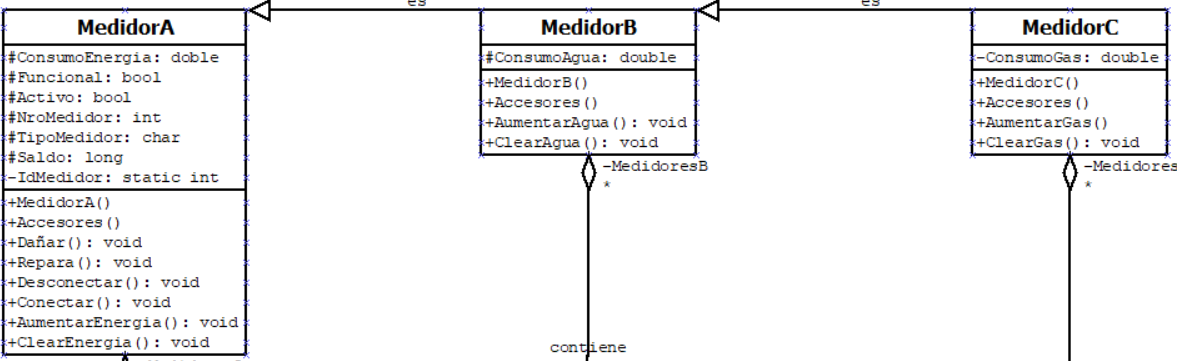
**Diagrama dia extendido**



Los tres tipos de medidores comparten los atributos y métodos del medidor tipo A, por ello esta clase es el padre, el hijo que viene siendo el medidor tipo B tiene además el atributo y los métodos correspondiente al consumo de agua y viene siendo el padre del medidor tipo C que además contiene su atributo y métodos propios correspondientes al consumo de gas.

**Atributos MedidorA:**

* **Consumo energía:** Es para tener el manejo del consumo de energía que genera el medidor.
* **Funcional:** Para controlar si el medidor esta funcionando (true) o esta dañado (false).
* **Activo:** Para controlar si el medidor esta activo (true) o si esta inactivo (false), que hace referencia a si la empresa desconecta el servicio.
* **NroMedidor:** Este es para darle un númeo diferente a cada objeto instanciado que lo controla el atributo IdMedidor.
* **TipoMedidor:** Para poder identificar que tipo de medidor es (A, B o C).}
* **Saldo:** Corresponde al manejo del saldo del medidor
* **IdMedidor:** Aumento unitario cada que se instancia un objeto y se le asigna ese número al atributo NroMedidor.

**Métodos MedidorA:**

* **MedidorA:** Es el constructor de la clase, inicializa los atributos boolean en true y establece el atributo NroMedidor.
* **Accesores:** Corresponde a los accesores de cada atributo solo con el get.
* **Dañar:** Cuando se llama al método pone en false el atributo Funcional para simular que el medidor se dañó.
* **Reparar:** Pone el atributo Funcional en true, para simular la reparación del medidor.
* **Desconectar:** Pone en false el atributo Activo para simular que la empresa desconectó el servicio
* **Conectar:** Pone en true el atributo Activo para simular que la empresa activo el servicio
* **AumentarEnergia:** Genera un número aleatorio que se le suma al atributo ConsumoEnergia para simular el aumento de energía y también va sumando el saldo multiplicando por 1000 al numero generado.
* **ClearEnergia:** Pone el ConsumoEnergía el Saldo en 0 .

**Atributos MedidorB:** Hereda los atributos del MedidorA que tienen visibilidad Protegido

* **ConsumoAgua:** Para llevar el control del consumo de agua que se genera.

**Métodos MedidorB:** Hereda los métodos del MedidorA

* **MedidorB:** Está en base del constructor del MedidorA, el cual ya genera el número para el atributo NroMedidor lo que hace que todos los objetos hijos y nietos tengan un número diferente y no hace necesario heredar el atributo IdMedidor, en lo que cambia es que este Constructor pone el atributo TipoMedidor igual a B.
* **Accesores:** También hereda los accesores y tiene el accesor del atributo ConsumoAgua solo con el get.
* **AumentarAgua:** Genera un numero aleatorio para simular el consumo de agua, este numero se le suma al atributo ConsumoAgua y también al Saldo multiplicado por 1500.
* **ClearAgua:** Pone el 0 el atributo de ConsumoAgua.

