DEPARTAMENTO DE INFORMÁTICA Y COMPUTACIÓN

TRABAJO Nº1

PROYECTO SOAP

COMPUTACIÓN PARALELA Y DISTRIBUIDA

1° SEMESTRE-2020

PROFESOR: SEBASTIÁN SALAZAR MOLINA

N° DE GRUPO: 9

RICARDO ABDÓN ALISTE GONZÁLEZ

DANIEL IGNACIO CAJAS ULLOA

RODRIGO ANTONIO CARMONA RAMÍREZ

1/8/2020

INTRODUCCIÓN

El avance tecnológico de los últimos años ha sido tan enorme, que podemos encontrar una gran cantidad de tecnologías, arquitecturas y lenguajes para elegir. Actualmente un gran número de empresas se han decantado por el desarrollo de aplicaciones que puedan trabajar sobre Internet porque permite la distribución global de la información. Cada una proporciona un marco de trabajo basado en la activación de objetos remotos mediante la solicitud de ejecución de servicios de aplicación a un servidor de aplicaciones.

Estas tecnologías han demostrado ser muy efectivas para el establecimiento de sitios Web, sin embargo presentan una serie de desventajas, dado que muchas de ellas son incompatibles entre sí, y tienen dependencia de la máquina servidora que se utiliza, así como del lenguaje de programación empleado. Existen diferentes plataformas que ofrecen soluciones, todas enmarcadas dentro del modelo de componentes, dentro de ellas los "Web Services", que se presenta como una alternativa para facilitar la intercomunicación entre diferentes arquitecturas de componentes y lenguajes, proponiendo una visión de arquitecturas basadas en "servicios" [1].

Esto ha llevado a la necesidad de considerar un nuevo modelo de computación distribuida de objetos que no sea dependiente de plataformas, modelos de desarrollo ni lenguajes de programación. Por todos estos motivos surge el concepto de SOAP (Simple Object Access Protocol) [2].

SOAP no es más que un mecanismo sencillo de expresar la información mediante un modelo de empaquetado de datos modular y una serie de mecanismos de codificación de datos entre dos puntos usando el lenguaje XML. Además, permite que dos sistemas de software se comuniquen independientemente de la plataforma de software y hardware que utilicen los dos equipos participantes [1]. Esto permite que SOAP sea utilizado en un amplio rango de servidores de aplicaciones que trabajen mediante el modelo de comunicación RPC (Remote Procedure Call) [2].

Los principales partes de un mensaje SOAP son los siguientes [2]:

- SOAP Envelope: Es el elemento más importante y de mayor jerarquía dentro del documento XML, y representa al mensaje que lleva almacenado dicho documento.

- SOAP Header: Es un mecanismo genérico que se utiliza para añadir características adicionales al mensaje SOAP. El modo en la que se añadan cada uno de los campos dependerá exclusivamente del servicio implementado entre cliente y servidor, de forma que cliente y servidor deberán estar de acuerdo con la jerarquía con la que se hayan añadido los distintos campos.

- SOAP Body: Es un contenedor de información en el cual se almacenarán los datos que se quieran transmitir de lado a lado de la comunicación (dependiendo del contexto de la solución que ofrece el Servicio Web).

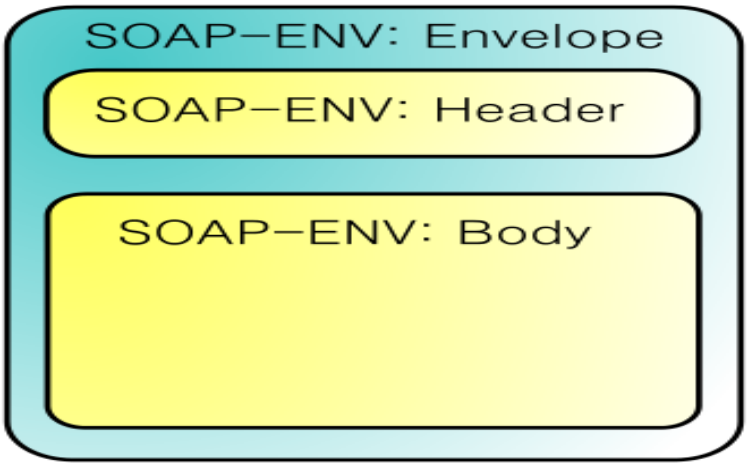

Figura 1: Estructura de un mensaje SOAP

El protocolo facilita esta cooperación de datos, estandarizando el formato y la estructura del mensaje que se pasa entre los equipos. Para visualizar mejor estos conceptos, se presentará un proceso básico del funcionamiento del SOAP mediante la siguiente figura [1]:

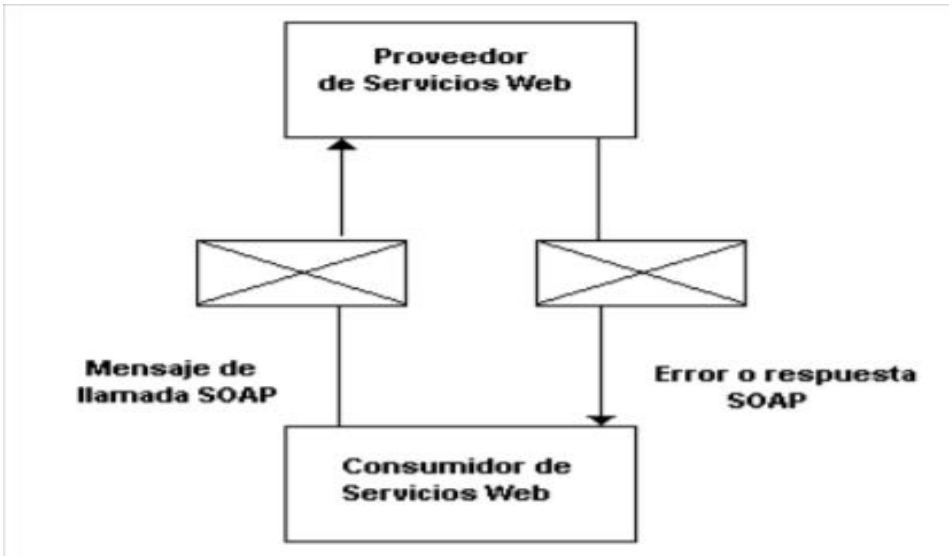

Figura 2: Solicitud / Respuesta de SOAP

En la figura 2, un cliente A solicita un precio de almacén a otro cliente B mediante un mensaje SOAP que define el artículo que interesa al cliente A. El cliente B recibe la solicitud, interpreta el mensaje SOAP y busca la solicitud en su base de datos. El cliente B crea una respuesta de SOAP incluyendo el precio del artículo solicitado y la devuelve al cliente A. El cliente A recibe el mensaje de SOAP, lo interpreta y después lo representa en pantalla para el usuario.

En la siguiente figura se podrá observar dicho flujo entre un servicio Web y una aplicación cliente que consume este servicio.

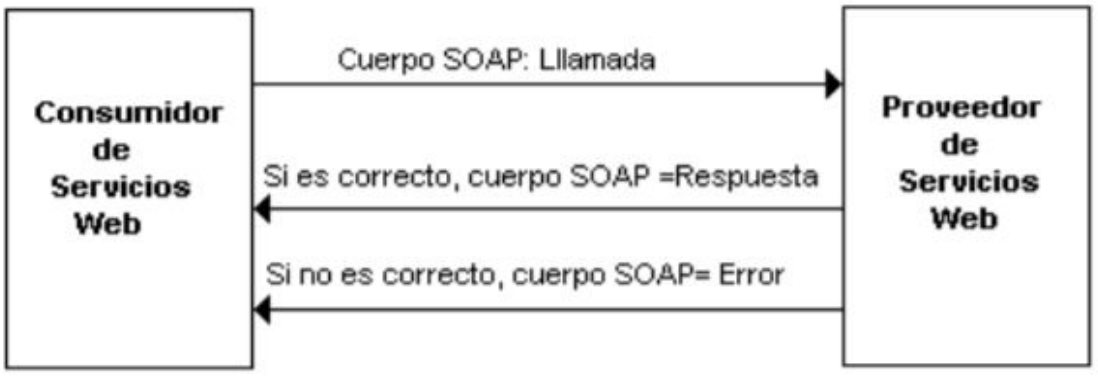

Figura 3: Flujo de los mensajes SOAP

En el caso de que el procedimiento sea realizado de forma correcta, el proveedor ha sido capaz de ejecutar la transacción correspondiente sin problemas, y le entregará una respuesta al cliente. Por el contrario, si el bloque que contiene información relativa a errores, SOAP define un elemento de uso opcional denominado **Fault**, utilizado en los mensajes de respuesta para indicar al cliente algún error ocurrido en el servidor [1].

