

CORPORACIÓN DE CRÉDITO CONTACTAR
AREA: PROGRAMACIÓN
PRUEBA TECNICA - INGENIERO DE DESARROLLO
2021

“Nuestra mayor debilidad reside en rendirnos. La forma más segura de tener éxito es intentarlo una vez más.”- Thomas A. Edison.

PRERREQUISITOS:

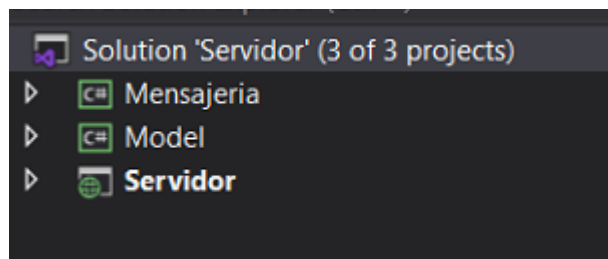
- Visual Studio 2017 (Profesional o Community)
- .Net Framework 4.6.1 y 4.5
- Entity Framework 6x
- Microsoft SQL Server 2017 o 2016 express Edition on Windows
- IIS (Internet Information Services)

CONTEXTO

Se desea Crear una solución de software con arquitectura cliente – servidor para la administración de clientes y el manejo de créditos de Contactar para lo cual se tiene 2 soluciones en Visual Studio:

A) SOLUCION SERVIDOR

Actualmente la solución servidor tiene 3 proyectos:



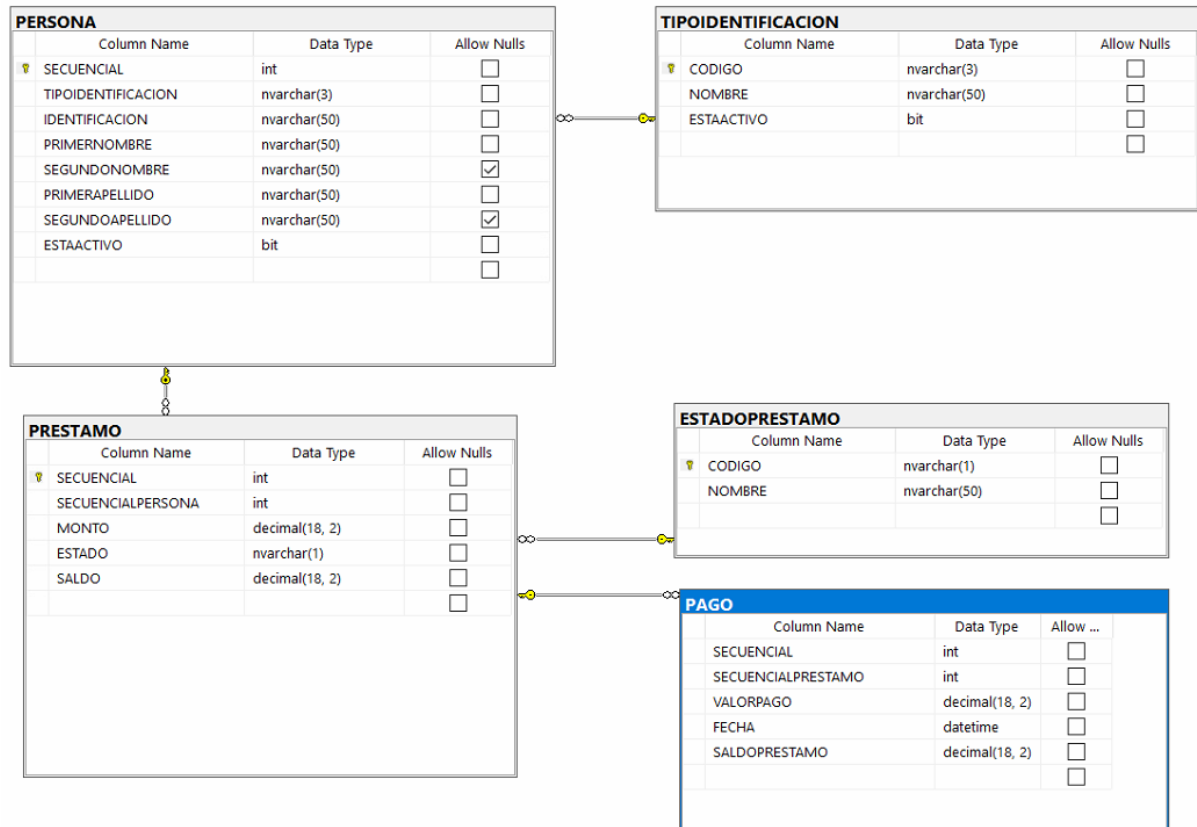
Mensajería: Permite definir las clases que se pueden serializar para enviar al cliente y servirán de enlace con los servicios.

