

LABORATORIO 3

El desarrollo del siguiente laboratorio debe hacerse de manera individual.

ENTREGABLE

1. Link del repositorio en Azure DevOps.
2. Debe adjuntar un documento en PDF donde se evidencien los resultados de cada una de las acciones posibles del sistema.
 - a. Ejemplo: Acción crear → Imagen del sistema.

PASO A PASO

PARTE 1: Haciendo uso de la arquitectura de acceso a datos y la arquitectura REST otorgada por el profesor, realice lo siguiente:

CenfoBanco lo contratado a usted para el desarrollo de un aplicativo que le permita llevar el control de toda la información de sus clientes, para ello en la primera etapa deberá desarrollar un aplicativo que le permita realizar los mantenimientos completos de:

1. Clientes
 - a. Cedula
 - b. Nombre
 - c. Apellido
 - d. Fecha Nacimiento
 - e. Edad
 - f. Estado Civil
 - g. Genero
2. Cuentas
 - a. Moneda
 - b. Tipo cuenta (Ahorros, corriente)
 - c. Saldo (Calculado en función de los movimientos de las cuentas)

3. Movimientos de la cuenta
 - a. Fecha movimiento
 - b. Tipo de movimiento(crédito/debito)
 - c. Monto del movimiento
 - d. Cuenta asociada
4. Operaciones de créditos
 - a. Monto
 - b. Tasa
 - c. Nombre de la línea
 - d. Cuota
 - e. Fecha inicio
 - f. Estado
 - g. Saldo operación (Calculado en función de los pagos)
5. Pagos de la operación de crédito
 - a. Fecha pago
 - b. Monto
 - c. Operación asociada
6. Direcciones
 - a. Provincia
 - b. Cantón
 - c. Distrito
 - d. Detalles

TECNOLOGIA A UTILIZAR

1. **Base de datos:** debe modelar la base de datos con las tablas necesarias para almacenar la información solicitada por el cliente. considere llaves primarias, llaves foráneas, índices y normalización de las estructuras.

Esta capa de desarrollarse usando:

- Base datos sql server alojada en Azure.

2. **Capa de acceso a datos:** implemente la capa de acceso de datos brindada por el profesor en el curso, esta capa debe permitir realizar las operaciones CRUD para cada una de POJOS definidos para el aplicativo.

Esta capa de desarrollarse usando:

- C# class library.

3. **Capa lógica o de negocio:** implemente la capa CORE, para el manejo de la lógica del negocio y del aplicativo.

En esta capa debe incorporar las siguientes validaciones:

- Cliente
 - Debe ser mayor de edad.
 - La fecha de nacimiento debe ser correcta.
- Cuenta
 - El saldo no puede ser negativo.
 - El tipo de cuenta solo puede ser A o C.
- Movimientos de la cuenta:
 - No debe haber movimientos en negativo.
- Operaciones de crédito:
 - No debe haber movimientos negativos.
- Direcciones
 - Debe validarse que sean provincias, cantones y distritos validos según distribución costarricense.

Esta capa de desarrollarse usando:

- C# class library.

4. **Capa API:** implemente la capa API, esta debe exponer todos los mantenimientos de las entidades anteriormente mencionadas, debe validar antes de procesar el request que los campos requeridos cuenten con valores y en el formato adecuado.
5. **Cliente:** desarrolle un proyecto que le permita al usuario las acciones básicas para cada mantenimiento.
 - a. Create
 - b. Update
 - c. Delete
 - d. Retrieve por id
 - e. Retrieve all

Dicho cliente debe desarrollarse usando:

- C# console application.

Este cliente debe consumir el WEB API.