

ARTICLE HISTORY

Received 15 September 2020
Accepted 02 November 2020

Jhon Calle

Escuela de Ingeniería de Sistemas y
Telemática
Universidad del Azuay
Cuenca, Ecuador
john6ero@es.uazuay.edu.ec

Pablo Lója

Escuela de Ingeniería de Sistemas y
Telemática
Universidad del Azuay
Cuenca, Ecuador
arevalop17@es.uazuay.edu.ec

Marcos Orellana

Laboratorio de Investigación y Desarrollo
en Informática (LIDI)
Universidad del Azuay
Cuenca, Ecuador
marore@uazuay.edu.ec

Priscila Cedillo

Laboratorio de Investigación y Desarrollo
en Informática (LIDI)
Universidad del Azuay
Cuenca, Ecuador
priscila.cedillo@ucuenca.edu.ec

Orquestación de Servicios RESTful y SOAP: Caso de estudio, asignación de visas Americanas.

RESTful and SOAP Services Orchestration: Case Study, American Visa Assignment.

Orquestación de Servicios RESTful y SOAP: Caso de estudio, asignación de visas Americanas.

RESTful and SOAP Services Orchestration: Case Study, American Visa Assignment.

Jhon Calle

Escuela de Ingeniería de
Sistemas y Telemática
Universidad del Azuay
Cuenca, Ecuador
john6ero@es.uazuay.edu.ec

Pablo Lója

Escuela de Ingeniería de
Sistemas y Telemática
Universidad del Azuay
Cuenca, Ecuador
arevalop17@es.uazuay.edu.ec

Marcos Orellana

Laboratorio de
Investigación y Desarrollo
en Informática (LIDI)
Universidad del Azuay
Cuenca, Ecuador
marore@uazuay.edu.ec

Priscila Cedillo

Laboratorio de Investigación y
Desarrollo en Informática (LIDI)
Universidad del Azuay
Cuenca, Ecuador
priscila.cedillo@ucuenca.edu.ec

Resumen— Este artículo presenta el análisis de un caso de asignación de visas americanas mediante la extracción de información de diferentes entidades. Para este problema se utilizan las siguientes fuentes de datos: i) información personal del solicitante, ii) información de los bienes inmuebles de la persona, y iii) el historial crediticio proveniente de entidades bancarias. Para la extracción de la información se crearon interfaces que simulaban el funcionamiento de dichas entidades, así como sus respectivas bases de datos y servicios web. Para el envío de datos, se implementó en la aplicación, el uso de una arquitectura orientada a servicios (SOA) mediante el desarrollo de un sistema de orquestación de servicios. En este contexto, se aplicaron conceptos del lenguaje BPEL, cuya finalidad es demostrar la importancia de una orquestación de servicios en empresas u organizaciones y la gran utilidad de disponer de servicios integrados. Los conceptos aplicados en el desarrollo del sistema final, corroboran su utilización, puesto que los módulos desarrollados, pueden integrarse a otros sin que exista la necesidad de volver a diseñarlos desde cero, lo que representa una ayuda sustancial en la búsqueda de soluciones óptimas para infraestructuras TI.

Palabras Clave— BPEL, Orquestación, Servicios Web, SOA, SOAP, Visas.

Abstract— This article presents the analysis of a case of U.S. visa assignment by extracting information from different entities. The following data sources are used for this issue:

(i) personal information of the applicant, (ii) information on the person's real estate, and (iii) credit history from banks. For the extraction of the information, interfaces were created that simulated the operation of these entities and their respective databases and web services. For data submission, the use of a service-oriented architecture (SOA) by developing a service orchestration system was implemented in the application. In this context, BPEL language concepts were applied, which aim to demonstrate the importance of service orchestration across enterprises or organizations and the great utility of having integrated services. The concepts applied in the development of the final system corroborate its use, since the modules developed can be integrated with others without the need to redesign them from scratch, which is a substantial aid in the search for optimal solutions for IT infrastructures.

Keywords— BPEL, Orchestration, SOA, SOAP, web services, Visas.

I. INTRODUCTION

En la actualidad, los servicios web son de gran utilidad para el desarrollo e integración de diversas plataformas y entornos heterogéneos distribuidos. A lo largo de estos años, los servicios web han permitido el acceso a la información requerida, para tratar con todo tipo de negocio en cualquier momento y en cualquier lugar; estos servicios se pueden describir como una aplicación o parte de la misma que la podemos encontrar en la web [1].

Constituyen entonces, módulos de software que se auto describen y que poseen autonomía; permiten realizar tareas concretas, además de caracterizarse por un fácil despliegue ya que se basan en tecnologías estándar HTML y HTTP [2].

Si bien, la orquestación de servicios web puede ser descrita como una integración de servicios; ésta puede ser interpretada también como una implementación de procesos por combinación de servicios en operaciones comerciales [3], permitiendo la creación de los mismos en alto nivel. Con estas características, la orquestación se basa en las interacciones entre servicios internos y externos que se dispongan para la ejecución de diversas tareas, mismas que son necesarias para el correcto funcionamiento de un sistema de información [4].

