



República Bolivariana De Venezuela
Universidad Nacional Experimental De Guayana
Vicerrectorado Académico
Coordinación General De Pregrado
Proyecto de carrera: Ingeniería en informática
Unidad curricular: Sistemas distribuidos

Evaluación I. Parte II

Profesora:
Virginia Padilla

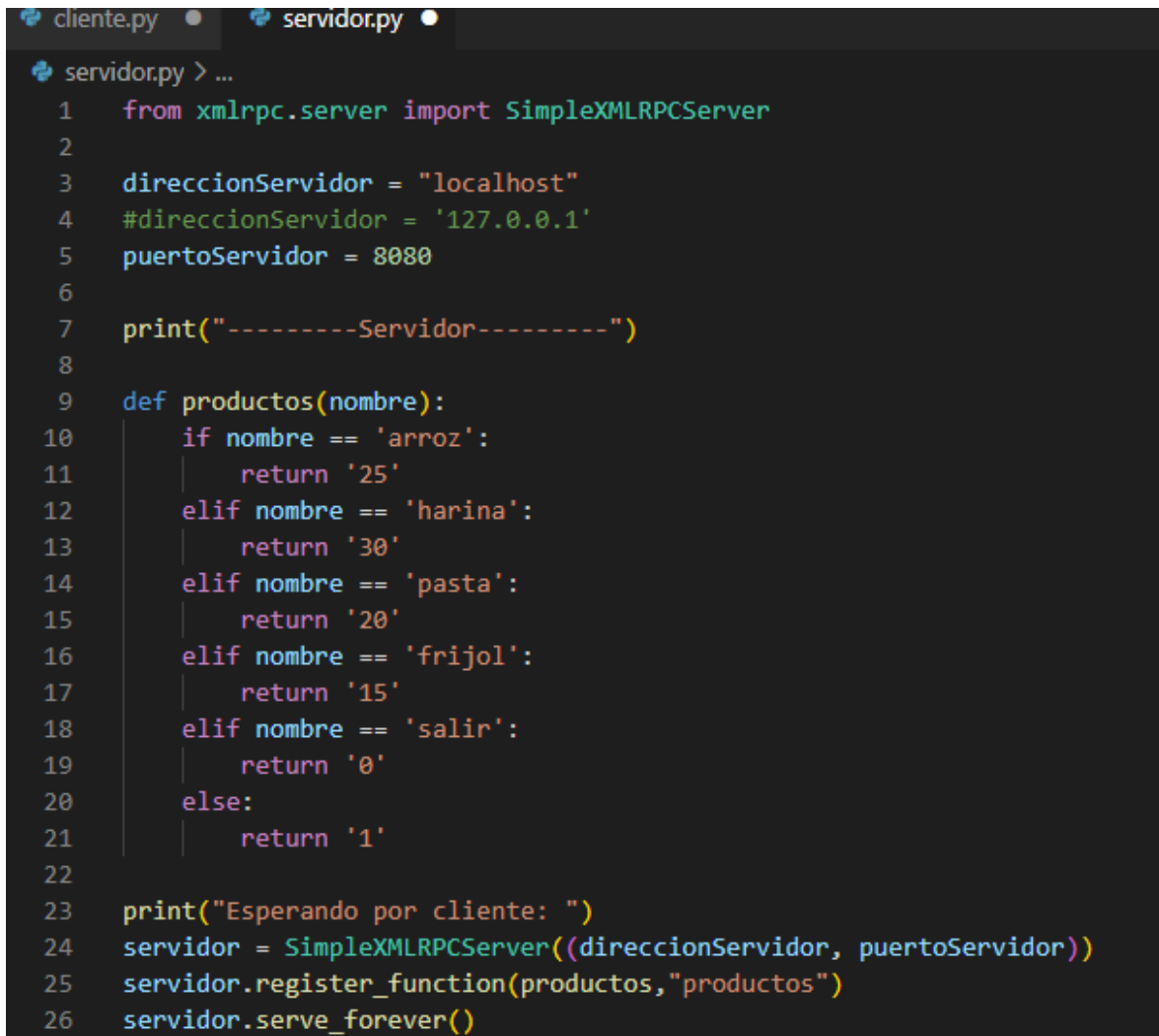
Bachiller:
Geruby Bermúdez C.I: 26.264.179

Ciudad Guayana, julio 2022

Llamada a procedimiento remoto (RPC) es una técnica que utiliza el modelo cliente-servidor para ejecutar tareas en un proceso diferente como podría ser en una computadora remota.