A la hora de realizar el diseño de SOAP se han tenido en cuenta una serie de consideraciones con el fin de cumplir una serie de objetivos claros, los cuales se muestran a continuación [2]:

- Establecer un protocolo estándar de invocación a servicios remotos que esté basado en protocolos estándares de uso frecuente en Internet, como son HTTP (Hypertext Transport Protocol) para la transmisión y XML (Extensible Markup Language) para la codificación de los datos.

- Independencia de plataforma hardware, lenguaje de programación e implementación del servicio Web.

Con todos los apuntes anteriores, se tiene una base de conocimientos que es posible utilizar para implementación de un prototipo SOAP, utilizando las "bondades" de los "Web Services".

RESOLUCIÓN DEL PROBLEMA

El proyecto SOAP será desarrollado en el lenguaje de programación Python 3 (de la versión 3.8 en adelante). Las razones de su utilización es por la rápida ejecución de dicho lenguaje en la máquina virtual, con un rendimiento bastante bueno. Además, es muy legible, dado que la sintaxis de Python es muy ordenada, muy parecida a la descripción de un pseudocodigo, y permite la escritura de programas cuya lectura resulta fácil de comprender en un corto periodo de tiempo [4]. Su forma de manipular y extraer datos de archivos como el de este trabajo también es sencillo de aplicar, más que otros lenguajes como C, Java, Dart, etc. En resumen, Python permite llevar a cabo buenas prácticas de programación.

La razón más importante de elegir este lenguaje por sobre otros lenguajes es porque se ha utilizado con anterioridad en anteriores proyectos, incluso pequeños programas donde se ha trabajado en la manipulación de archivos Excel, generando muy buenas sensaciones.

```
def ordenar(carrera): ###Funcion encargada de ordenar el listado de alumnos de una carrera; utiliza el metodo Quicksort; el parametro que recive es una lista
    tope=len(carrera)
    izquierda=[]
    derecha=[]
    centro=[]
    if(tope>1): ###Corrobora caso en el que solo ahi un elemento; en caso de tener mas de 1 elemento
        pivote=carrera[0][1] ###Define el pivote para separar en elementos mayores, menores o iguales a este
        for i in carrera:
            if(i[1]>pivote):    ###Si es mayor, va a una lista llamada "Izquierda"
                izquierda.append(i)
            elif(i[1]==pivote): ###Si es igual, va a una lista llamada "Centro"
                centro.append(i)
            elif(i[1]<pivote):  ###Si es menor, va a una lista llamada "Derecha"
                derecha.append(i)
        return ordenar(izquierda)+centro+ordenar(derecha) ###Finalmente, regresa la union de las 3 listas, pero aplicandole esta misma funcion a izquierda y derecha
    else:
        return carrera ###En el caso de solo ser 1 elemento, se devuelve directamente el arreglo, ya que no ahi nada que ordenar
```

Figura 4: Ejemplo de un código ordenado en Python

En este programa se requiere obligadamente instalar o descargar de internet las siguientes librerías para crear el servicio de SOAP:

- Spyne: Es un juego de herramientas de llamadas a procedimiento remoto (***Remote Procedure Call,*** RPC) de Python que facilita la exposición de los servicios en línea que tienen una API bien definida utilizando múltiples protocolos [5]. Es bastante importante para el consumo del SOAP durante su ejecución.

- Openpyxl: Es una librería de Python que se utiliza para tratar con varios archivos de Excel (xlsx/xlsm/xltx/xltm). Se utiliza para realizar tareas como leer o escribir datos en el archivo, dibujar algunos gráficos, acceder a la hoja, modificar (añadir y eliminar) en la hoja de Excel, formatear, estilizar en la hoja, entre otras tareas [6].

