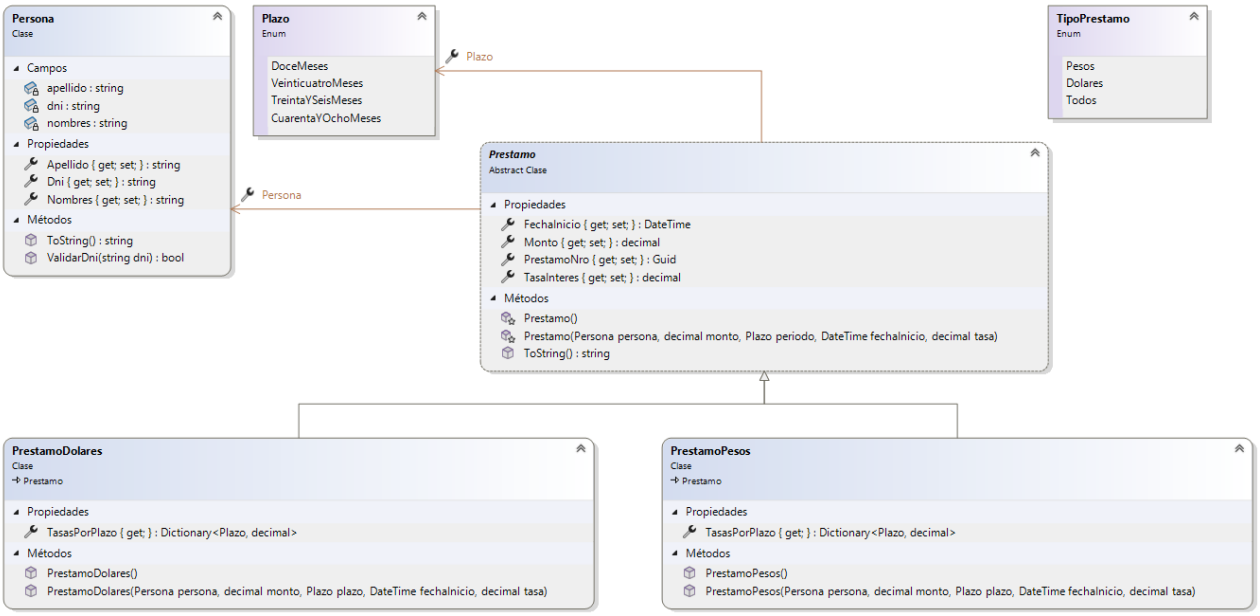
Recuperatorio Segundo Parcial POO

El sistema trata de manejar los movimientos de una entidad financiera, que realiza préstamos personales, tanto en dólares como en pesos, utilizando el sistema alemán para la cancelación de los pagos.