Un punto a considerar son las limitaciones de tiempo para establecer una consistencia temporal [4]. Entonces, es de suma importancia el uso de métodos unificados con sintaxis y semántica definida, con el objetivo de verificar y validar las propiedades que estos cumplen; para ello, es necesaria una arquitectura orientada a servicios (SOA), la cual permite crear sistemas escalables, facilitando la interacción ya sea en sistemas propios o de terceros, proporcionando ventajas; entre ellas: reutilización, agilidad y acoplamiento; esto, con la ayuda de una colección de servicios [5].

La implementación de una arquitectura basada en procesos, se ha venido implementando desde hace algunos años en organizaciones TI. Los principales beneficios que brinda SOA son la agilidad comercial y la reutilización. Cuando los requerimientos del negocio cambian, el costo de mantener la solución sincronizada puede ser muchas veces mayor que el de construir la solución desde sus componentes modulares. Si una solución se entrega con un enfoque SOA, junto con la gestión de procesos empresariales y las tecnologías de servicios web, se puede gestionar ágilmente con relación a las ya construidas con un enfoque tradicional [6]. El uso de una orquestación en modelos empresariales ya definidos, conlleva un cambio radical en los mismos, debido a que estos tienen que ser creados de forma general [1].

Un uso práctico que se puede dar a la orquestación es la inclusión de flujos de trabajos, muy usados dentro de los procesos de negocios. Gracias a que la orquestación está basada en una arquitectura de servicios, puede aplicarse a una arquitectura empresarial (EA - Enterprise Architecture); ésta coopera con una visión más abstracta de los activos de

una empresa de TI. El enfoque SOA permite crear redes de sistemas que interactúan entre sí, y a los mismos se les puede asignar un flujo de trabajo, es decir, dotarlos de una jerarquía dentro de la empresa. El orquestador contiene la definición de flujos de trabajo y los ejecuta. La primera categoría del sistema representa la capa de flujo de trabajo, mientras que la segunda, se encuentra representada por la capa SOA, que expone los servicios a la capa de flujo de trabajo. De acuerdo a este modelo de orquestación para la automatización de flujos de trabajo, todo el sistema puede considerarse como un gráfico dirigido en el que el conductor central organiza los flujos entre varios sistemas participantes [7].

Para la realización de una orquestación en el análisis de este caso particular, se realizó un enfoque centrado en los servicios, es decir, los mismos deben poseer interfaces WSDL, para que de este modo puedan ser incluidos dentro del motor BPEL, dado que los servicios RESTful no utilizan WSDL, se presentaron dificultades al momento de integrarlos al BPEL [8] [18]. Mediante WSDL se puede definir de mejor manera un mensaje SOAP, y en caso de interactuar con un servicio del mismo tipo ya creado, solo es necesario que se proporcione el WSDL que funge como contrato entre cliente y proveedor. El uso de servicios SOAP es muy útil en entornos de tipo empresarial distribuidos, en cambio REST, está más encaminado a una conexión directa punto a punto. Además como punto a favor de SOAP, mediante la seguridad origen-destino, que la misma proporciona, es de mayor utilidad en la creación y uso de servicios de tipo asincrónico.

En este artículo, se presenta la simulación de un sistema de obtención de visas, el cual, según el historial crediticio de las personas, calificará a una persona a fin de saber si esta puede obtener una visa Americana. Para la esta simulación, se dispone de un sistema de registro, el cual contiene la información completa de la persona, a su vez, un sistema de comprobación de deudas, donde se verifica su historial crediticio, esto ayudará a determinar si puede o no calificar para la visa Americana. Los indicadores que se tomarán en cuenta son: las propiedades hipotecadas que disponga el solicitante y las deudas pendientes que disponga con alguna entidad bancaria; en ambos casos, si el solicitante dispone de al menos una hipoteca o varias deudas pendientes, cabe la posibilidad que la persona trata de evadir estas responsabilidades saliendo del país, por lo que la solicitud será cancelada hasta que sus movimientos financieros mejoren. Cada sistema se desarrollará en diferentes ambientes de programación y contarán con el

uso de servicios RESTful para la obtención de la información requerida, servicios SOAP para la creación de los micro servicios necesarios, y un sistema gestor de base de datos en donde se almacenará toda la información necesaria y requerida.

Para cumplir con este objetivo, es necesario que los servicios trabajen de forma conjunta, ordenada, y eficaz; lo que en conjunto forma parte de un proceso global que cumple con el objetivo primordial. Es necesario resaltar que si existen fallas en la implementación de la orquestación, se podría generar errores en el sistema tales como: fallos en el envío y/o recepción de información, o el envío de información errónea.