La aplicación implementada sigue la estructura cliente-servidor, donde el cliente realizara consultas al servidor sobre los precios de ciertos productos en específico. Esta aplicación está elaborada con Python.

Estructura del servidor.py



```
servidor.py > ...
1  from xmlrpc.server import SimpleXMLRPCServer
2
3  direccionServidor = "localhost"
4  #direccionServidor = '127.0.0.1'
5  puertoServidor = 8080
6
7  print("-----Servidor-----")
8
9  def productos(nombre):
10     if nombre == 'arroz':
11         return '25'
12     elif nombre == 'harina':
13         return '30'
14     elif nombre == 'pasta':
15         return '20'
16     elif nombre == 'frijol':
17         return '15'
18     elif nombre == 'salir':
19         return '0'
20     else:
21         return '1'
22
23  print("Esperando por cliente: ")
24  servidor = SimpleXMLRPCServer((direccionServidor, puertoServidor))
25  servidor.register_function(productos, "productos")
26  servidor.serve_forever()
```

Figura 1. Servidor.py

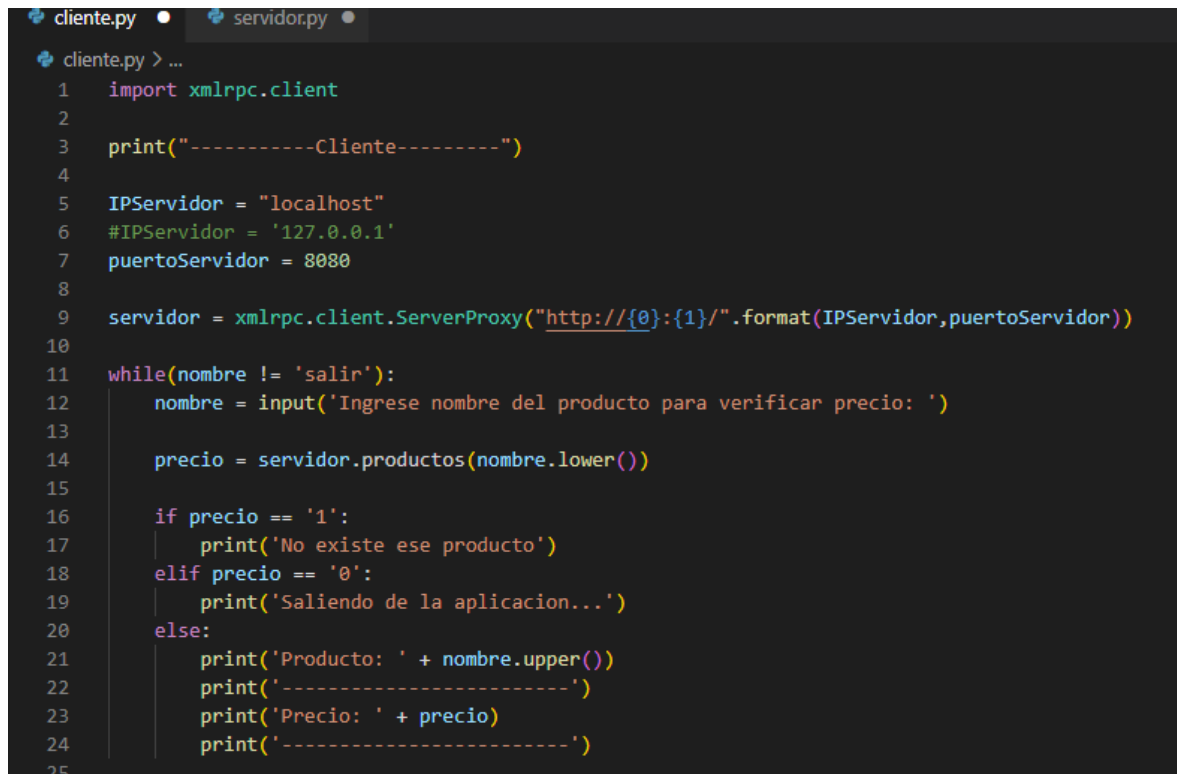
xmlrpc.server proporciona un marco de servidor básico para servidores XML-RPC escritos en Python. Los servidores pueden ser independientes, utilizando SimpleXMLRPCServer.