Model: Contiene el modelo de la base de datos en EntityFramework y los métodos que interactuarán con los servicios, métodos de lógica de negocio y de persistencia.

Servidor: Contiene los servicios que podrá consumir el cliente. No contiene Lógica de negocios.

BASE DE DATOS

El siguiente es el diagrama Entidad – Relación de la base de datos denominada TestContactar la cual esta mapeada en Model mencionado anteriormente.



En la tabla PERSONA se guardan los potenciales clientes de Contactar a quienes posteriormente se podrá asignarles un crédito que se registrará en la tabla PRESTAMO (Solo se trabajará con capital, no se calcularán intereses u otros componentes de tablas de amortización):

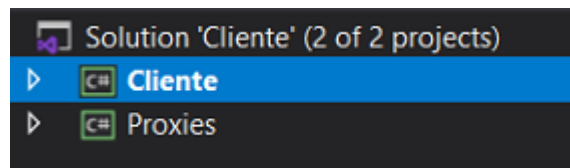
- El campo SECUENCIAL es un campo autoincremental y será utilizado como el identificador del préstamo.
- El campo SECUENCIALPERSONA es el Identificador de la tabla PERSONA (Identificador del cliente)
- El campo MONTO es el valor del crédito que se le otorga al cliente (este valor no puede ser menor a \$ 500.000 ni superior a 20 millones)
- El Campo SALDO lleva el control del saldo del crédito, Inicialmente será igual al monto prestado, y a medida que se realicen abonos éste valor irá disminuyendo por el valor de los abonos hasta llegar a cero.
- El Campo ESTADO es el estado en el que se encuentra el préstamo, inicialmente solo tendrá dos estados:
V = VIGENTE cuando se le otorga el crédito al cliente.
C = CANCELADO cuando la totalidad de la deuda haya sido pagada, es decir cuando el saldo sea igual a Cero.

Una vez que se le asigne un Préstamo a Un cliente, éste podrá realizar abonos a su crédito que se registrarán en la tabla PAGOS y se podrán realizar tantos abonos como el cliente desee hasta que el saldo del monto prestado quede en cero.

- El campo SECUENCIAL es un campo autoincremental y será el identificador del pago realizado.
- El campo SECUENCIALPRESTAMO es el identificador del préstamo (SECUENCIAL de la tabla PRESTAMO)
- El campo VALORPAGO registra el valor del pago o abono realizado al préstamo que no podrá superar el valor del saldo del préstamo.
- El campo FECHA registra la fecha y hora en la cual se realiza el abono, debe ser tomada automáticamente del sistema (equipo) en el momento del pago.
- El campo SALDOPRESTAMO lleva el control del saldo del préstamo después de cada abono realizado.

B) SOLUCIÓN CLIENTE

La solución cliente está compuesta por dos proyectos:



Cliente: Contiene los formularios para ingreso, edición, validación y edición de datos.

Proxies: Contiene las **Referencias Web** a los servicios.

EJERCICIO:

Por favor lea con mucha atención todo el ejercicio antes de iniciar con el desarrollo.

Se debe complementar la solución de software realizando correcciones, modificaciones y adiciones al código inicialmente entregado siguiendo los siguientes puntos:

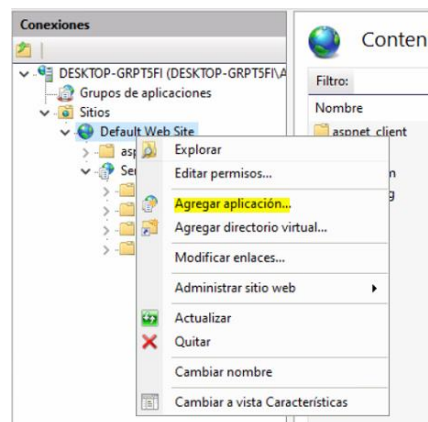
1. Implementación de la solución servidor:

1.1 Descomprimir el archivo TestContactar.zip en el disco D:

1.2 Publicar los servicios:

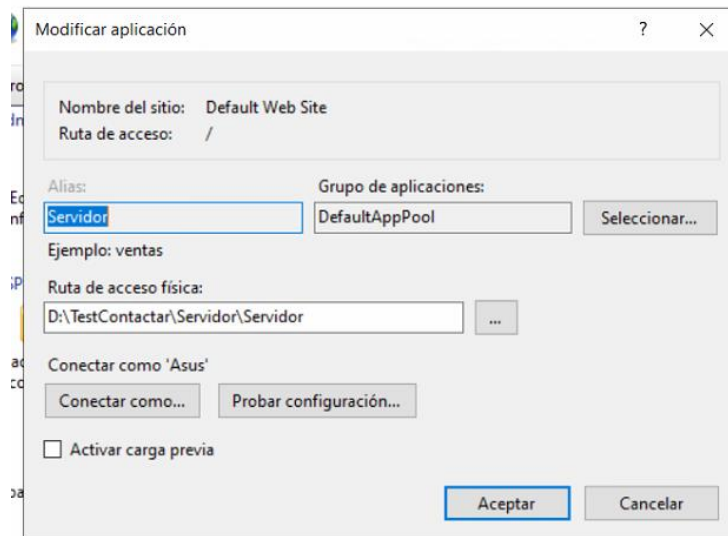
1.2.1 Abrir IIS

1.2.2 Agregar aplicación:

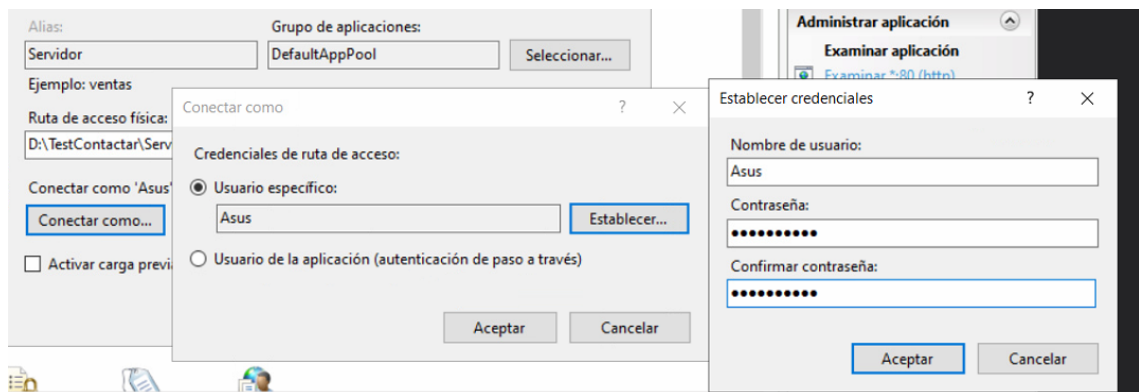


1.2.3 Configurar ruta y alias

En Alias se deberá configurar como **Servidor**

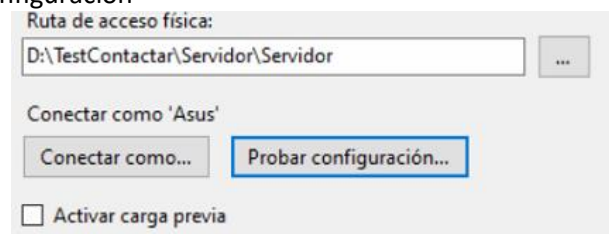


1.2.4 Agregar credenciales de conexión Clic en **Conectar Como...**

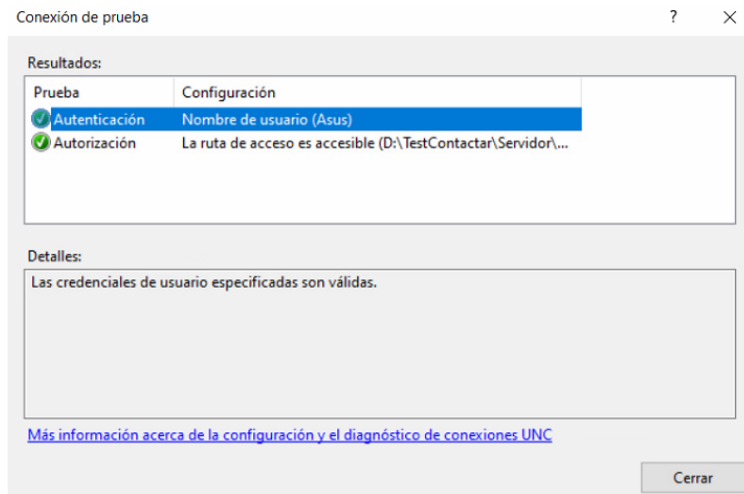


Y diligenciar los datos, el Nombre de usuario es el nombre del usuario de Windows con el que se inicia sesión al equipo y clic en Aceptar y Aceptar.

1.2.5 Prueba de Configuración: Clic en Probar Configuración



Si el usuario de ingreso y contraseña son correctas se mostrará el siguiente mensaje



1.2.6 Prueba de servicio:

Ingresar en la siguiente URL para verificar si el servicio quedó correctamente publicado:

<http://localhost/Servidor/Servicios/PersonaWS.asmx>

Si la solución servidor está correctamente publicada en IIS se mostrará la siguiente Imagen en el navegador

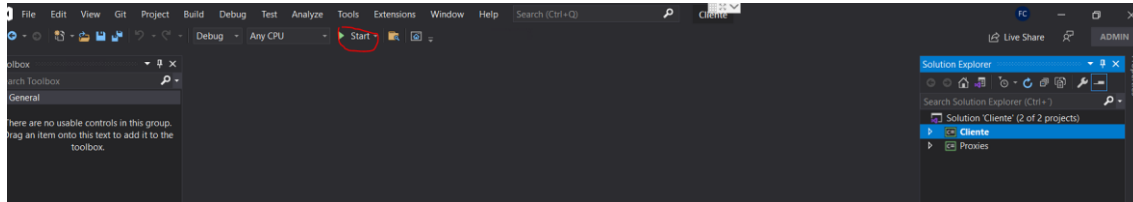


2. Restaurar la base de datos.

Restaurar la base de datos **TestContactar** (TestContactar.bak) con el mismo nombre en SQL Server Express. (Si tiene una versión diferente de sql express se adjunta el script para restaurar la base de datos con el nombre “ScriptBakcupTestContactar.sql”)

3. Ejecutar solución Cliente

