

Enunciado POO

Desarrollar una aplicación utilizando la Programación Orientada a Objetos y publicarla en Github como una rama independiente y que realice lo siguiente:

1. Permitir la carga de, al menos una Persona y sus Cuentas Bancarias. Una persona puede tener 0 a n cuentas.
2. Definir las clases Persona y Cuenta Bancaria, utilizando los principios de Abstracción, Encapsulamiento y Modularidad. Definir Atributos, Métodos Constructores, Generales y de Acceso a los Atributos.
 - a. Persona: Apellido y Nombre, CUIL, Sexo, Dirección, Teléfono.
 - b. Cuenta Bancaria: Banco (BNA, BCOFSA, GALICIA, FRANCES, HIPOTECARIO), Nro. De Cuenta, Tipo (CA/CC), CBU, Moneda (\$/US\$)
3. Establecer una Relación de Composición, entre una Persona y sus Cuentas Bancarias.
4. Implementar un algoritmo de validación de CBU, el cual funciona de la siguiente manera:



Dada una CBU: 2850590940090418135201

Validación 1º bloque (28505909) - DigitoVerificador = 9

Obtener:

$$\text{SUMA1} = 2*7 + 8*1 + 5*3 + 0*9 + 5*7 + 9*1 + 0*3 = 14 + 8 + 15 + 0 + 35 + 9 + 0 = 81$$

$$\text{DIFERENCIA1} = 10 - \text{Último digito de SUMA1} = 10 - 1 = 9$$

Si DIFERENCIA1 = DigitoVerificador2 => CBU OK

Validación 2º bloque (40090418135201) - DigitoVerificador = 1

Obtener:

$$\text{SUMA2} = C1*3 + C2*9 + C3*7 + C4*1 + C5*3 + C6*9 + C7*7 + C8*1 + C9*3 + C10*9 + C11*7 + C12*1 + C13*3$$

$$\text{SUMA2} = 12 + 0 + 0 + 9 + 0 + 36 + 7 + 8 + 3 + 27 + 35 + 2 + 0$$

$$\text{SUMA2} = 139$$

$$\text{DIFERENCIA2} = 10 - \text{Último digito de SUMA2} = 10 - 9 = 1$$

Si DIFERENCIA2 = DigitoVerificador => CBU OK

5. Implementar un método de Validación de CUIL en la clase Persona, según algoritmo presentado en el enunciado de la fila 2 del Primer Parcial.

Dado un CUIL 20-12345678-DigitoVerificador, el cálculo para validar es el siguiente:

$$(2*5)+(0*4)+(1*3)+(2*2)+(3*7)+(4*6)+(5*5)+(6*4)+(7*3)+(8*2)$$

$$=10+0+3+4+21+24+25+24+21+16 = 148$$

$$\text{Division entera } 148/11 = 13$$

$$\text{Calcular el Resto } 148 - (13 * 11) = 5 \text{ o bien } 148 \bmod 11 = 5$$

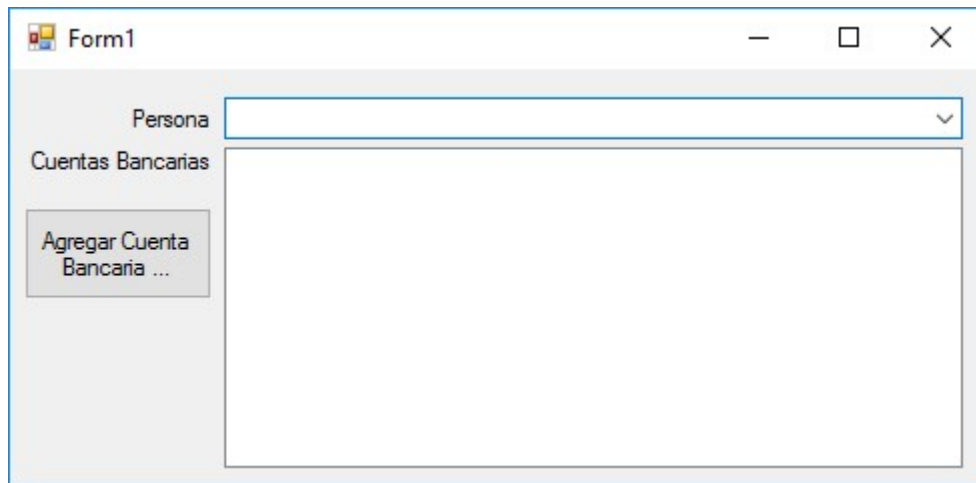
Si Resto = 0 entonces ***DigitoVerificador***= 0.

