacionada con la finalización masiva de contratos en el ERP corporativo. A partir del análisis realizado, se identificó que el procedimiento utilizado anteriormente, basado en modificaciones directas mediante queries sobre la base de datos, no reproducía el flujo funcional utilizado por el ERP y podía generar inconsistencias en la información del sistema y en otros aplicativos que dependían de ella. Ante esta situación, se desarrolló una herramienta capaz de realizar el procesamiento masivo mediante el consumo de los servicios SOAP proporcionados por el ERP.

Durante el desarrollo, el consumo de los WS permitió identificar, mediante las respuestas proporcionadas por el ERP, que los contratos debían cumplir determinadas condiciones para poder ser finalizados correctamente. El análisis de las advertencias (warnings) permitió reconocer escenarios como facturación pendiente, saldos a favor o en contra y condiciones relacionadas con las lecciones del estudiante. A partir de estos resultados, se incorporaron a la herramienta los mecanismos necesarios para identificar y tratar dichas condiciones durante el procesamiento.

La solución permitió mantener la capacidad de procesar múltiples contratos y, al mismo tiempo, utilizar los mecanismos de integración proporcionados por el ERP. Asimismo, mediante la parametrización implementada, fue posible adaptar la herramienta a las diferentes configuraciones regionales consideradas durante el proyecto. Esto permitió utilizar una misma solución para México, Colombia, Brasil, Ecuador y Chile, sin requerir una implementación independiente para cada país.

Las pruebas realizadas permitieron validar el funcionamiento de la herramienta bajo diferentes condiciones y configuraciones. En las pruebas de comunicación con el servicio SOAP se analizaron 835 llamadas correspondientes a los cinco países considerados. Los tiempos promedio obtenidos fueron de 1,141.31 ms para México, 952.14 ms para Colombia, 810.09 ms para Brasil, 1,133 ms para Ecuador y 910.09 ms para Chile. De manera complementaria, en una prueba de procesamiento integral con 577 registros, se obtuvo un tiempo total de 37 minutos con 18.75 segundos, equivalente a un promedio aproximado de 3.88 segundos por registro. Estos resultados permitieron evaluar tanto el comportamiento de la comunicación con el ERP como el rendimiento de la herramienta durante el procesamiento completo.

En cuanto al impacto operativo, los registros consultados en el sistema de mesa de ayuda permitieron observar una disminución en las solicitudes de reactivación de contratos y en los incidentes relacionados con inconsistencias después de la implementación de la herramienta. Estos resultados representan un indicador del cambio realizado en el proceso, aunque su comportamiento debe interpretarse considerando que pueden existir otros factores que influyan en la cantidad de solicitudes e incidentes registrados.

Desde el punto de vista profesional, el proyecto permitió aplicar conocimientos de Ingeniería en Computación en las diferentes etapas de desarrollo de una solución empresarial, desde el análisis del problema y la identificación de requerimientos hasta el diseño, programación, integración con WS SOAP, acceso a bases de datos, parametrización, pruebas, corrección de errores y optimización del rendimiento. También permitió comprender la importancia de analizar el funcionamiento de un sistema empresarial antes de realizar modificaciones directamente sobre su información.

Uno de los principales aprendizajes obtenidos fue que una solución para un proceso masivo no debe considerar únicamente la cantidad de registros que puede procesar, sino también la forma en que la operación se encuentra implementada dentro del sistema. En este caso, el análisis de las respuestas de los WS permitió conocer las condiciones necesarias para completar los contratos y adaptar la herramienta para atenderlas. Esto demostró que es posible desarrollar una solución externa que amplíe las capacidades de un software empresarial sin modificar directamente sus servicios existentes, utilizando los mecanismos de integración que este proporciona.

Finalmente, con base en el análisis realizado, el desarrollo de la herramienta, las pruebas efectuadas y los resultados obtenidos, se considera que el objetivo general fue cumplido. La solución permitió automatizar la finalización masiva de contratos mediante el consumo de servicios SOAP, considerando las condiciones identificadas para cada contrato y proporcionando un mecanismo controlado para ejecutar la operación. Con ello, se atendió la limitación de procesamiento masivo del ERP y se redujo la dependencia de modificaciones directas sobre la base de datos, contribuyendo a disminuir el riesgo de inconsistencias en el ERP y en los demás aplicativos que dependen de su información.

11. Bibliografía

- IBM. (2023). Web services overview. IBM Documentation.
<https://www.ibm.com/docs/en/integration-bus/10.0?topic=overview-web-services>
- Microsoft. (s. f.). Descripción general de Windows Forms.
<https://learn.microsoft.com/es-es/dotnet/desktop/winforms/overview/>
- Progress Telerik. (s. f.). Fiddler Classic. <https://www.telerik.com/fiddler/fiddler-classic>
- SAP. (s. f.). ¿Qué es ERP?
<https://www.sap.com/latinamerica/products/erp/what-is-erp.html>
- Tidwell, D., Snell, J., & Kulchenko, P. (2001). Programming Web Services with SOAP. O'Reilly.
- World Wide Web Consortium. (2000). Simple Object Access Protocol (SOAP) 1.1. <https://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- World Wide Web Consortium. (2001). Web Services Description Language (WSDL) 1.1. <https://www.w3.org/TR/2001/NOTE-wsdl-20010315>
- World Wide Web Consortium. (2007). Position paper: Toward a hybrid solution for the Web of services. <https://www.w3.org/2007/01/wos-papers/iona>