Para la conexión de un cliente a un servidor se instanció un objeto **ServerProxy** dándole el URI del servidor. En este caso el servidor se ejecuta en el puerto 8080 de la dirección IP 127.0.0.1 (localhost).

Como el servidor contendrá los productos con sus respectivos precios a los cuales el cliente podrá acceder, se definió una función **productos**.

SimpleXMLRPCServer contiene clases para crear un servidor multiplataforma e independiente del lenguaje utilizando el protocolo XML-RPC. De esta forma se crea el **servidor**, pasándole como parámetro el puerto y la dirección IP. Dentro del servidor se registra la función productos antes creada, la cual podrá recibir como parámetro, el nombre del producto a buscar. Con **serve_forver()** el objeto servidor puede procesar una o muchas solicitudes.

Estructura del cliente.py



```
cliente.py • servidor.py •
cliente.py > ...
1  import xmlrpc.client
2
3  print("-----Cliente-----")
4
5  IPServidor = "localhost"
6  #IPServidor = '127.0.0.1'
7  puertoServidor = 8080
8
9  servidor = xmlrpc.client.ServerProxy("http://{0}:{1}/".format(IPServidor,puertoServidor))
10
11 while(nombre != 'salir'):
12     nombre = input('Ingrese nombre del producto para verificar precio: ')
13
14     precio = servidor.productos(nombre.lower())
15
16     if precio == '1':
17         print('No existe ese producto')
18     elif precio == '0':
19         print('Saliendo de la aplicacion...')
20     else:
21         print('Producto: ' + nombre.upper())
22         print('-----')
23         print('Precio: ' + precio)
24         print('-----')
25
```

Figura 2. Cliente.py

El **lower()** es un método convierte todos los caracteres en mayúsculas de una cadena en caracteres en minúsculas y los devuelve.

El **upper()** es un método convierte todos los caracteres en minúsculas de una cadena en caracteres en mayúsculas y los devuelve.

xmlrpc.client es un método de RCP que utiliza XML pasado a través de HTTP(S) como transporte. Este método permite que un cliente puede llamar a métodos con parámetros en un servidor remoto y recuperar datos estructurados.

xmlrpc.client.ServerProxy() permite crear una instancia de un objeto que gestiona la comunicación con el servidor XML-RPC remoto. El primer argumento es una instancia de fábrica de transporte (es una instancia interna para https: URL), el segundo argumento requerido es un URI (el cual es la URL del servidor).

Una vez establecida la conexión con el servidor, el cliente puede ingresar el nombre del producto del cual desea verificar el precio, y con la función ***productos***, se realiza la búsqueda dentro del servidor. La respuesta a esta búsqueda se almacena en la variable ***precio***.

Funcionamiento de la llamada a procedimiento remoto en la aplicación

1. El cliente hace la llamada al procedimiento remoto mediante un mensaje a través de la red. Este se detiene (ya que es un proceso síncrono) para esperar la respuesta del servidor y así poder continuar su ejecución.
2. El servidor recibe la petición y desempaqueta el mensaje para extraer la información necesaria para realizar la tarea.
3. El servidor ejecuta la tarea (busca la información del producto que solicito el cliente).
4. El servidor crea un mensaje de respuesta para el cliente en el que incluye el resultado de la tarea que este le pidió realizar.
5. El cliente recibe y desempaqueta el mensaje de respuesta del servidor el cual contiene el nombre y precio del producto solicitado.
6. El cliente repite el proceso hasta que ingrese la palabra '*salir*' y de esta manera se termina la conexión entre cliente-servidor.