Si Resto = 1 entonces ***DigitoVerificador***= 9.

Sino ***DigitoVerificador***= 11 – Resto.

6. En la interfaz gráfica implementar:

- a. Página Principal (Form1) Muestra un combo con el listado de Personas y debajo un listado de las Cuentas Bancarias cargadas para la persona seleccionada. Un botón para el alta de una nueva cuenta bancaria para la persona seleccionada, si no hay persona seleccionada, debe permitir cargar una persona y su cuenta bancaria.

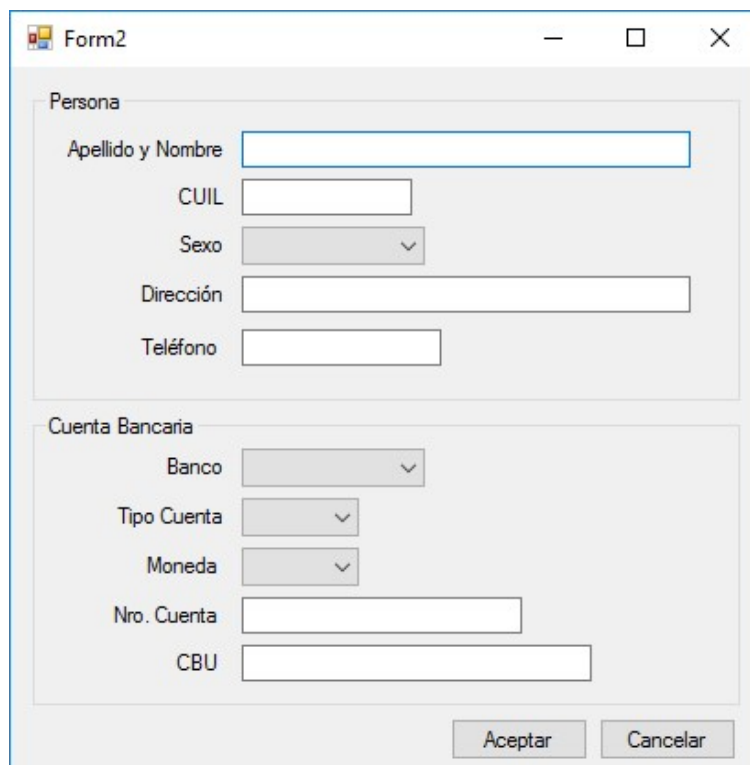


The screenshot shows a window titled "Form1". It contains a "Persona" dropdown menu. Below it is a section labeled "Cuentas Bancarias" which is currently empty. To the left of this section is a button labeled "Agregar Cuenta Bancaria ...".

- b. Carga de la Persona y/o Cuenta Bancaria, con Validación de CUIL y CBU.

- c. *Opcional*: Modificar/Eliminar Cuentas Bancarias y Personas.

Repositorio: <https://github.com/UNaF-TICs-Programacion3-2016/CuentasBancarias>



The screenshot shows a window titled "Form2". It is divided into two main sections: "Persona" and "Cuenta Bancaria". The "Persona" section contains fields for "Apellido y Nombre", "CUIL", "Sexo" (a dropdown), "Dirección", and "Teléfono". The "Cuenta Bancaria" section contains fields for "Banco" (a dropdown), "Tipo Cuenta" (a dropdown), "Moneda" (a dropdown), "Nro. Cuenta", and "CBU". At the bottom right of the window are two buttons: "Aceptar" and "Cancelar".