1. Crear las ***entidades*** siguientes en la capa de entidades del proyecto. ***(1 punto)***

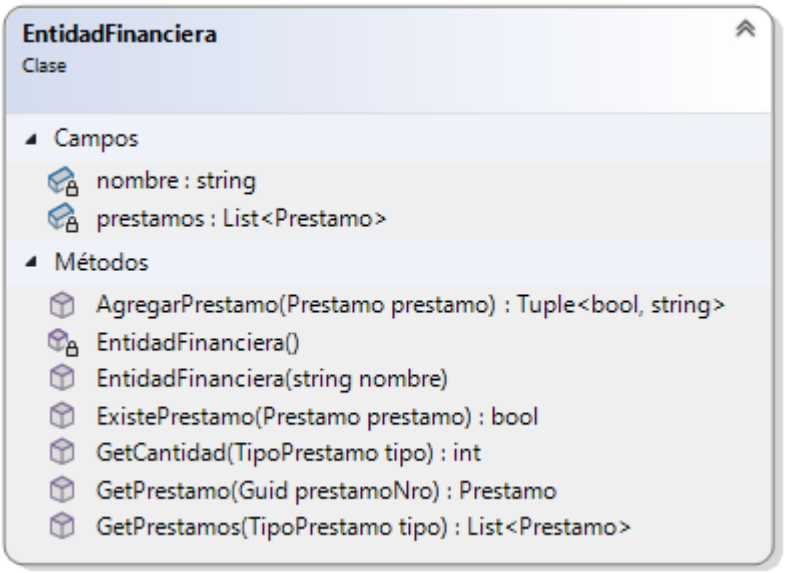


1.1. Enumeraciones ***Plazo*** y ***TipoPrestamo***:
   1.1.1. ***Plazo*** tiene como miembros 12, 24, 36 y 48 como valores de la misma.
   1.1.2. ***TipoPrestamo*** tiene como miembros Pesos, Dolares y Todos, comenzando en 1.
1.2. Clase ***Persona***.
   1.2.1. Todos sus atributos son de tipo ***nullable string*** y ***privados*** solo accesibles a través de sus propiedades de lectura y escritura.
   1.2.2. No se aceptan ***nulos*** en el ***nombre,*** así como tampoco en el ***apellido***.
   1.2.3. Tiene un método de clase para validar que el número de DNI sea de 8 números.
   1.2.4. No se aceptan los números de documentos que no hayan sido validados.
   1.2.5. Dos personas son iguales si tienen el mismo ***DNI***.
1.3. Clase ***Préstamo***:
   1.3.1. Los préstamos tienen un atributo ***persona*** de tipo ***Persona***.
   1.3.2. Todos los préstamos son tomados por un determinado ***Plazo***.
   1.3.3. Los préstamos tienen una ***fecha de inicio***.
   1.3.4. Los préstamos, según su plazo, y el tipo del mismo, tendrán una ***tasa de interés anual*** de tipo decimal.
   1.3.5. Los préstamos serán tomados por un determinado ***monto*** de tipo decimal.
   1.3.6. El atributo ***PrestamoNro***, será de tipo ***Guid*** y se establecerá en el constructor del mismo utilizando➔ PrestamoNro=Guid.NewGuid();
   1.3.7. Todos los atributos tienen propiedades de lectura/escritura.
   1.3.8. La clase tiene un método ***ConfigurarTasaIntereses*** que, fijará la tasa de interés, basándose en el plazo y en los datos de la propiedad ***TasasPorPlazo***.
   1.3.9. La clase tiene un método ***CalcularCuotas*** que devuelve una lista con los datos de las cuotas de cada préstamo.
   1.3.10. La ***sobreescritura del método ToString()***, debe mostrar el tipo de préstamo, la fecha, el monto, los intereses y el plazo, todos los datos en línea separadas.

1.3.11. ***Clases derivadas***: las clases ***PrestamoDolares*** y ***PrestamoPresos***, derivan de la clase ***Prestamo*** e implementan algún método, variable, o variables estáticas para suministrar la tasa anual de acuerdo con el plazo del préstamo, según la siguiente tabla:

Tipo Préstamo	Plazo	Tasa Anual
Dólares	12	5%
	24	6%
	36	7%
	48	8%
Pesos	12	25%
	24	30%
	36	35%
	48	40%

2. Capa de Datos***(1 punto).***



2.1. La clase ***EntidadFinanciera*** tendrá como atributos privados, una ***lista de préstamos*** y un ***string*** para almacenar su ***nombre***.

2.2. ***Constructores***:

2.2.1. Tendrá un constructor de clase para recibir el nombre.

2.2.2. Tendrá un constructor privado que será el único lugar donde se instanciará la ***lista de préstamos***.

2.2.3. Tendrá un método ***AgregarPrestamo*** que recibirá un ***préstamo*** y devolverá una ***tupla <bool,string>*** informando si se pudo agregar el préstamo o no, y, en el caso afirmativo, devuelve los datos del préstamo (reutilizar código), caso contrario una leyenda aclaratoria.

2.2.4. Tendrá un método ***ExistePrestamo***, que recibirá un ***préstamo*** y, constatará que no haya otro préstamo con los mismos datos, esto es, misma persona, fecha, monto, plazo y tipo.

2.2.5. Tendrá un método ***GetPrestamo***, que recibirá un ***Guid*** con el código del préstamo que se desea consultar, y devolverá el préstamo, si es que el mismo existe.

2.2.6. Tendrá un método ***GetPrestamos***, que recibirá un ***TipoPrestamo*** (enumeración), indicando los préstamos a informar, devolviendo ***la lista*** de los mismos.

2.2.7. Tendrá un método GetCantidad, que recibirá un ***TipoPrestamo*** (enumeración), indicando la cantidad de préstamos.

3. Utilizar el proyecto de ***Windows Form*** suministrado para manejar la entidad financiera. ***(1 punto)***

3.1. Formulario: ***frmPrestamos***

1.1.1. ***Botón Nuevo***, abre el formulario y devuelve los datos de un préstamo que se guardará en la lista de préstamos de la entidad.

1.1.2. Luego de cada operación se debe mostrar un mensaje aclaratorio de lo sucedido, reutilizando código, y se debe actualizar la grilla de la billetera, además se deberá actualizar la cantidad de préstamos.