Partiendo de estas premisas el este artículo se encuentra organizado de la siguiente manera. La sección 2 describe todo el procedimiento y las herramientas que fueron necesarias tanto para la construcción del entorno de simulación, así como el desarrollo de la orquestación de los servicios involucrados, detallando de manera concisa cada una de las etapas a seguir. La sección 3 describe los resultados y observaciones que se obtuvieron al realizar la simulación. Finalmente, la sección 4 presenta las conclusiones y el trabajo futuro.

II. MATERIALES Y MÉTODOS

A. Modelamiento de la solución.

Para el desarrollo del caso de estudio, se debe considerar las siguientes restricciones: una persona calificará para la obtención de una visa, dependiendo de su historial crediticio en todas las entidades bancarias en las que posea una cuenta, y de las propiedades que la persona disponga y si éstas a su vez, no se encuentran hipotecadas.

Para obtener estos datos, se requiere consultas de las siguientes entidades: Registro Civil, Registraduría de Propiedad, y Superintendencia de Bancos. Para ejecutar la simulación, se han creado bases de datos en distintos gestores, y aplicaciones que simulan su funcionamiento. Para obtener la información de cada fuente, se crearon servicios web, que se encargaron de realizar la funcionalidad de cada entidad.

La orquestación de estos servicios se realizó con el lenguaje denominado BPEL, lo que permite que los servicios estén disponibles y a la espera que sean utilizados por el orquestador. A su vez se devuelve el conjunto de información para su posterior evaluación, es decir, si la persona califica o no para la visa (Fig. 1.).

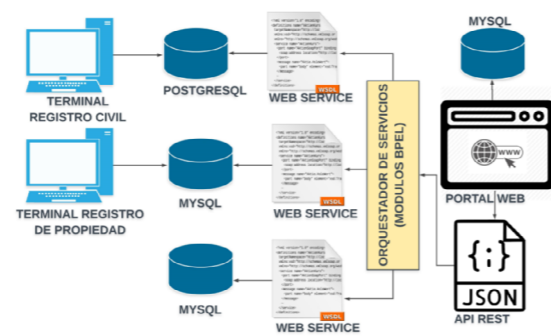


Fig. 1. Arquitectura Servicios.

La Fig. 1. muestra la topología utilizada para la orquestación, en donde se observa los diferentes elementos que la componen, entre ellos, los diferentes terminales correspondientes a las entidades: Registro Civil, y Registraduría de propiedad, y Portal web del sistema de asignación de visas, cada una con su base de datos y sus servicios web. Y en el centro el orquestador que es el encargado de dar funcionalidad al sistema.

B. Creación de las Bases de Datos

Con relación a los gestores de bases de datos, se ha utilizado dos diferentes, MySQL y Postgres, los cuales almacenan la información pertinente a cada entidad (ver Fig.1).

La siguiente sección, realiza una descripción detallada de la composición de cada una de las bases de datos creadas:

- Aplicativo Registro Civil: Utilizar Postgres como gestor de base de datos, misma que cuenta con un objeto de tipo tabla denominado persona, la cual almacena la información de la misma, los campos que componen dicha tabla son: id, número de identificación, nombres, apellidos, sexo, instrucción, nivel académico y estado civil.

- Aplicativo Registraduría de Propiedad: En este caso el gestor usado es MySQL. La base de datos cuenta con dos objetos de tipo tabla: persona y bienes, la primera registra la información básica de un usuario (cedula, nombres, apellidos, entre otros) mientras que, la segunda contiene la información de bienes; en ella se pueden apreciar los siguientes campos: código de propietario, dirección, tamaño, cotización, fecha de registro, tipo de inmueble y un campo adicional sobre el estado del inmueble, es decir, si posee alguna hipoteca.

- Base de datos de créditos: Se refiere al almacén de datos en donde se encuentra la información crediticia de los usuarios, esta al igual que la tabla anterior se encuentra

en el gestor MySQL y cuenta con una sola tabla, misma que contendrá los siguientes campos: id, nro de identificación de usuario, banco asociado, tipo de cuenta o responsabilidad y finalmente el estado de la misma.

- Aplicativo web sistema de visas: Para este aplicativo se ha usado como gestor a MySQL, este almacena información del usuario, y la información referente a la decisión de otorgar o no la visa de acuerdo a los factores evaluados, es decir, su estado de aprobación.

Cabe recalcar que para la simulación e implementación se han creado tablas con la cantidad de columnas necesarias; en la realidad estos sistemas requieren más columnas por cada usuario ingresado y por ende de un esquema de mayor magnitud. Además, se debe también considerar que al ser entidades diferentes que gestionan su propia información, pueden llegar a ser similares muchas de las columnas, por el hecho mismo de que gestionan su propia información.

