

Evaluación Desarrollador NodeJS:

1. Diagrama Entidad-Relación, se requiere llevar el manejo y registro de la siguiente información de transacciones bancarias
 1. Clientes: se desea llevar registro de la información básica de un cliente: **nombre completo, DPI, fecha de nacimiento, sexo, país nacimiento.**
 2. Producto: a los clientes se les ofrece dos tipos de productos cuentas de ahorros o monetarias y para cada producto se debe tener: **el nombre del producto, la tasa de interés, tipo de producto (Ahorros o monetarios) y Calculo de interés (Diario o Mensual).**
 3. Cuenta: se debe llevar el control de las cuentas de los clientes tomando en cuenta la siguiente información: **número de cuenta, el producto al que está asociada la cuenta, el cliente al que pertenece la cuenta, estatus (Activa o Inactiva), y el saldo de cuenta.**
 4. Códigos de operación: es un catálogo que sirve para identificar los tipos de transacciones que se puede realizar la información que debe manejar es: **Código de operación, descripción, tipo de operación (Crédito o Débito).**
 5. Transacciones Cuenta: se debe llevar el registro de las transacciones de crédito y débito de las cuentas teniendo en cuenta la siguiente información: **ID único de la transacción, cuenta que está asociada a la transacción, Código de operación asociado a la transacción, monto, fecha de la transacción y código de operación.**

Realice el diagrama entidad relación con la información anterior, y agregue todos los atributos de control que crea necesarios.

2. Desarrollo de APIs: para el desarrollo de los APIS debe crear su esquema de datos en una base de datos MySQL y toda la información que necesite que contengan las tabla para que funcione el API la debe agregar manualmente no debe realizar ningún mantenimiento de datos.

- API para registro de transacciones (**Obligatorio**): debe desarrollar un api que tendrá como parámetros de entrada los siguientes datos: numero de la cuenta, fecha de la transacción, monto, código de operación (Débito o Crédito).

El API debe registrar la transacción tomando en cuenta que la cuenta tiene que estar activa, tener lo fondos suficientes para las transacciones de débito, debe actualizar el saldo de la cuenta y debe retornar un JSON con los siguientes atributos

- `codigo_respuesta`: 0 si todo estuvo bien, caso contrario un código de error.

- descripcion_respuesta la palabra OK si todo estuvo bien, caso contrario descripción del error.
- id_transaccion El ID único de la transacción generada si es que no hubo error.
- APIs para consultas:
 - Api para obtener listado de saldos de cuentas con la siguiente información **(Opcional)**,
 - número de cuenta,
 - nombre del cliente al que pertenece la cuenta,
 - tipo de cuenta (Monetario o Ahorro),
 - Tasa de interés de la cuenta,
 - saldo de la cuenta, estatus de la cuenta
 - Api para obtener listado de transacciones de un cliente específico, en un rango de fechas el API debe recibir el código del cliente y el rango de fechas y debe mostrar **(Obligatorio)**.
 - ID de la transacción,
 - número de cuenta asociada a la transacción
 - Nombre del cliente al que pertenece la cuenta
 - fecha y hora de la transacción
 - monto
 - código de operación
 - Descripción de la transacción
 - tipo de operación (Débito o Crédito)

3. Entregables:

1. Modelo Entidad Relación
2. Scripts para creación de base de datos en MySQL
3. Scripts utilizados para inserción de datos de pruebas clientes, productos, cuentas, y códigos de operación.
4. Proyecto de Node donde están los APIs, utilizar VisualCODE como IDE para el desarrollo.
5. Utilizar node versión 16.
6. De preferencia utilizar type script como lenguaje de desarrollo.
7. Indicar donde se debe parametrizar la aplicación para la conexión a la base de datos
8. **Mandar todo en un archivo ZIP, con el nombre evaluación-{nombre del desarrollador}.zip**

4. El tiempo estimado para hacer esta prueba es de dos horas.