1.1.3. Formulario: ***frmPrestamoAE***

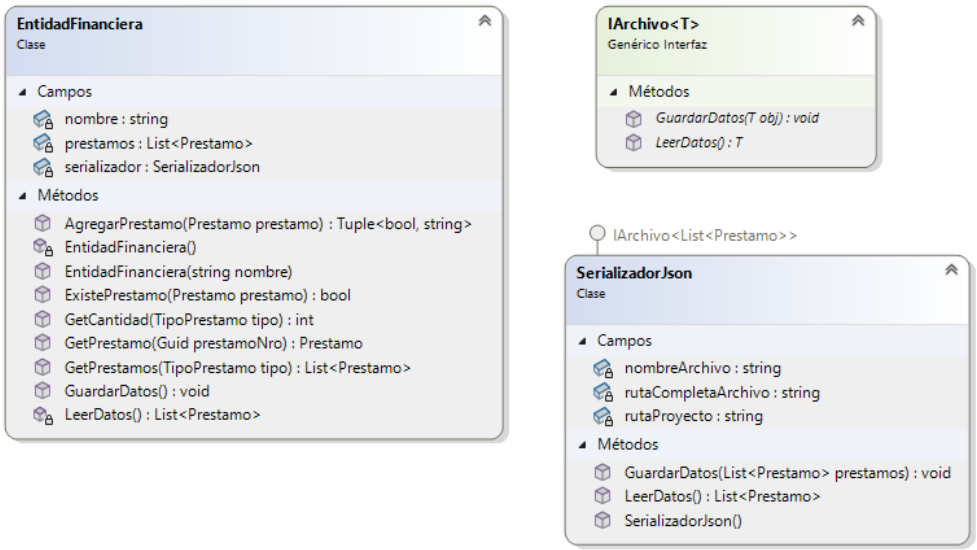
1.1.4. Al ***cargar el formulario***, el ***combo*** debe tener sus ítems completos con los ***plazos*** de los préstamos.

1.1.5. El botón ***Validar*** primeramente, verificará que el número de documento ingresado sea válido (***utilizar método previamente desarrollado***) y luego, habilitará a llenar los campos correspondiente al tomador del préstamo, siempre y cuando el documento sea válido, caso contrario mostrará un mensaje de error.

1.1.6. Cuando el elemento seleccionado en el ***combo de plazos*** cambia, se debe actualizar la tasa anual del préstamo, según el plazo y el tipo de préstamo.

1.1.7. El botón ***OK*** chequea que todo esté como corresponde y crea el préstamo con todos sus datos, caso contario muestra los mensajes de error pertinentes.

4. Agregar lo siguiente en la capa de Datos del proyecto. (***1 punto***)



1.2. Interface ***IArchivo<T>*** :

1.2.1. Interface genérica que tiene el enunciado de los métodos ***GuardarDatos*** y LeerDatos.

1.3. Clase ***SerializadorJson***:

1.3.1. Implementa la interface ***IArchivo<T>*** con el objeto ***List<Prestamo>.***

1.3.2. Tiene atributos privados

1.3.2.1. ***nombreArchivo***➔Prestamos.json

1.3.2.2. ***rutaPrograma***➔es la ruta donde se ejecuta el programa.

1.3.2.3. ***rutaCompletaArchivo***➔ruta completa donde se deberá guardar y leer el archivo Json.

1.3.3. El método ***GuardarDatos***, deberá guardar todos los préstamos

1.3.4. El método ***LeerDatos***, deberá proporcionar toda la lista de préstamos que se efectuaron.

1.4. Clase ***EntidadFinanciera*** (Parte 2):

1.4.1. Tiene además una propiedad de tipo ***SerializadorJson***.

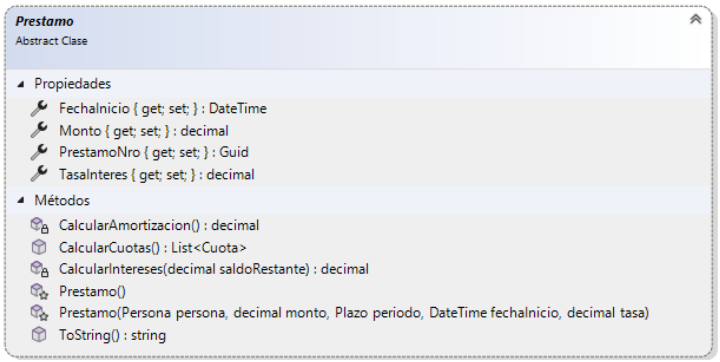
1.4.2. La entidad ahora debe contar con un método para ***GuardarDatos*** utilizando el ***SerializadorJson***, para guardar los datos en un archivo ***Json***, llamado “Prestamos.Json”.

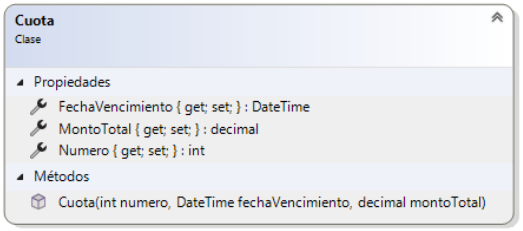
1.4.3. La ***entidad***, debe contar con un método para poder recuperar dichos datos del archivo Json, e incorporarlos al atributo correspondiente, utilizando también el ***SerializadorJson***.

