

1. Realizar el siguiente desarrollo dado los siguientes 2 casos de uso.
 - a. Acreditar y debitar a una cuenta de cliente. Se debe considerar lo siguiente:
 - I. Se debe solicitar el numero de cliente, numero de cuenta y el monto a acreditar/debitar, la fecha se asume que el servicio la pondrá por defecto.
 - II. Si se trata de un debito se debe mandar el valor en negativo. Ejemplo: -125.00
 - III. Si la cuenta no existe al menos debe existir el cliente y se debe considerar en automático el registro de la cuenta y su monto.
 - IV. La respuesta que se enviara solo debe de contener un mensaje similar a este “Su transacción esta en proceso” o “Transacciones recibidos”, luego de esa respuesta en un método asíncrono enviar la respuesta de la transacción de la base de datos enviarlo por una cola de RabbitMQ o tópico de Kafka.
 - b. Consultar el histórico de transacciones realizadas a una cuenta. Se debe considerar lo siguiente:
 - I. Se permitirá filtrar por el numero de cuenta y numero de cliente, pero si se omite el número de cuenta deberá mostrar todas las cuentas del cliente.
 - II. Considerar que al mostrar las transacciones de la cuenta se debe mostrar la información completa del cliente y su saldo actual.
2. Adicional se agregan los siguientes requerimientos de diseño y codificación.
 - Define las estructuras de entrada/salida de las operaciones planteadas en los casos de uso respectivos.
 - Utilice el patrón de diseño DAO, DTO para las capas de datos, servicio y encapsulamiento de datos.
 - Utilice el principio SOLID “Single Responsibility” para el manejo de las funcionalidades antes descritas.
 - Utilice el principio SOLID “Dependency Inversion” para el manejo de instancias de objetos eficiente.
 - Establece en la respuesta parámetros de código y descripción de los casos exitosos y/o fallidos. Utilice las buenas practicas para el establecimiento de códigos de éxito y error.
 - Implementar el manejo controlado de excepciones y uso de correcto de logs (niveles de logs en cada componente interno).
 - Desarrollar 2 pruebas unitaria. Utilice el framework apropiado para el desarrollo de las siguientes pruebas unitarias:
 - ✓ Comprobar la conexión a la base de datos.
 - ✓ Consulta de las transacciones de la cuenta, información del cliente y saldo (prueba basada en literal “b”)