C. Desarrollo de Aplicativos

El desarrollo de los aplicativos se ha realizado usando diferentes lenguajes de programación:

- Java: Con este lenguaje se ha desarrollado la aplicación de escritorio para el Registro Civil. Para este desarrollo se ha empleado el entorno de desarrollo Netbeans; además, esta aplicación contempla el paradigma de programación orientado a objetos (POO) para el manejo de información de usuarios. Además, para la conexión a la base de datos Postgres se ha usado el driver denominado PostgreSQL JDBC, permitiendo realizar un mantenimiento (CRUD - Create, Read, Update, Delete) de las tablas [9].

- Visual Basic: Se ha realizado el aplicativo de escritorio en donde se registra la propiedad del usuario, el entorno de desarrollo usado para la implementación es Visual Studio. De manera similar, se ha usado POO para gestionar de manera fácil y correcta la información, tanto de usuarios como de bienes [10].

Finalmente el desarrollo del aplicativo web está compuesto de la siguiente manera:

- Backend: Se ha utilizado PHP como lenguaje de programación para accionar los diferentes componentes HTML. En este caso, para consultar la información del usuario y almacenarla [11].

- FrontEnd: Adicionalmente, para el diseño se ha utilizado un Framework CSS

denominado Bootstrap, que ofrece clases con estilos previamente realizados, y que pueden aplicarse fácilmente, esto con el objetivo de que el usuario disponga de una mejor interacción con el sistema [12].

D. Creación de Servicios Web

Una vez creados los diferentes aplicativos que simulan las entidades a ser evaluadas, se procede con el desarrollo de los servicios web que solicitarán y proporcionarán la información correspondiente. Para ello, se ha optado por crear servicios de tipo SOAP y REST.

Para consumir la información del Registro Civil, Registraduría de propiedad e historial de crédito, se ha realizado servicios del tipo SOAP y la codificación en lenguaje php, ya que este ofrece un kit de herramientas para desarrollo e implementación de este tipo. A este tipo de herramienta se le conoce como NuSOAP, la misma que está basada en SOAP 1.1, WSDL 1.1 y HTTP 1.0/1.1. Además, la herramienta está compuesta con una serie de clases que facilitan la implementación tanto de servidores (proveen servicios) como de clientes (consumen servicios). Algo particular y a considerar es que se ha optado el uso de esta herramienta, debido a que el componente BPEL proporcionado por OpenESB no acepta otras versiones de SOAP.

Los servicios mencionados, se han creado únicamente para consulta, estos tienen como parámetro de entrada el número de identificación del usuario, en caso de existir se devuelve la información completa del mismo. Por ejemplo. En el primer caso se devuelve toda información de usuario que se encuentra en el Registro Civil, por otra parte, el segundo devuelve un listado de bienes registrados correspondientes al usuario, y finalmente se devuelve todo su historial crediticio. Toda esta información coopera para tomar la decisión en cuanto a la aprobación de visas.

Por otra parte, el aplicativo web implementa servicios de tipo REST, con el objetivo de no centrarse en un solo tipo de servicio, ya que una de las principales ventajas que BPEL ofrece es la integración de servicios previamente creados. Entonces, para la implantación, es necesaria la instalación de un micro framework denominado Slim con el administrador de dependencias denominado Composer, luego es necesaria la configuración de un fichero denominado .htaccess el cual contiene las directivas de comportamiento del servidor. A continuación, se indica la definición del archivo:

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^ index.php [QSA,L]
```


La primera línea activa el motor de redirecciones, las siguientes dos indican que, si el archivo con el nombre especificado en el navegador no existe, o el directorio en el navegador no existe, entonces proceda a la regla de reescritura siguiente.

Luego de configurado el fichero se procede a crear el servidor, es importante definir los tipos de métodos aceptados por el servicio, en este caso se ha definido de la siguiente manera: ('Access-Control-Allow-Methods', 'GET, POST, PUT, DELETE, PATCH, OPTIONS').

Finalmente, se procede a la creación de los métodos que componen los servicios y registrarlos en el aplicativo; lo que comprende dos principales, el primero, enviará el número de identificación al WSDL proporcionado por BPEL y retornará la información del usuario en un texto con un formato de tipo JSON, mismo que será útil para visualizarlo en el aplicativo web y posteriormente generar el reporte final. Por otra parte, el segundo servicio es de ingreso de datos, es decir, una vez generado el reporte y asignado el estado de aprobación se procede a guardar los datos en la base local.

E.Implementación de BPEL

La etapa relevante es la implementación de la orquestación de servicios; como se mencionó con anterioridad, se ha utilizado un conjunto de herramientas que permiten su resolución y aplicación. Es importante recalcar que el módulo BPEL que se encuentra en OpenESB ofrece los tipos de variable estándar, de modo que, su uso es limitado, debido a normalmente se cuenta con una estructura de datos compleja, es decir, pueden ser objetos o un conjunto de ellos, sin embargo, la herramienta ofrece la posibilidad de crear esquemas XML propios con el fin de definir a criterio propio los tipos de datos a utilizar. Además, el módulo BPEL ofrece una vista de diseño de la aplicación, lo cual facilita la creación de estos procesos, algo a tomar en cuenta es que todos los servicios externos que se han añadido al módulo están descritos en WSDL (Servicios SOAP).

Como primer paso de la implementación se han agregado los diferentes servicios web tipo SOAP, mismos que se encargan de validar y enviar los datos desde las siguientes entidades: Registro Civil, Registraduría de Propiedad y la Superintendencia de Bancos. Cada uno de ellos tiene un parámetro de entrada con el cual retorna la información si encuentra coincidencia en el identificador. Posteriormente, se procede a crear un documento WSDL al cual se le asigna un nombre, el mismo que es necesario para su posterior invocación, como recomendación es preferible un nombre acorde al funcionamiento

del servicio, en este estudio se definirán tanto el parámetro de entrada como el de salida y se elegirán los tipos de datos a usar.

En BPEL existen varias actividades que se pueden realizar. A continuación, se listan las utilizadas para la realización del proceso de Consulta de Datos:

- Receive:** Es el que recibirá el parámetro de entrada principal, es decir una vez creado el módulo por completo, este necesitará de un parámetro para realizar todo el proceso, mismo que corresponde al número de identificación del usuario, que será enviado por el aplicativo web correspondiente al sistema de asignación de visas.
- Assign:** Esta actividad se encarga de asignar valores a las variables, en ella, existe la posibilidad de aplicar operaciones lógicas, con cadenas, entre otras funcionalidades. Es decir, puede contener una o más operaciones elementales.
- Invoke:** Encargado de invocar los diferentes servicios web.
- Reply:** Retorna el mensaje o la información de respuesta al documento WSDL una vez que se ha cumplido toda la secuencia de operaciones.
- If:** Estructura condicional de tipo Booleana.
- Sequence:** Realiza actividades en orden secuencial, se presenta luego de una estructura condicional.

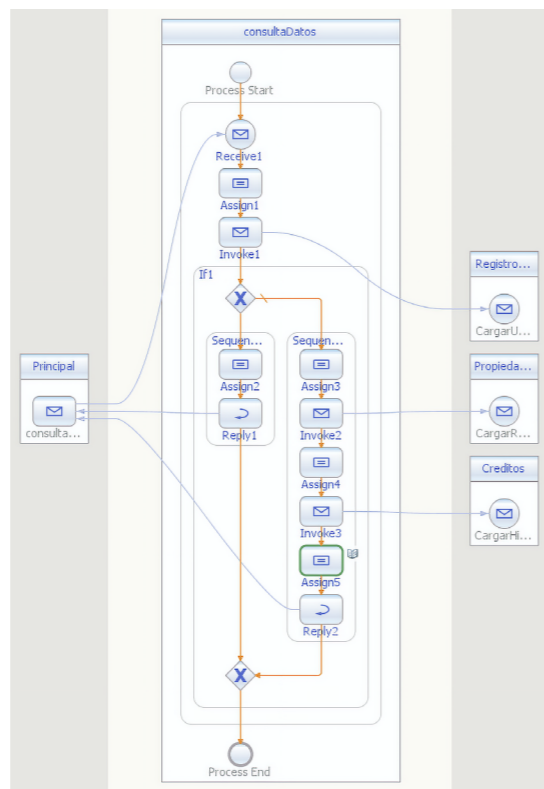


Fig. 2. Proceso Módulo BPEL.

En la Fig. 2. se aprecia la estructura del proceso, en donde se usa la actividad de tipo Receive para obtener el código ingresado por el usuario y almacenarlo, luego existe una actividad Assign, la cual une el parámetro almacenado con el parámetro de entrada del elemento Invoke, este a su vez hace la petición al Registro Civil, la respuesta recibida por Registro Civil, será interpretada con la actividad condicional "if", esta actividad retorna un valor booleano, en caso de ser negativo se asigna la salida a un elemento Reply y este envía el resultado a la respuesta del proceso, terminando de esta manera el proceso de consulta de datos. Caso contrario, empieza la siguiente secuencia, es decir, una vez verificada la existencia del usuario en el Registro Civil, se procede a llamar a los servicios restantes para obtener los datos de bienes y del historial de crédito. Una vez realizadas todas las peticiones correspondientes, se procede a asignar la respuesta del proceso, en este momento se combinan los resultados de los tres servicios web invocados y se los envía a la salida del proceso.