- Base64: Este módulo permite realizar conversiones de bytes a valores que caben dentro del rango de caracteres imprimibles ASCII [7]. Es necesario para llevar a cabo las instrucciones de codificación solicitadas en el servicio del SOAP.

- Lxml: Es un conjunto de herramientas API que permite un fácil manejo de los archivos XML y HTML, extremadamente rápida al analizar documentos de gran tamaño, y proporciona una fácil conversión de datos a tipos de datos de Python [8].

- Re: Este módulo se utiliza para la manipulación de Unicodes (strings) y cadenas de 8 bits (bytes) [9].

- Mimetypes: Permite realizar conversiones entre un nombre de archivo (o URL) y el tipo MIME asociado a la extensión del nombre de archivo. Se proporcionan conversiones de nombre de archivo a tipo MIME y de tipo MIME a extensión de nombre de archivo [10].

Cabe destacar que Python es un lenguaje orientado a objetos, por lo que puede emplear los materiales y las herramientas para diseñar y programar un sistema que otorga este paradigma, o sea, las características más importantes de los modelos de abstracción del mundo real, utilizando clases y objetos en el procesamiento de datos de entrada para obtener otros de salida, como el problema a resolver en este informe.

La construcción del SOAP fue basado en la siguiente estructura de desarrollo (sujeto a corrección):

1. Probar el lenguaje escogido y testear su rendimiento con aplicaciones desarrolladas con anterioridad o extraídas de código libre en la web, ya que es importante hacer pruebas con Python para saber si la ejecución de nuestro programa va a ser rápido y seguro.

2. Investigar las librerías necesarias para trabajar en la construcción del servicio SOAP con Python (Spyne, Openpyxl, etc), ya que en el programa se realizan conversiones a texto plano, hay que extraer datos del archivo csv, entre otras tareas.

3. Construir un algoritmo de prueba para llenar el archivo final (xlsx) con todos los requisitos de presentación, dado que se han realizado trabajos o funciones creadas con anterioridad en el manejo de Excel con este lenguaje de programación.

4. Se planificó una estrategia para manejar los datos extraídos de cada estudiante del archivo csv para determinar sus resultados dependiendo de la carrera. Una de las ventajas de Python es que permite reducir las líneas de código necesarias para este algoritmo, donde es posible calcular las ponderaciones de varias carreras al mismo tiempo.

Calculo Ponderaciones de Ing. en Computación mención Informática

Rut	Nem	Ranking	Matemáticas	Lenguaje	Ciencias	Historia
18397824	567	628	695	625	674	678
18870513	592	643	702	604	667	562
19526381	645	684	705	515	700	567

Calculo de ponderación del alumno: Nem * 0,10 + Ranking * 0,25 + Matemáticas * 0,35 + Lenguaje * 0,20 + Ciencias o Historia * 0,10

Ponderación Rut 18397824 = 567 * 0,10 + 628 * 0,25 + 695 * 0,35 + 625 * 0,2 + 678 * 0,1
Ponderación Rut 18397824 = 649,75

Ponderación Rut 18870513 = 592 * 0,10 + 643 * 0,25 + 702 * 0,35 + 604 * 0,2 + 667 * 0,1
Ponderación Rut 18870513 = 653,15

Ponderación Rut 19526381 = 645 * 0,10 + 684 * 0,25 + 705 * 0,35 + 515 * 0,2 + 700 * 0,1
Ponderación Rut 19526381 = 655,25

Figura 5: Ejemplo ilustrativo del proceso de cálculo que realiza el programa en Python

5. Para establecer las mejores opciones que tiene el alumno en la elección de la carrera, se ha decidido crear una lista de toda la población de estudiantes registradas en el archivo csv, la cual contiene el rut del estudiante, un índice para la carrera a postular y su ponderación. Dependiendo de la cantidad de personas registradas, se elige la carrera donde más le conviene en términos del puntaje obtenido (el resultado más alto).