1.5. Reflejar en la capa de presentación los cambios de guardado y recuperación de datos, mostrando los mismos en la grilla y mediante mensajes ilustrativos de ambas operaciones.

Si resolviste todo hasta acá y anda aprobaste con 4

5. Capa de Entidades (2da parte). ***(3 puntos)***





5.1. La clase ***Cuota***, tiene como atributos ***Numero*** como entero, ***FechaVencimiento*** como DateTime y ***MontoTotal*** como decimal, todos accesibles a través de sus propiedades de lectura y escritura.

5.2. La clase ***Prestamo***, incorpora métodos para el cálculo de las cuotas del préstamo:

5.2.1. La clase tiene un método privado para calcular la ***amortización*** del préstamo:

5.2.1.1. Fórmula➔Amortización=$\frac{Monto}{Tasa\ Mensual}$, donde Monto es el total del préstamo y la tasa mensual, es el porcentaje de la tasa de interés anual, dividido 12.

5.2.2. La clase tiene un método privado para calcular el ***interés*** de cada cuota, donde el mismo recibe como parámetro, el monto restante del préstamo, y sobre el mismo se calcula el interés

5.2.2.1. Fórmula➔$SaldoRestante * Tasa\ Mensual$, donde el saldo restante es, el monto del préstamo menos la amortización y, la tasa mensual es igual que en el punto anterior.

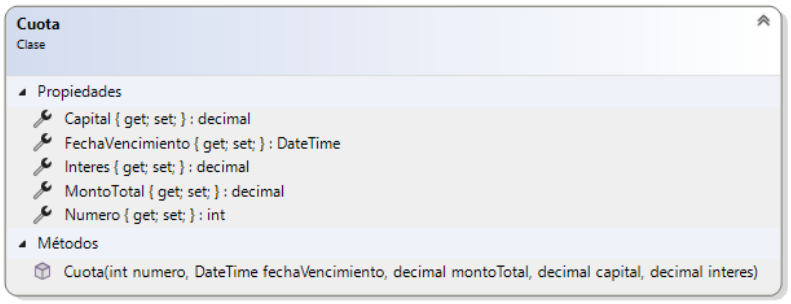
5.2.3. La clase tiene un método que devuelve una ***List<Cuota>*** con el detalle de las cuotas del préstamo, teniendo en cuenta que, la cantidad de cuotas debe coincidir con el plazo tomado, que la fecha de vencimiento de cada cuota es a los 30 días, desde la fecha de la toma del préstamo, y que los intereses se calculan sobre el saldo restante del préstamo.

5.3. Mostrar Detalle de cuotas.

5.3.1. En el formulario principal, seleccionar un préstamo y mediante la pulsación del botón correspondiente, mostrar el formulario ***frmDetallePrestamo*** con los datos del mismo y el detalle de las cuotas.

5.3.2. Toda la información numérica de las cuotas e intereses tienen que estar con 2 decimales.

6. Modificaciones y agregados ***(1 punto)***

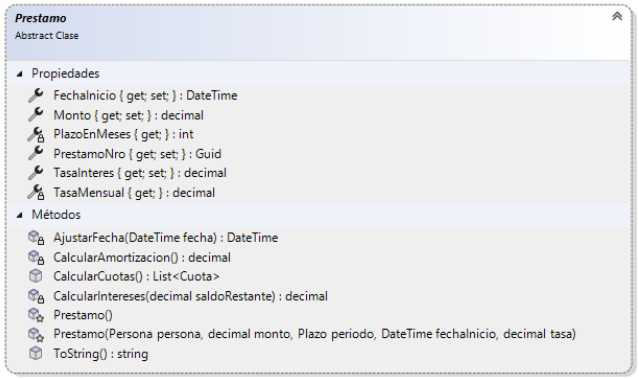


6.1. La clase ***Cuota*** incorpora los atributos ***Capital*** e ***Interes***, ambos de tipo decimal.

6.2. La clase ***Prestamo***, ahora, discrimina en los atributos agregados a la clase ***Cuota*** los importes pertinentes, siendo ***MontoTotal*** un campo calculado.

6.3. Mostrar en el formulario de detalle, los nuevos importes agregados en la clase.

6.4. Ajustar los vencimientos de las cuotas:



6.4.1. La clase ***Prestamo***, incorpora un *método privado* para ajustar el vencimiento de la cuota, si la misma cae en un día de fin de semana.

7. Filtros: ***(1 punto)***

7.1. Filtrar los registros por tipo de préstamos.

7.2. Ajustar la cantidad de registros.

8. Ordenamiento: ***(1 punto)***

8.1. Mostrar los registros ordenados por Monto del préstamo, ascendente y descendente.

8.2. Botón Actualizar, muestra todos los registros en el orden en el que fueron ingresados