Una vez ejecutado todo el proceso de asignación de variables y validaciones se ha concluido con la creación e implementación del módulo. Cabe mencionar que el BPEL está diseñado para este tipo de escenarios, lo cual hace que los tiempos de implementación sean más cortos [13].

Finalmente, para realizar pruebas de funcionamiento del módulo, no basta con solo crearlo, es necesaria la creación de una Composite Application (aplicación compuesta) en la cual se pueden incluir múltiples módulos BPEL u otros tipos de módulos de Java Business Integration (Integración Empresarial de Java). Una vez agregado el módulo BPEL el cual se ha creado con anterioridad, se procede a agregar un conector, existen varios tipos: SOAP, REST, FTP, HTTP, entre otros. Para este caso particular ha utilizado uno de tipo SOAP. Como paso final, se realiza el despliegue de la Aplicación, el WSDL creado por OpenESB correspondiente al servicio de orquestación se lo puede encontrar en la siguiente ruta: localhost:9080/CompositeVisasService1/ConsultaUsuario?WSDL. Esta ruta, contiene el servidor local, ya que fue en este sitio donde se publicaron, además, contiene un puerto dedicado en este caso 9080. De esta manera el servicio orquestado está listo y a la espera de ser consumido.

F.Integración BPEL con aplicativo web

Cuando se han realizado todos los procedimientos anteriores de implementación y sus respectivas pruebas, es necesaria la fase

de puesta en producción de los servicios web, la orquestación, y en este caso del aplicativo principal, que es el desarrollado para la web. Se realiza entonces la invocación del servicio usando el lenguaje PHP, el mismo que contiene un formulario, en donde es necesario ingresar un campo de entrada: el número de identificación de un ciudadano. Posteriormente, este dato es enviado a través del servicio web creado y antes mencionado, mismo que invocará al recurso de orquestación, que a su vez hará la petición de la información a los servicios correspondientes. Obteniendo finalmente respuesta del mismo, los datos obtenidos son necesarios para la realización de los procesos internos, que verificarán y generarán el reporte final, posteriormente, se procede a guardar la información en la base de datos correspondientes.

III. RESULTADOS Y DISCUSIÓN

El modelamiento de la solución se realizó de manera efectiva utilizando el lenguaje BPEL, ya que permite crear un modelo de empresa multifuncional y adaptable a cualquier tipo de situación; es decir, permite su reutilización, con lo cual se evita crear una aplicación en específico para cumplir ciertas funciones puntuales requeridas por una empresa. La simulación de las entidades del Registro Civil, y Registraduría de Propiedad, plantea un entorno de simulación semejante a la realidad. Varias organizaciones sean estas públicas o privadas, realizan consultas a otras empresas para obtener información sea de personas, productos, etc. Esta información, se encuentra en diversos gestores de bases de datos, utilizando diversos entornos de desarrollo en múltiples lenguajes de programación. En el presente caso, la obtención de la información del solicitante se realiza mediante servicios web creados específicamente para cada entidad, esto facilita su uso dentro del modelo BPEL, ya que una de las ventajas que lo caracteriza como es la integración de servicios ya creados. Una orquestación provee de un ahorro de tiempo a la organización, mediante una sola interfaz, obtiene los datos relevantes acerca del solicitante y a su vez, según los mismos, lo califica apto o no para la obtención de una visa. Además, al estar disponible toda la información en una sola interfaz, se reduce el número de empleados en el proceso, lo cual aumenta la productividad dentro de la empresa, ya que se puede centrar esfuerzos en realizar otras actividades más complicadas, lo cual mejora considerablemente la calidad de servicio.

La utilización de la herramienta Open ESB, potencia el uso de recursos; los componentes son de fácil instalación y manejo. Este

software ofrece una interfaz amigable, su diseño es de gran similitud al IDE NetBeans, y, los aplicativos pueden ser creados, ya sea mediante componentes o código XML puro, evidentemente siguiendo una sintaxis y semántica adecuada. Adicionalmente, brinda una gran ayuda para empezar a desarrollar en BPEL, ya que dispone de una considerable cantidad de herramientas más que suficientes si el modelo a crear no requiere muchas especificaciones, si este es el caso, el uso de herramientas más completas, como puede ser Oracle JDeveloper resultaría más conveniente. El sistema de otorgamiento de visas puede ser reutilizado en otros entornos que lo necesiten. La obtención de información y procesamiento, tanto del Registro Civil, y de la propiedad, pueden ser usados en numerosos procesos, por ejemplo: al solicitar un crédito en cualquier entidad bancaria, en donde es obligatorio conocer las deudas tanto del solicitante, o en caso de que el solicitante esté casado, se requerirá información de su cónyuge, y así determinar si califica o no al crédito [14].