Abrir la solución Cliente y ejecutarla en modo debug



Con la publicación del servidor en IIS, la base de datos restaurada y las cadenas de conexión correctamente configuradas se podrá probar que el cliente se encuentre funcionando correctamente accediendo a la siguiente ruta **Personas->Crear Personas->Clic botón Listar**.

Se deberá listar los clientes existentes actualmente en la base de datos:

Adicionar		Listar					
	SECUENCIAL	CODIGO TIPO IDENTIFICACION	IDENTIFICACION	PRIMERNOMBRE	SEGUNDONOMBRE	PRIMERAPELLIDO	ESTAATIVO
▶	1	CC	123456	JUAN		PEREZ	<input checked="" type="checkbox"/>
	2	CC	124	JULIAN	SEBASTIAN	RODRIGUEZ	<input checked="" type="checkbox"/>
	3	CC	111	LUISA	FERNANDA	ORTIZ	<input checked="" type="checkbox"/>

4. En la ruta anterior (Personas->Crear Personas->Clic botón Listar)

Se listan las personas que se encuentran creadas en la base de datos. Agregar a la Grilla El campo SEGUNDOAPELLIDO de la tabla PERSONAS para que sea visualizado en el listado. El campo se debe agregar después del PRIMERAPELLIDO y antes de ESTAACTIVO

- Se debe actualizar el diseño de la tabla PAGO mediante sentencias sql las cuales debe adjuntar en el archivo “**Punto 5 – Actualizar tabla PAGO.sql**” :

- 5.1 Crear el comando para actualizar el campo SECUENCIAL como llave primaria y como autoincremental
- 5.2 Escriba la sentencia para modificar el campo VALORPAGO a tipo de dato entero.