La Figura 3 representa el proceso de interacción entre cliente-servidor en la aplicación de consulta de productos.

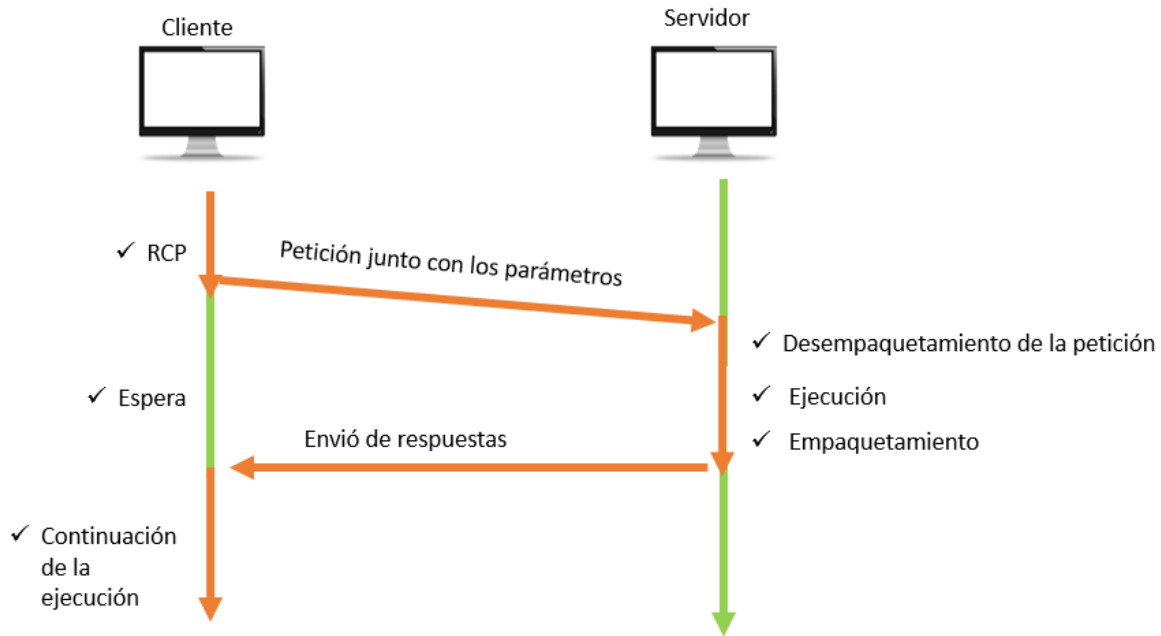


Figura 3. Gráfico cliente-servidor

Interacción Cliente-Servidor en la aplicación

En la siguiente imagen se puede apreciar que, por cada consulta realizada por el cliente, el servidor proporciona una respuesta, hasta que el cliente ingresa la palabra 'salir' y se termina la interacción.

```

Administrador: C:\Windows\system32\cmd.exe - py servidor.py
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\vict 7\Desktop\Examen1_parte2>cd C:\Users\vict 7\Desktop\Examen1_parte2
C:\Users\vict 7\Desktop\Examen1_parte2>py servidor.py
Servidor
Esperando por cliente:
127.0.0.1 - - [27/Jul/2022 22:32:52] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [27/Jul/2022 22:33:03] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [27/Jul/2022 22:33:13] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [27/Jul/2022 22:34:17] "POST / HTTP/1.1" 200 -

Administrador: C:\Windows\system32\cmd.exe
que el equipo de destino denegó expresamente dicha conexión
C:\Users\vict 7\Desktop\Examen1_parte2>py cliente.py
Cliente
Ingrese nombre del producto para verificar precio: arroz
Producto: ARROZ
Precio: 25
Ingrese nombre del producto para verificar precio: Harina
Producto: HARINA
Precio: 30
Ingrese nombre del producto para verificar precio: Pasta
Producto: PASTA
Precio: 20
Ingrese nombre del producto para verificar precio: salir
Saliendo de la aplicación...
C:\Users\vict 7\Desktop\Examen1_parte2>_
  
```

Figura 4. Interacción Cliente-Servidor