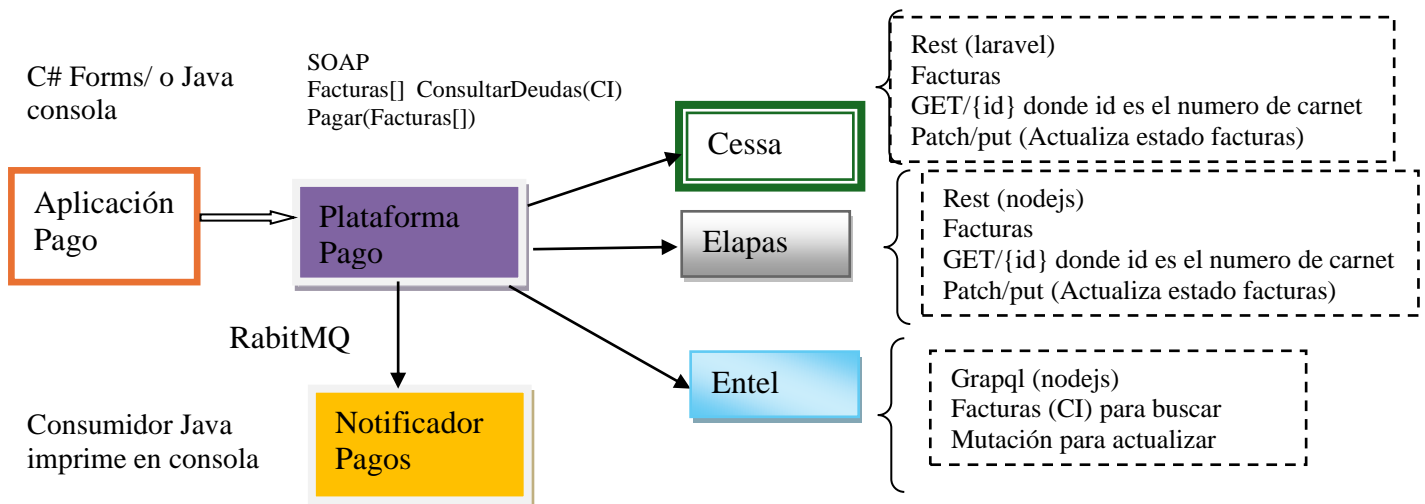


1. Realizar un sistema utilizando la tecnología Soquets tcp multiusuario para , que cliente envíe un número y este valor se vaya sumando en el servidor y le devuelva la suma acumulada hasta que el cliente introduce 0 y le muestra un mensaje con la cantidad de veces que se realizó la acumulación. (10 puntos)

Ej	
Cliente	Servidor
2	acumulado 2
5	acumulado 7
9	acumulado 16
0	cantidad veces 3, acumulado 16

2. Realizar un sistema de pago de servicios básicos en base al siguiente esquema:



Aplicación Pago Cliente escrito en java o c# que se comunica con la plataforma de pago a sus métodos Facturas[] ConsultarDeudas(CI) y Pagar(Facturas[])

- Factura es una clase que tiene los campos :Empresa, NroFactura, CI, NombreCompleto, Periodo, Monto, Estado (Pendiente, Pagado)
- Debe tener la opción de llamar a estos métodos

Plataforma Pago Escrito en c# ,

- El método consultarDeudas llama a los tres servicios y genera un solo arreglo de facturas con las deudas del cliente
- El método pagar recorre por el arreglo de facturas recibido y llama a la empresa que corresponde para realizar el pago y cambiar el estado a pagado
- Luego de esto manda a una cola en rabbitmq la operación solicitada y los valores en un json

Cessa

- Escrito en laravel
- Tiene conexión a su bd a la tabla facturas
- Método get/{id} para buscar facturas de una persona donde el id es el carnet
- Método put/{id} para pagar la factura donde id es el numero de factura

Buena Suerte!

Elapas

- Escrito en nodejs
- Tiene conexión a su bd a la tabla facturas
- Método get/{id} para buscar facturas de una persona donde el id es el carnet
- Método put/{id} para pagar la factura donde id es el numero de factura

Entel

- Escrito en nodejs
- Tiene conexión a su bd a la tabla facturas
- Permite buscar por ci en grapql
- Permite actualizar mediante el número de factura a través de una mutación

Notificador Pago

- Cliente Java que lee desde la cola e imprime en pantalla la información enviada desde la plataforma de pago