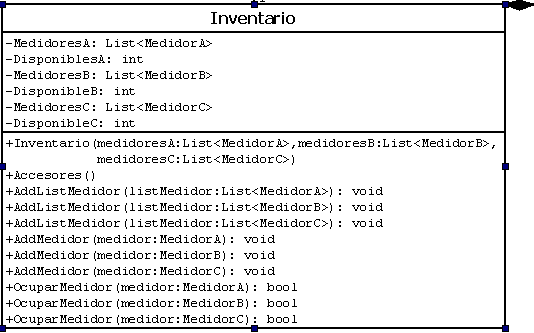
**Atributos MedidorC:** Hereda todos los atributos del MedidorB y por medio de este los atributos del MedidorA.

* **ConsumoGas:** Para llevar el consumo del gas, de la misma forma que los atributos ConsumoAgua y ConsumoEnergia

**Métodos MedidorC:** Hereda todos los métodos del MedidorB y por medio de este los métodos del MedidorA.

* **MedidorC:** Constructor en base del constructor del MedidorB que esta en base del constructor del MedidorA, que inicializa loas atributos y genera el numero para el atributo NroMedidor, lo diferente a los otros, es que este pone el atributo TipoMedidor igual a C.
* **Accesores:** Hereda los accesores y se le añade el accesor del ConsumoGas solo con el get.
* **AumentarGas:** Genera el aleatorio para simular el aumento del gas y se le suma al saldo este número multiplicado por 500.
* **ClearGas:** Para poner en 0 el atributo del consumo del gas.

Estas tres clases tiene una relación de agregación con la clase Inventario, debido a que esta clase tiene los medidores de cada tipo y es de agregación porque los medidores pueden estar por fuera de esta clase.



La clase Inventario contiene los medidores que están disponibles por parte de la empresa (por esto algunos medidores pueden estar por fuera de esta clase).

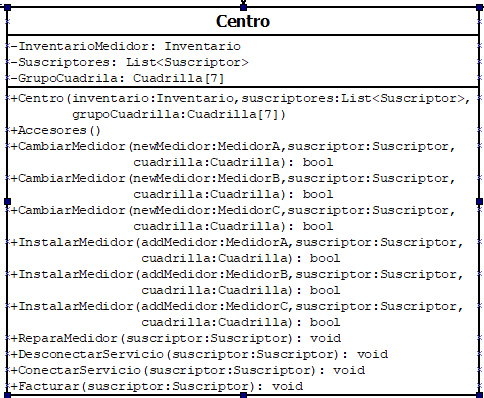
**Atributos Inventario:**

* **MedidoresA, MedidoresB, MedidoresC:** Los atributos corresponde a una lista de las clases MedidorA, MedidorB y MedidorC.
* **DisponibleA, DisponibleB, DisponibleC:** Los tributos corresponden al tamaño de las listas MedidoresA, MedidoresB, MedidoresC, respectivamente, porque solo se controla los que esta disponibles y cada vez que se utiliza un medidor de la lista, este medidor es eliminado de la lista que se utilizó.

**Métodos Inventario:**

* **Inventario:** Constructo que recibe las listas de los medidores.
* **Accesores:** Accesores de los atributos solo con el get
* **AddListMedidor:** Método sobre cargado que sirve para llenar las listas de medidores cuando estas ya estén vacías.
* **AddMedidor:** Método sobre cargado para añadir un medidor a la lista correspondiente, se valida que este medidor no esté en la lista por medio del atributo NroMedidor.
* **OcuparMedidor:** Se utiliza para eliminar el medidor de la lista correspondiente, se busca el medidor por medio del atributo NroMedidor.

Esta clase tiene una relación de composición con la clase Centro, debido a que es la empresa la única que contiene un inventario de los medidores disponibles.



