

Tecnológico nacional de México

Alumno:

Fernandez Guzmán Juan de Dios

Cordero Hernández Jesús

Cruz Ruiz Eliuh proceso

Materia:

Programación orientada a Objetos

Clave de la materia:

2g2a

Catedrático:

Rafael Herrera García

Contenido

Ejercicio 1:.....	4
Problemática:.....	4
Análisis:	4
Diagrama Uml:	6
Creación de clase “Prendas”:.....	7
Declaración de atributos:.....	7
Declaración métodos:	11
Creación de clase “lote”:.....	14
Declaración de Atributos:	15
Declaración métodos:	17
Creación de clase “Ganancias”:	20
Declaración de Atributos:	20
Declaración métodos:	21
Relaciones clases:.....	24
Ejercicio 2:.....	25
Análisis:	25
Diagrama uml:.....	26
Creación de clase “Fruta”:	26
Declaración atributos:.....	27
Declaración métodos:	29
Creación de clase “Periodo”:	35
Declaración atributos:.....	36
Declaración métodos:	37
Relación de clases:.....	40
Ejercicio 3:.....	41
Problemática:.....	41
Análisis:	41
Diagrama Uml:	43
Creación de la clase:.....	43
Clase composición, atributos:.....	45
Creación clase observación, atributos:	47
Creación de la clase Posicion espacial:	48

OPERACIONES DE LAS CLASES :.....	50
CLASE 1:.....	50
OPERACIONES CLASE OBSERVACION:.....	55
OPERACIONES CLASE POSICION ESPACIAL:	60
RELACIÓN 1: CuerpoCeleste → Observacion.....	67

Ejercicio 1:

Problemática:

En cierta empresa de fabricación de ropa, se requiere sistematizar el proceso, se fabrican diversidad de prendas de vestir tanto para caballeros como damas, no se fabrica ropa de niños, de cada prenda se tiene un registro de modelo, tela, costo de producción por pieza, genero (masculino, femenino o mixto),

OPERACION	TIPO DE DATO	ARGUMENTO	ALCANCE	
contructor		Modelo, tela, costo, temporada, genero	publico	temporada (primavera, verano, otoño o invierno), cuando se produce un lote de cada prenda, se registra número de lote, número de piezas, fecha de fabricación.
Obtenermodelo	String		public	
Obtenertela	String		public	
Obtenercosto	float		public	
Obtenertemporada	String		public	De cada lote debe calcularse el costo de
Otenergenero	String		public	producción del lote, así como el monto de recuperación que se obtendrá después de su venta, el precio de venta por pieza es siempre 15% más del costo de producción y por lote completo es por cada pieza el 5% más del costo de producción.

Análisis:

ATRIBUTO	TIPO DE DATO	CARDINALIDAD	OBLIGATORIEDAD	
Modelo	String	1	Si	
Tela	String	1	Si	
Costo	float	1	Si	
Temporada	String	1	Si	
Genero	Genero	1	Si	

Prenda

OPERACION	TIPO DE DATO	ARGUMENTO	ALCANCE	Prenda
contructor		Modelo, tela, costo, temporada, genero	publico	
Obtenermodelo	String		public	

Obteneretela	String	public
Obtenercosto	float	public
Obtenertemporada	String	public
Otengenero	String	public

ATRIBUTO	TIPO DE DATO	CARDINALIDAD	OBLIGATORIEDAD	
Numerodelote	int	1	Si	
Numero De Prendas	int	1	Si	
Fecha De Fabricacion	LocalDate	1	Si	Lote
Costo	float	1	si	

OPERACION	TIPO DE DATO	ARGUMENTO	ALCANCE
Constructor		Numerodelote, Numerodeprendas, Fechadefabricacion, costo	publico
ObtenerNumerodelote	int		public
ObtenerNumerodePrendas	int		public
ObtenerfechadeFabricacion	LocalDate		public
ObtenerCosto	String		public

ATRIBUTO	TIPO DE DATO	CARDINALIDAD	OBLIGATORIEDAD	
CostoProduccion	float	1	Si	
Ganancia	float	1	si	Ganancia

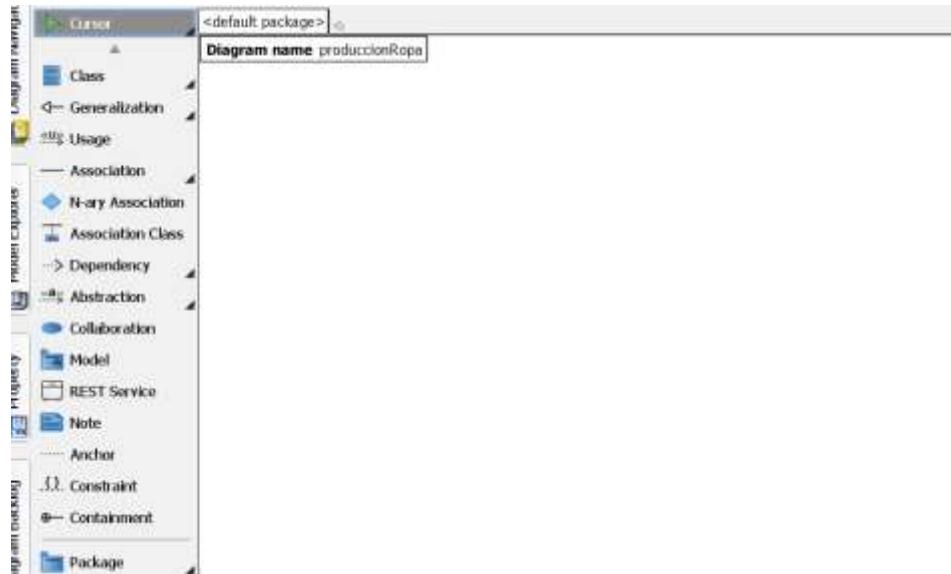
Ganancia

OPERACION	TIPO DE DATO	ARGUMENTO	ALCANCE
contructor		CostoProduccion, ganacia	publico
ObtenerCostoProduccion	float		public