PAGO			
Column Name	Data Type	Allow N...	
SECUENCIAL	int	<input type="checkbox"/>	
SECUENCIALPRESTAMO	int	<input type="checkbox"/>	
VALORPAGO	decimal(18, 2)	<input type="checkbox"/>	
FECHA	datetime	<input type="checkbox"/>	
SALDOPRESTAMO	decimal(18, 2)	<input type="checkbox"/>	
		<input type="checkbox"/>	

6. Al ingresar una identificación el campo de Identificación y al hacer clic en el botón Buscar, se consulta en la base de datos si un cliente ya está registrado, (Ejemplo: Identificación 124). Pero cuando se busca una identificación que no existe se presenta el siguiente error (Ejemplo Identificación = 654).

Corregir el error mostrado en la captura de pantalla

7. Actualmente todas las personas que se registran se guardan con Cédula de Ciudadanía - CC como tipo de identificación por defecto. En la ventana para creación de personas agregar una lista desplegable que permita seleccionar el tipo de identificación. Para cargar esta lista desplegable se deberá construir un nuevo Servicio Web que retorne la lista de los tipos disponibles en la tabla TIPOIDENTIFICACION.
8. Registro de préstamos:
- 8.1 Agregar opción al menú existente que se denomine “Préstamos”.
- 8.2 Crear un formulario únicamente con los siguientes campos:
- Identificación
 - Monto

Con estos datos se debe registrar todos los campos requeridos en la tabla PRESTAMO. Se recuerda que por defecto todos los préstamos nuevos se registran en estado VIGENTE.

- 8.3 Crear un Servicio Web denominado “Préstamo” con los métodos necesarios para el registro de los préstamos desde el cliente.
- 8.4 Validar que un cliente no pueda tener más de un préstamo en estado VIGENTE, en caso de intentar registrar un nuevo crédito a un cliente que ya tiene un préstamo VIGENTE se debe mostrar el mensaje: “Cliente ya tiene un préstamo vigente, Por favor cancele el préstamo para otorgarle uno nuevo”.

Tener en cuenta las restricciones mencionadas en la descripción de la estructura de la base de datos. **TODAS RESTRICCIONES DEBEN SER VALIDADAS DESDE LA SOLUCIÓN SERVIDOR.**

9. Registro de Pagos

9.1 Agregar opción al menú existente que se denomine “Pagos”.

9.2 Crear un formulario únicamente con los siguientes campos:

- Secuencial Prestamo
- Valor

Con estos datos se debe registrar los abonos realizados a un préstamo teniendo en cuenta las restricciones mencionadas en la descripción de la estructura de la base de datos.

9.3 Crear un Servicio Web denominado “Pago” con los métodos necesarios para el registro de los pagos desde el cliente.

- El cliente puede registrar tantos pagos como desee sin superar el valor total del préstamo.
- El campo SALDOPRESTAMO inicia con el mismo monto del préstamo y por cada abono irá disminuyendo automáticamente (Realizar la actualización desde código, no en triggers de base de datos)

Ejemplo:

Creación del préstamo:

Prestamo 1

MONTO del préstamo 1.000.000

SALDO: 1.000.000

ESTADO: VIGENTE

Pagos:

Pago1 por 200.000

Resultado después de Pago1 para Prestamo 1

MONTO del préstamo 1.000.000

SALDO: 800.000

ESTADO: VIGENTE

Pago2 por 300.000

Resultado después de Pago2 para Prestamo 1

MONTO del préstamo 1.000.000

SALDO: 500.000

ESTADO: VIGENTE

Pago3 por 500.000

Resultado después de Pago3 para Prestamo 1

MONTO del préstamo 1.000.000

SALDO: 0

ESTADO: CANCELADO

10. Se desea conocer el valor del interés que se ha dejado de cobrar en cada préstamo.

Proponga y construya una solución que involucre el uso de servidor y cliente para visualizar el préstamo, la tasa que se aplica y el valor del interés que se debería aplicar a cada préstamo con las siguientes condiciones:

- El valor de la tasa debe ser tomada directamente consultando la tabla TASA donde:
 - el código PAR indica el valor de la tasa que se debe aplicar a los préstamos de los clientes con cédula terminada en dígito Par.
 - el código IMPAR indica el valor de la tasa que se debe aplicar a los préstamos de los clientes con cédula terminada en dígito Impar.
- El valor del interés se calculará de la siguiente forma:
 $\text{MONTO} * \text{Valor Tasa (Campo VALOR de la tabla TASA)} * 12$
Esta fórmula aplicará Independientemente del número de pagos que realice el cliente.