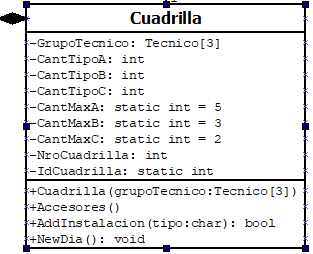
**Atributos Centro**

* **InventarioMedidor:** Contiene un objeto de la clase Inventario
* **Nombre:** Un atributo para el nombre de la empresa.
* **GrupoCuadrilla:** Es un vector de tipo Cuadrilla de 7 posiciones que controla las 7 cuadrillas que tiene la empresa para hacer las operaciones a los medidores.

**Métodos Centro:**

* **Centro:** Constructor del centro que recibe el inventario, la lista de suscriptores y el grupo de la cuadrilla
* **Accesores:** los accesores de los atributos solo con el get.
* **CambiarMedidor:** Para cambiar el medidor de un suscriptor, se necesita un medidor que este contenido en el InventarioMedidor para llamar al método OcuparMedidor de dicha clase y validar de que hayan disponibles, además se requiere también enviar una Cuadrilla que este en GrupoCuadrilla para validar de que se puede hacer la operación, también se debe enviar el suscriptor que se buscará en la lista Suscriptores por medio del atributo NroSuscriptor para simular el cambio del medidor.
* **InstalarMedidor:** Para instalar el medidor de un suscriptor, se necesita un medidor que este contenido en el InventarioMedidor para llamar al método OcuparMedidor de dicha clase y validar de que hayan disponibles, además se requiere también enviar un Cuadrilla que este en GrupoCuadrilla para validar de que se puede hacer la operación, también se debe enviar el suscriptor que se buscará en la lista Suscriptores por medio del atributo NroSuscriptor para simular el cambio del medidor.
* **RepararMedidor:** Se necesita enviar el suscriptor que se buscará en la lista Suscriptores por medio del atributo NroSuscriptor para llamar el método Reparar del medidor que tenga el suscriptor
* **DesconectarServicio:** Se necesita enviar el suscriptor que se buscará en la lista Suscriptores por medio del atributo NroSuscriptor para llamar el método Desconectar del medidor que tenga el suscriptor
* **ConectarServicio:** Se necesita enviar el suscriptor que se buscará en la lista Suscriptores por medio del atributo NroSuscriptor para llamar el método Conectar del medidor que tenga el suscriptor
* **Facturar** Se necesita enviar el suscriptor que se buscará en la lista Suscriptores por medio del atributo NroSuscriptor para llamar el método Pagar del suscriptor

La clase Centro, tiene una relación con la clase Cuadrilla de composición debido a que la cuadrilla hace parte de la empresa.



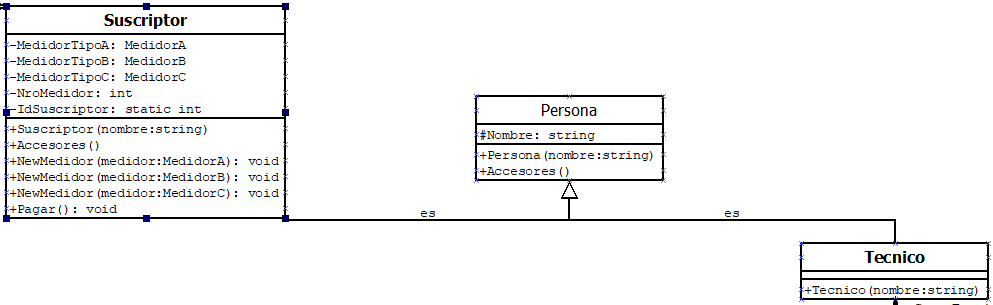
**Atributo Cuadrilla**

* **GrupoTecnico:** Este atributo es un vector de 3 posiciones de tipo Tecnico que simula los técnicos que están en una cuadrilla
* **CantTipoA:** Se usa para controlar la cantidad de medidores tipo A han instalada y/o cambiado en el día.
* **CantTipoB:** Se usa para controlar la cantidad de medidores tipo B han instalada y/o cambiado en el día.
* **CantTipoC:** Se usa para controlar la cantidad de medidores tipo C han instalada y/o cambiado en el día.
* **CantMaxA:** Se usa para controlar la cantidad de medidores tipo A máxima que pueden instalar y/o cambiar en el día.
* **CantMaxB:** Se usa para controlar la cantidad de medidores tipo B máxima que pueden instalar y/o cambiar en el día.
* **CantMaxC:** Se usa para controlar la cantidad de medidores tipo C máxima que pueden instalar y/o cambiar en el día.
* **NroCuadrilla:** Atributo para controlar que cada cuadrilla tenga un numero diferente
* **IdCuadrilla:** Atributo que controla el NroCuadrilla para que siempre se han diferentes.