6. Después, se realizó una pequeña investigación sobre cuál es el mejor algoritmo de ordenamiento para los estudiantes seleccionados en el proceso de admisión para todas las carreras. Finalmente, el método Quicksort fue el escogido, por la rapidez del mismo durante la ejecución del código, en desmedro de otros algoritmos como el ordenamiento burbuja por su lento rendimiento al trabajar con grandes volúmenes de datos.

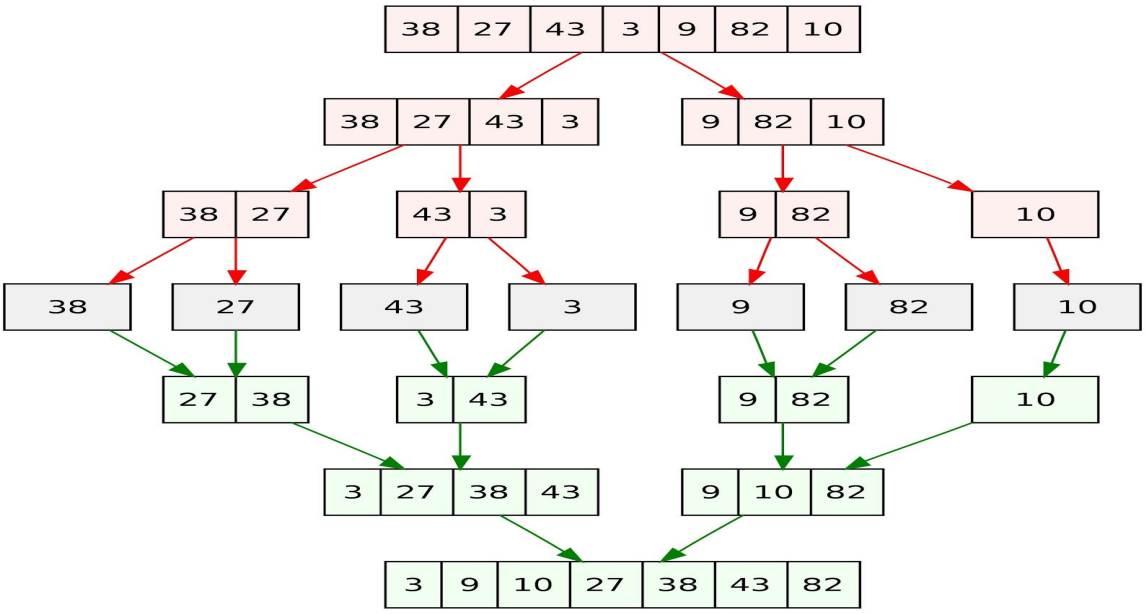

Figura 6: Ejemplo ilustrativo de cómo funciona el ordenamiento Quicksort

7. Construcción del algoritmo de ordenamiento de este programa aplicando el método Quicksort (Figura 4).

8. El siguiente paso fue escribir el código de selección de los alumnos para que estos sean inscritos en la carrera que más le conviene por el puntaje obtenido. Este algoritmo llevará a cabo varias iteraciones con ese objetivo en concreto, donde puede agregar y eliminar estudiantes registrados en una carrera en particular. Una vez que ingresa al postulante, el programa busca cual es la carrera donde tiene la ponderación más alta y trata de registrarlo en el sistema. Si obtiene el resultado suficiente para ingresar a dicha carrera, queda inscrito de forma automática. En caso que no tener la ponderación, se decidió crear una especie de lista de espera, con el fin de hacer un reubicamiento en el proceso de admisión.

9. Se emplea el algoritmo de ordenamiento Quicksort, que organiza los puntajes de manera decreciente, es decir, del puntaje más alto al estudiante con el resultado

más bajo registrado en la carrera correspondiente. Este código ejecuta varias iteraciones donde los alumnos con los mejores puntajes son ingresados en la carrera más adecuada (por las razones expuestas anteriormente). En caso de tener un curso con todos los cupos ocupados, el postulante con el puntaje más bajo queda en la lista de reubicamiento para comprobar si hay un estudiante con mejor puntaje, y en caso de que se cumpla dicha condición, se elimina el registro de datos de ese estudiante de la carrera en cuestión.