11. Crear un Procedimiento Almacenado para listar los préstamos por cliente, denominado "ListarPrestamos" que reciba como parámetro el número de identificación del cliente. En caso de recibir como parámetro el valor 0 deberá listar todos los préstamos de todos los clientes.

Los campos que debe mostrar el Procedimiento Almacenado son:

Tipoidentificación, número Identificación, nombre cliente, SecuencialPrestamo, monto, estado.

Adjuntar script del Procedimiento almacenado con nombre "Punto 11 – ListarPrestamos.sql"

12. Crear un Procedimiento Almacenado que liste todos los pagos que ha realizado un determinado cliente de todos sus préstamos y que reciba como parámetro el número de identificación. En caso de recibir como parámetro el valor 0 deberá listar todos los pagos de todos los préstamos.

Los campos que debe mostrar el Procedimiento Almacenado son:

Número Identificación, nombre cliente, SecuencialPrestamo, Valor del Pago, fecha de pago, valoracumuladodepagos (Suma de todos los valores pagados de todos los préstamos por persona)

Adjuntar script del Procedimiento almacenado con nombre "Punto 12 – ListarPagos.sql"

13. Crear un procedimiento almacenado que muestre las siguientes estadísticas recibiendo como parámetros FECHAINICIO y FECHAFIN.

Maximo valor pagado, mínimo valor pagado, valor promedio de pagos, cantidad total de pagos.

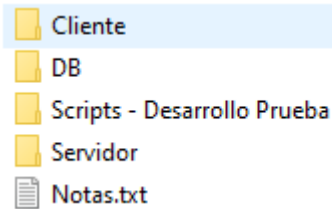
Adjuntar script del Procedimiento almacenado con nombre "Punto 13 – EstadisticaDePagos.sql"

En la base de datos se encuentra una tabla denominada NUMEROS que contiene los números del 1 al 10.000. Cree las sentencias sql para realizar las siguientes consultas:

14. Del campo VALOR mostrar aquellos valores que corresponden a números primos.
15. Del campo VALOR mostrar aquellos valores que son múltiplos de 5
16. Actualizar únicamente el campo VALOR de los múltiplos de 3 por el siguiente cálculo
 $\text{VALOR} = \text{VALOR} * 100$

ENTREGA DEL DESARROLLO DEL EJERCICIO:

Se deberá entregar un archivo comprimido denominado “Desarrollo-TestContactar-NOMBRE-APELLIDO.zip” que tendrá la siguiente estructura:



- **Cliente y Servidor:** Contienen el código fuente del desarrollo del ejercicio.
- **DB:** Contiene el backup resultado del desarrollo del ejercicio denominado “TestContactar_Resultado.bak”
- **Scripts – Desarrollo Prueba:** Contienen los scripts de las consultas y los procedimientos almacenados solicitados.
- **Notas: (OPCIONAL):** Alguna explicación que se quiera adicionar con respecto a:
 - Desarrollo de la prueba.
 - Cometarios de la prueba.
 - Propuestas de Mejora de código u optimización.
 - Cualquier otro tipo de comentario a tener en cuenta para revisión de la prueba, ejecución del código, etc.

PUNTO EXTRA 1 (OPCIONAL). (5 Puntos)

Cargar el desarrollo completo de la prueba con la estructura anterior sin comprimir en un repositorio de git hub Y enviar el enlace para descargar.

PUNTO EXTRA 2. (OPCIONAL). (20 Puntos)

Visualizar el resultado del procedimiento almacenado del punto 11 – ListarPrestamos en un reporte **.rdlc** en Cliente, utilizando servicios web.

ESCALA DE CALIFICACION	
NUMERO EJERCICIO	PUNTOS
1	0
2	0
3	0
4	5
5	5
6	7
7	10
8	10
9	12
10	15
11	7
12	7
13	7
14	5
15	5
16	5
TOTAL	100
EXTRA 1 – OPCIONAL	5
EXTRA 2 – OPCIONAL	20
TOTAL PLUS	125

¡MUCHOS ÉXITOS!