**Métodos Cuadrilla:**

* **Cuadrilla:** El constructor recibe el vector con los 3 tecnicos y aumenta el atributo IdMedidor que se iguala al atributo NroMedidor para que todos los objetos tipo Cuadrilla tenga este atributo diferente.
* **Accesores:** Los accesores correspondientes a todos los atributos.
* **AddInstalacion:** Este método recibe un char que seria para el tipo de instalación (A, B o C), para simular que la cuadrilla realizó una instalación o cambio de medidor, retorna un boolean, true si la cantidad de instalaciones o cambios es menor que la máxima cantidad permitida y aumenta uno al atributo correspondiente al tipo de instalación (CantTipoA, CantTipoB o CantTipoC).
* **NewDia:** Pone los atributos CantTipoA, CantTipoB y CantTipoC en 0.

Esta clase contiene técnicos, por lo cual tiene una relación de composición con la clase Tecnico, debido a que los objetos de este tipo solo tienen sentido si se encuentran en la clase Cuadrilla.



Las clases Tecnico y Suscriptor tiene una relación con la clase Persona de tipo herencia, debido a que las primeras son personas. También la clase Suscriptor tiene una relación de asociación con la clase Centro, porque el Centro le presta un servicio a un Suscriptor, por el cual este obtiene un tipo de medidor, en este sentido Suscriptor tiene una relación de composición con MedidorA, MedidorB y MedidorC.

**Atributos Persona:**

* **Nombre:** es un string que determina el nombre de la persona

**Métodos Persona:**

* **Persona:** El constructor de persona que valida el nombre.
* **Accesores:**  Corresponde al accesor del nombre.

**Atributos Tecnico:** Tiene el Nombre que hereda de persona.

**Métodos Tecnico:**

* **Tecnico:** Constructor de técnico en base al constructor de Persona.

**Atributos Suscriptor:** Hereda el atributo Nombre de Persona.

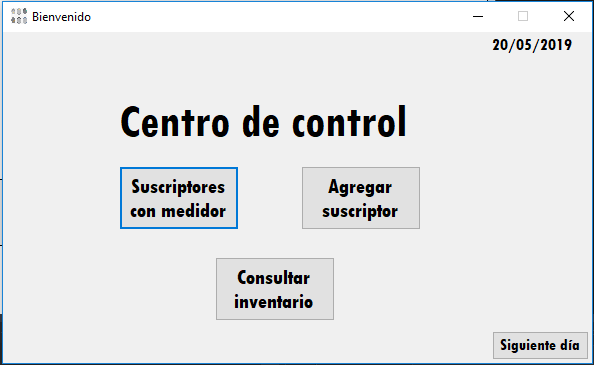
* **MedidorTipoA:** Es de tipo MedidorA por si el suscriptor adquiere un medidor de tipo A
* **MedidorTipoB:** Es de tipo MedidorA por si el suscriptor adquiere un medidor de tipo B
* **MedidorTipoC:** Es de tipo MedidorA por si el suscriptor adquiere un medidor de tipo C
* **NroSuscriptor:** Este atributo es para diferenciar a los diferentes números.
* **IdSuscriptor:** Cuando se instancia un objeto aumenta 1 y se iguala al atributo NroSuscriptor para que todos los objetos tengan un numero diferente.

**Métodos Suscriptor:**

* **Suscriptor:** Constructor en base al constructo de Persona
* **Accesores:** Corresponde a los accesores de los atributos (dos de ellos en null, debido a que el suscriptor solo tiene un medidor)
* **NewMedidor:** Método sobre cargado para cada tipo de medidor. Recibe un medidor y el atributo correspondiente lo hace igual al parámetro, los otros dos atributos los pone en null.
* **Pagar:** Por medio de if valida que atributo que no este en null para llamar a los métodos ClearEnergia, ClearAgua y ClearGas para simular que pagó los servicios y el saldo del medidor se ponga en 0 (por medio del método ClearEnergia que todos los medidores lo tienen).