En esta comparativa observamos que el sistema de créditos, es un sistema mucho más complejo, por lo que su desarrollo se podría realizar en la herramienta Oracle JDeveloper, debido a que este es un sistema más completo con respecto al usado en el presente caso de estudio. Se puede destacar las similitudes entre ambos modelos, ya que se sigue una estructura similar, y los pasos necesarios para la orquestación. Indistintamente de la herramienta utilizada, el caso de una aplicación de créditos, posee una construcción más extensa.

Un punto clave a considerar, son las capacidades que tenga el equipo en el cual se realice la orquestación de servicios. En el caso de Open ESB, basta con solo un equipo para el desarrollo de la aplicación en BPEL; en caso que se requiera integrar procesos más extensos, y se requiera el uso de Oracle JDeveloper, es recomendable el uso de varios computadores para equilibrar el uso de recursos. Adicional, como se requiere el uso del gestor de base de datos Oracle XE, es necesario analizar si no se encuentran versiones previas instaladas en el computador, y comprobar que los puertos de escucha se encuentren habilitados.

En estos dos casos, orquestar ayuda a procesar servicios que dispongan de un alto coste de ejecución; generalmente, en grandes empresas, la obtención de información, mediante servicios, tarda desde horas hasta días y en caso de no disponer de una arquitectura orientada a servicios, se dedicaría mucho tiempo a monitorizar los procesos. Para esto, conviene el uso de mensajes dentro del orquestador, mediante los cuales, se informe de manera

rápida y concisa sobre los tiempos que se demora en obtener los datos, tal caso, que se sobrepase los mismos, alertará al usuario [14]. Una complicación que se puede encontrar al usar esta herramienta, es la no compatibilidad con servicios creados en VB.NET, ya que los servicios tipo SOAP que este ofrece están basados en SOAP 1.1 y 1.2.

IV. CONCLUSIONES

La orquestación de servicios al ser un proceso central que toma el control de todos los servicios que se encuentren conectados con un propósito común, brindan beneficios a empresas TI, ya que mejora la efectividad de las mismas.

Las opciones en orquestación que brinda Open ESB, permiten crear un ambiente de desarrollo entendible usando módulos BPEL, además de ofrecer otro tipo de módulos para el manejo de estructuras de información, mismos que pueden ser integrados para trabajar en conjunto, aunque existen ciertas restricciones en comparación con otras herramientas.

Además, como característica opcional, es de fácil instalación y su modo de diseño es entendible, este ofrece un servidor que se encarga de publicar nuestros servicios para su posterior prueba y uso.

El uso de BPEL, al ser un lenguaje ya estandarizado a nivel mundial, ayuda a reducir la complejidad que requiere realizar una orquestación. Las herramientas y componentes usados, están disponibles en cualquier software que permita su implementación. Al disponer ya de un estándar, facilita a las empresas su uso y a su vez reduce costos de desarrollo. El no disponer de un estándar obligaría a cada organización o empresa a crear su propio conjunto de reglas y esto produciría poca colaboración entre diversos servicios web. Para el caso particular de estudio, la simulación de un entorno con diferentes entidades creadas en diferentes lenguajes de programación y usando diferentes gestores de datos, ayuda en gran manera a comprobar los beneficios que conlleva una orquestación, estos se ven reflejados en el tiempo que toma la verificación de datos del solicitante, la reducción de empleados al realizar este proceso, y al ser reutilizable, se pueden automatizar procesos repetitivos.

Para concluir, se puede apreciar, el potencial que brinda a una empresa cambiar a una arquitectura basada en servicios (SOA). Independientemente de la herramienta que se use, el realizar una orquestación brinda nuevos

horizontes, apegados a una mejor calidad de servicio para sus clientes y empleados, siendo esto de vital importancia en un mundo competitivo donde un error, sea este grande o pequeño, puede significar pérdidas económicas o mucho peor, el cierre definitivo de la misma. Un posible trabajo futuro para la arquitectura SOA con el uso de BPEL, sería interesante la implementación de un inventario de funcionalidades con categorizaciones para un desarrollo con asistencia de la inteligencia artificial. La idea central sería el desarrollo de un recomendador que guíen al desarrollador de un sistema en la utilización de componentes de otros sistemas que aporten en el objetivo del sistema final.