Puntajes de los estudiantes seleccionados en Ing. Comercial

17061567	18903826	16790313	16563673	19245708	19563662	18924067	19735596
630	578	701	566	659	644	593	625

Índice	Rut	Puntaje
1	16790313	701
2	19245708	659
3	19563662	644
4	17061567	630
5	19735596	625
6	18924067	593
7	18903826	578
8	16563673	566

Figura 7: Ejemplo ilustrativo de lo que realizará el algoritmo de ordenamiento y el formato de datos que tendrá la hoja de Excel con los puntajes de los estudiantes de cada carrera

10. Implementar/crear el algoritmo adecuado para llenar los datos de los estudiantes ingresados en cada una de las carreras (separadas por hoja en el archivo xlsx) disponibles en la universidad. Posteriormente, se agregó una condición que permite identificar la ponderación del alumno dependiendo del índice de la carrera con la cual se ingresan los campos en el archivo Excel (para más información, revisar la documentación del código fuente del programa).

Registro de los estudiantes en hojas de Excel

Input: Lista de listas de estudiantes seleccionados

Output: Ninguno (generación de Excel con seleccionados)

```
01|Creacion del Excel
02|Se comienza iteración por cada listado de alumnos:
03|     Se obtiene el numero de la carrera correspondiente
04|     Creación de la hoja que le corresponde
05|     Se establecen los nombres de las columnas
06|     Se comienza iteración por cada alumno del listado:
07|          Ingreso de índice, RUT y puntaje ponderado del alumno
08|Se procede a asignarle nombre al Excel y guardar el documento
```

Figura 8: Pseudocódigo de los registros de los estudiantes en el archivo Excel

11. Desarrollar un código para presentar el tipo mime en el SOAP (tanto recibido, como el que se debe devolver al consumidor de la API); esto mediante la validación usando las extensiones y el string base64.

12. Hacer pruebas y testeos del código escrito, verificando si muestra en pantalla los resultados esperados, corrigiendo los errores que se presenten durante la ejecución del programa (hipotéticamente hablando), para que así el servicio esté funcionando sin mayores problemas.

13. Realizar la documentación correspondiente de la versión definitiva del código escrito en el archivo del lenguaje Python.

Con estos pasos anteriormente mencionados, se logró construir un procedimiento que ha permitido generar el archivo solicitado utilizando el software SoapUI, el cual es una herramienta diseñada en la prueba y el desarrollo de aplicaciones de tipo webservice en arquitecturas orientadas a servicios (SOA) y Representational State Transfers (REST).

Si la ejecución es llevada a cabo sin problemas, el programa va a generar un archivo xslx en la carpeta donde se encuentra guardada el archivo de Python, tal como se puede apreciar en la siguiente imagen:

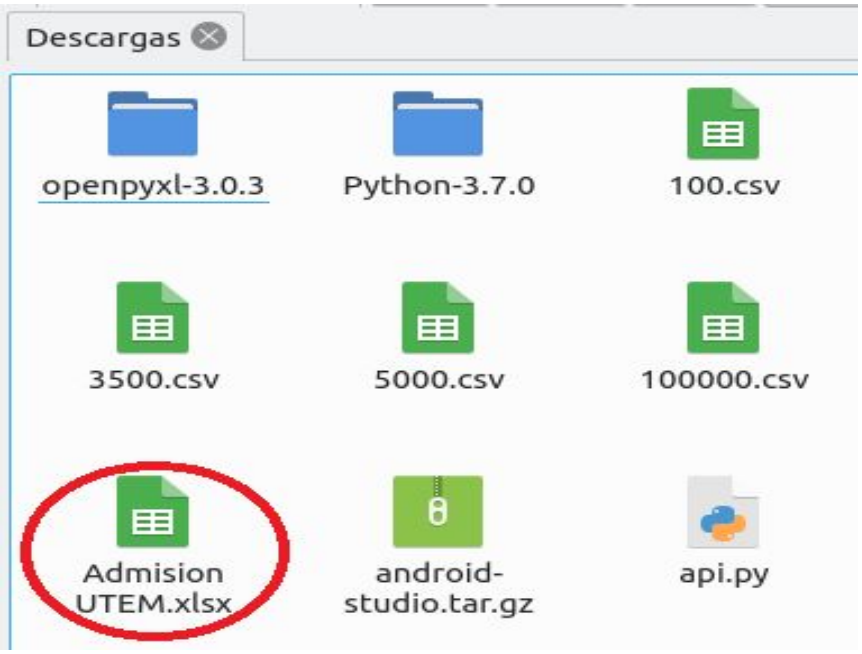

Figura 9: El archivo queda almacenado en la carpeta Descargas de la Máquina Virtual