**Visual**

**FmInicio:** Cuando se abre el form se llama a un método Inicializar que lee los archivos, crea los suscriptores, los medidores y el centro para simular la empresa.



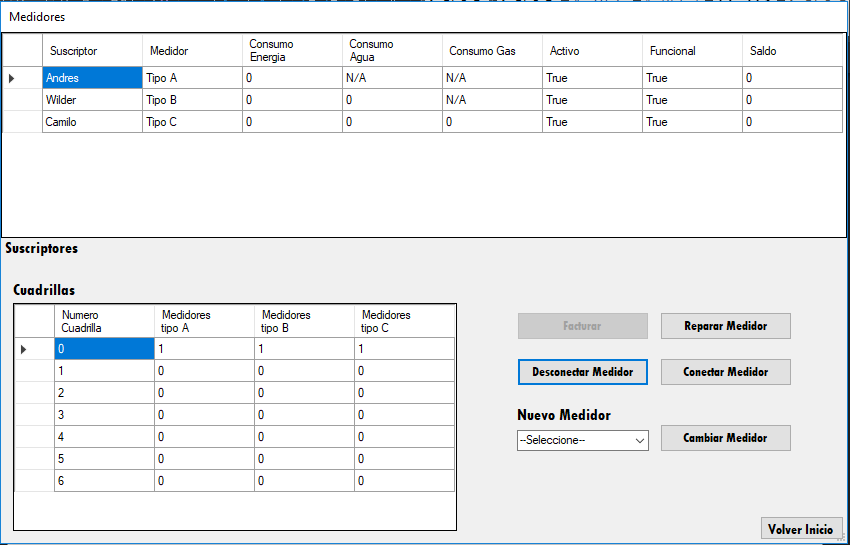
**Botons:**

* **BtnMedidores (Suscriptores con medidor):** Abre el form FmMedidores, que contiene una tabla para los suscriptores de la empresa y la cuadrilla, además de botones para realizar las operaciones en los suscriptores.
* **BtnAddSuscriptor (Agregar suscriptor):** Abre el form FmSuscriptores, que contiene una tabla de las cuadrillas para agregar un suscriptor e instalarle un medidor.
* **BtnInventario (Consultar inventario):** Abre el form FmInventario que contiene una tabla del inventario de la empresa, se pueden añadir n medidores a la lista.
* **BtnSiguiente (Siguiente dia):** Simula el paso del tiempo por día que se pone en el label LbDia, además se genera un numero aleatoria para determinar si se llama el método Dañar del medidor del suscriptor para simular que el medidor se dañó y se llama el método NewDia de la Cuadrilla para simular el paso del tiempo en estas.

**Labels:**

* **LbDia (Dia):**  Se pone el día que se esta simulando en fecha corta (DD/MM/AAAA).
* **LbCentro (Centro de control):** Para poner el título.

**FmMedidores:** Cuando se abre se llama un método Inicializar que llena las dos tablas según las listas que se encuentran en el objeto tipo Centro (GetCentro) en el form FmInicio.