RECONOCIMIENTO

Los autores desean agradecer al Vicerrectorado de Investigaciones de la Universidad del Azuay por el apoyo financiero y académico, así como a todo el personal del Laboratorio de Investigación y Desarrollo en Informática (LIDI).

REFERENCIAS

- [1] C. G. Bernardo, "Loan system in brazilian financial institution - A SOA application," Proc. 9th Int. Conf. Inf. Technol. ITNG 2012, pp. 293-298, 2012, doi: 10.1109/ITNG.2012.50.
- [2] Y. Chtouki, H. Harroud, P. O. Box, and A. H. Li, "Service Orchestration Algorithm for Web Services : Evaluation and Analysis," vol. 10, no. 5, pp. 208-218, 2013.
- [3] V. W. Chu, R. K. Wong, S. Fong, and C.-H. Chi, "Emerging Service Orchestration Discovery and Monitoring," IEEE Trans. Serv. Comput., vol. 10, no. 6, pp. 889-901, 2015, doi: 10.1109/tsc.2015.2511000.
- [4] K. Benghazi, M. Noguera, C. Rodríguez-Domínguez, A. B. Pelegrina, and J. L. Garrido, "Real-time web services orchestration and choreography," CEUR Workshop Proc., vol. 601, pp. 142-153, 2010.
- [5] S. Kumari and S. K. Rath, "Performance comparison of SOAP and REST based Web Services for Enterprise Application Integration," 2015 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2015, pp. 1656-1660, 2015, doi: 10.1109/ICACCI.2015.7275851.
- [6] N. Zhou and L. J. Zhang, "Analytic architecture assessment in SOA solution design and its engineering application," 2009 IEEE Int. Conf. Web Serv. ICWS 2009, pp. 807-814, 2009, doi: 10.1109/ICWS.2009.117.

- [7] T. Ploom, A. Glaser, and S. Scheit, "Platform based approach for automation of workflows in a system of systems," c2013 IEEE 7th Int. Symp. Maint. Evol. Serv. Cloud-Based Syst. MESOCA 2013, pp. 12-21, 2013, doi: 10.1109/MESOCA.2013.6632730.
- [8] K. He, "Integration and orchestration of heterogeneous services," 2009 Jt. Conf. Pervasive Comput. JCPC 2009, pp. 467-470, 2009, doi: 10.1109/JCPC.2009.5420139.
- [9] Oracle Corporation, "¿Qué es Java y para qué es necesario?," Oracle Corporation, 2018. https://www.java.com/es/download/faq/whatis_java.xml (accessed Jun. 12, 2020).
- [10] Microsoft, "Herramientas de desarrollo e IDE gratuitos | Visual Studio Community." <https://visualstudio.microsoft.com/es/vs/community/> (accessed Jun. 11, 2020).
- [11] R. Lerdorf, H. Magnusson, P. Olson, and L. Kahwe Smith, "PHP: ¿Qué es PHP? - Manual," [Http://Php.Net/](http://Php.Net/), 2001. <https://www.php.net/manual/es/intro-what-is.php> (accessed Jun. 15, 2020).
- [12] Bootstrap Team, "Download · Bootstrap v4.5." <https://getbootstrap.com/docs/4.5/getting-started/download/> (accessed Jun. 17, 2020).
- [13] OpenESB, "OpenESB Home." <https://www.open-esb.net/> (accessed Jul. 18, 2020).
- [14] M. Orellana and L. M. Arévalo, "Orquestación de servicios web aplicado a una solicitud de crédito comercial utilizando la herramienta Oracle BPEL Process Manager.," pp. 1-137, 2013.

AUTHORS



Jhon Calle

Estudiante de la Carrera de Ingeniería de Sistemas y Telemática de la Universidad del Azuay. Certificado obtenido del VII Congreso Ecuatoriano de Tecnologías de la información y Comunicación – TICEC 2019.



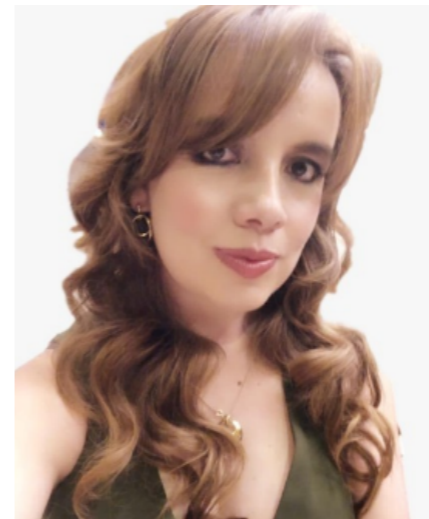
Pablo Lója

Estudiante de la Carrera de Ingeniería de Sistemas y Telemática de la Universidad del Azuay. Certificado obtenido del VII Congreso Ecuatoriano de Tecnologías de la información y Comunicación – TICEC 2019.



Marcos Orellana

Ingeniero en Sistemas de la Universidad del Azuay. Master en Gestión de Sistemas de Información e Inteligencia de Negocios de la Universidad de las Fuerzas Armadas (ESPE) y Master en Docencia Universitaria de la Universidad del Azuay. Candidato a doctor en Ciencias Informáticas en la Universidad Nacional de la Plata Argentina. Docente-Investigador en las líneas de Ciencia de los Datos e Inteligencia Artificial en la Universidad del Azuay. Responsable y Director del Laboratorio de Investigación y Desarrollo en Informática (LIDI).



Priscila Cedillo

Ingeniera de Sistemas por la Universidad de Cuenca, Máster en Ingeniería de Software, Métodos Formales y Sistemas de Información por la Universitat Politècnica de Valencia en España, Magister en Telemática por la Universidad de Cuenca, Ph. D. en Informática por la Universitat Politècnica de Valencia. Estancia de investigación en el National Institute of Informatics en Tokyo y un Posdoctorado en la Universidad Politècnica de Valencia. Autora de aproximadamente 80 artículos indexados. Sus líneas de investigación: Cloud Computing, Calidad de Software, Model Driven Engineering, Internet of Things y Ambient Assisted Living.