CONCLUSIÓN

Se puede definir que SOAP es un recurso bastante útil en la creación de programas o aplicaciones con un enfoque distribuido, como el caso trabajado en este proyecto. Cabe destacar algunas de las ventajas que proporciona al programador, ya que es capaz de realizar operaciones con pocos recursos, también sobre múltiples instancias del proceso

que comparten la misma operación, son independientes del sistema operativo, proporciona un protocolo de comunicación bastante robusto, etc.

Gracias a la utilización de un lenguaje como Python, se logró desarrollar un programa más optimizado de lo esperado, ahorrando enormes cantidades de líneas de código en el proceso. Además, la sintaxis ha permitido crear algoritmos bien ordenados, evitando malas prácticas de programación que no hubiera sido posible en otros lenguajes tipo C o C++.

Es posible deducir que el servicio desarrollado tiene muchas similitudes con un programa RPC, donde un nodo de red (el cliente) envía un mensaje de solicitud a otro nodo (el servidor) y este último inmediatamente le responde con un mensaje al cliente [3], incluso se puede afirmar que es precisamente un RPC como cualquier otro.

Uno de los objetivos primordiales era que el servicio tenga las capacidades de trabajar a buen ritmo cuando es necesario procesar un gran volumen de datos, un ejemplo de esto serían las enormes cantidades de transacciones que se realizan diariamente en el sitio web de un banco, cabe destacar que dependiendo que las capacidades tanto del software como del hardware pueden influir en el rendimiento del servicio, es decir, opera bajo ciertas condiciones.

BIBLIOGRAFÍA

[1] Á. Reyes, «http://cybertesis.uach.cl/,» 9 Abril 2004. [En línea]. Available: http://cybertesis.uach.cl/tesis/uach/2004/bmfcir457i/doc/bmfcir457i.pdf. [Último acceso: 12 Junio 2020].

[2] A. Oliva, «http://bibing.us.es/,» Abril 2006. [En línea]. [Último acceso: 12 Junio 2020].

[3] J. Acevedo, «http://repositorio.uchile.cl/,» Enero 2007. [En línea]. Available: http://repositorio.uchile.cl/bitstream/handle/2250/104492/acevedo_j.pdf?sequence=3&isAllowed=y. [Último acceso: 12 Junio 2020].

[4] A. Marzal, I. Gracia y P. García, «http://mmc.geofisica.unam.mx/,» 2014. [En línea]. Available: http://mmc.geofisica.unam.mx/femp/Herramientas/Lenguajes/Python/s93.pdf. [Último acceso: 12 Junio 2020].

[5] Spyne, «spyne.io,» [En línea]. Available: http://spyne.io/#inprot=HttpRpc&outprot=JsonDocument&s=rpc&tpt=WsgiApplication&validator=true. [Último acceso: 15 Junio 2020].

[6] Sonoo Jaiswal, «https://www.javatpoint.com/,» [En línea]. Available: https://www.javatpoint.com/python-openpyxl. [Último acceso: 15 Junio 2020].

[7] A. Amin, «https://stackabuse.com/,» 11 Diciembre 2019. [En línea]. Available: https://stackabuse.com/encoding-and-decoding-base64-strings-in-python/. [Último acceso: 15 Junio 2020].

[8] Stack Abuse, «https://stackabuse.com/,» 10 Abril 2019. [En línea]. Available: https://stackabuse.com/introduction-to-the-python-lxml-library/. [Último acceso: 17 Junio 2020].

[9] Python, «https://docs.python.org/,» [En línea]. Available: https://docs.python.org/3/library/re.html. [Último acceso: 17 Junio 2020].

[10] Python, «https://docs.python.org/,» [En línea]. Available: https://docs.python.org/2/library/mimetypes.html. [Último acceso: 17 Junio 2020].