**Botones:**

* **BtnFacturar (Facturar):** Se activo cuando se esta en el día 30 del mes, el cual lo dice el label LbDia del form FmInicio, cuando esta activo se selecciona una fila de la table de suscriptores y para llamar el método Facturar del centro que se le envía como parámetro el suscriptor seleccionado y se actualiza la tabla volviendo a llamar el método Inicializar.
* **BtnReparar (Reparar Medidor):** Se selecciona una fila de la tabla suscriptores y se llama el método RepararMedidor del centro que se le envía como parámetro el suscriptor seleccionado y se actualiza la tabla volviendo a llamar el método Inicializar.
* **BtnDesconectar (Desconectar Medidor):** Se selecciona una fila de la tabla suscriptores y se llama el método DesconectarServicio del centro que se le envía como parámetro el suscriptor seleccionado y se actualiza la tabla volviendo a llamar el método Inicializar.
* **BtnConectar (Conectar Medidor):** Se selecciona una fila de la tabla suscriptores y se llama el método ConectarServicio del centro que se le envía como parámetro el suscriptor seleccionado y se actualiza la tabla volviendo a llamar el método Inicializar.
* **BtnCambiar (Cambiar Medidor):** Se selecciona una fila de la tabla suscriptores y una fila de la tabla de cuadrilla, además una opción del combo box CbTipoMedidor, se llama el método CambiarMedidor del centro que se le envía como parámetro, el medidor nuevo, el suscriptor seleccionado y la cuadrilla seleccionada, y se actualizan las tabla volviendo a llamar el método Inicializar.
* **BtnVolver (Volver Inicio):** Para cerrar el form actual y se abra el form FmInicio.

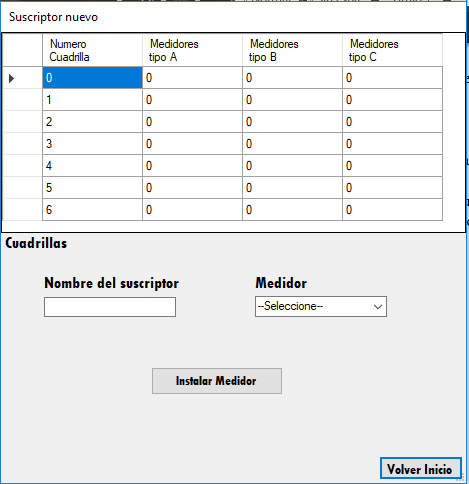
**TextBox:**

* **CbTipoMedidor:** Combo box para seleccionar una opción para el nuevo medidor que se desea cambiar.

**Tables:**

* **TbSuscriptores:** Para visualizar los suscriptores, se pueden seleccionar para hacer las diferentes operaciones con los botones.
* **TbCuadrilla:** Para visualizar las cuadrillas con el numero de operaciones que han realizado por tipo de medidor, se pueden seleccionar una para que realice el cambio del medidor de un suscriptor.

**FmSuscriptores:** Al abrirse se llama el método Inicializar para cargar la información de la tabla cuadrilla.



**Botones:**

* **BtnInstalar (Instalar Medidor):** Se debe copiar un nombre en la caja de texto TxNombre, seleccionar una fila de la tabla y una opción del CbTipoMedidor para realizar una nueva instalación en un nuevo suscriptor.
* **BtnVolver (Volver Inicio):** Para cerrar el form actual y abrir el form FmInicio.

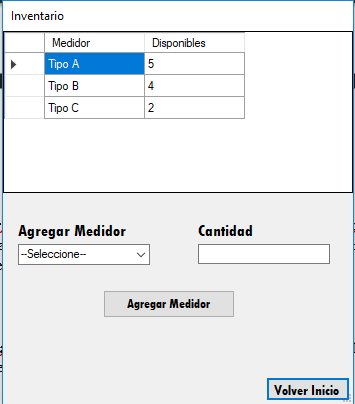
**Table:**

* **TbCuadrilla:** Para visualizar la información de la lista de la cuadrilla del GetCentro que se encuentra en el form FmInicio.

**TextBox:**

* **TxNombre:** El nombre que se asignará al nuevo suscriptor.
* **CbTipoMedidor:** Para seleccionar una opción del medidor que se desea instalar.

**FmInventario:** Cuando se abre se llama el método Inicializar para cargar la información del inventario del objeto GetCentro que esta en el form FmInicio



**Botones:**

* **BtnAgregar (Agregar Medidor):** Se selecciona la opción del medidor del CbTipoMedidor y la cantidad en el TxCantidad para agregar n medidores al inventario.
* **BtnVolver (Volver Inicio):** Para volver al form FmInicio.

**Table:**

* **TbInventario:** Para visualizar la información del inventario del objeto GetCentro.

**TextBox:**

* **TxCantidad:** Se escribe la cantidad de medidores que se desea agregar al inventario
* **CbTipoMedidor:** Se selecciona el tipo de medidor que se desea agregar al inventario.