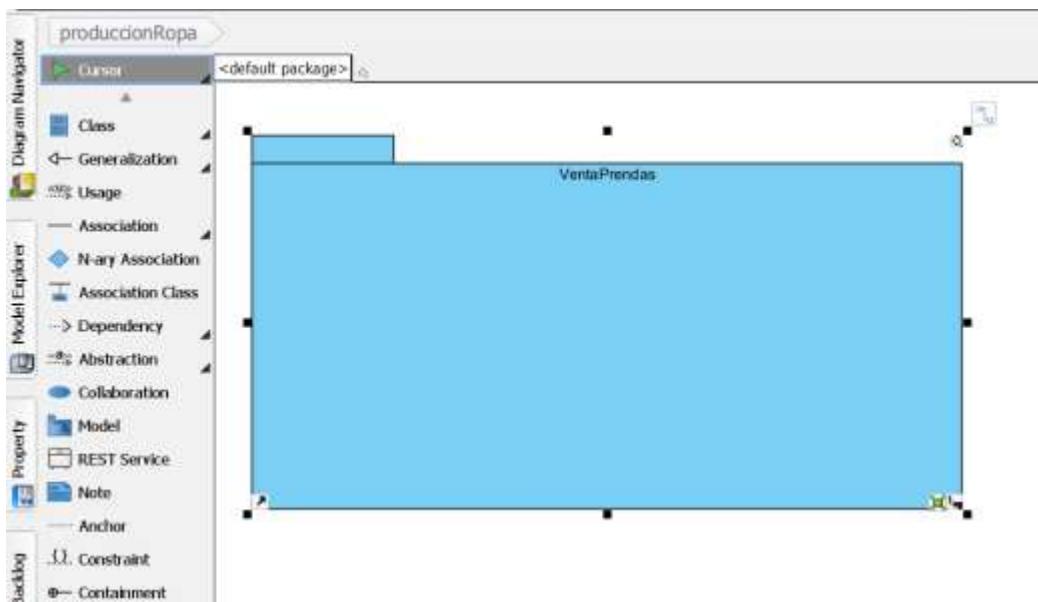
CalcularGanancia	float	CostoProduccion,numeroDePrendas public
------------------	-------	--

Diagrama Uml:

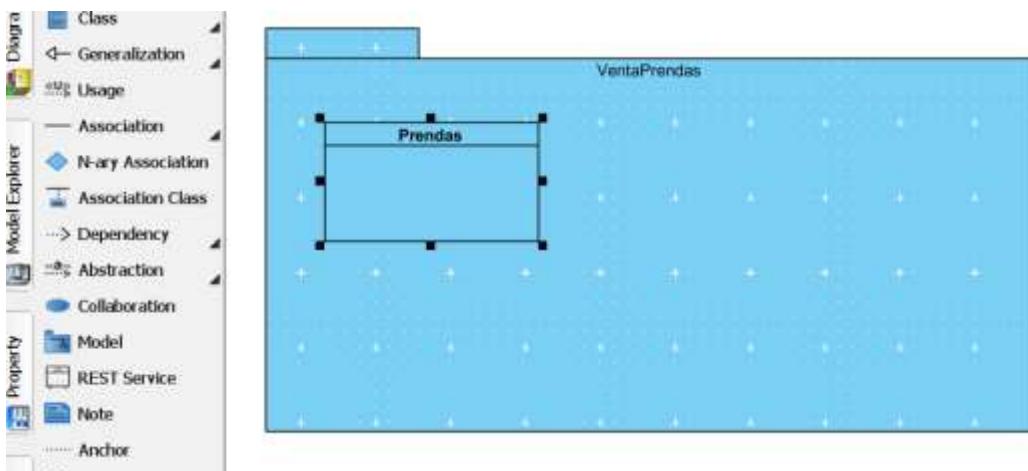
Creación de la clase:



Creación del package:



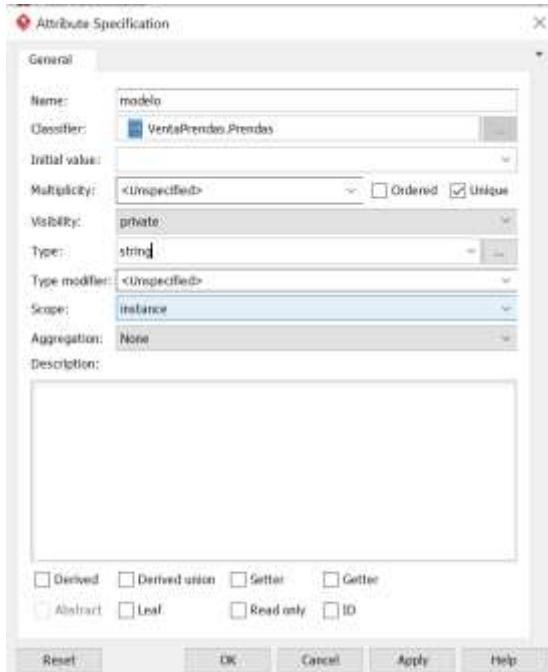
Creación de clase “Prendas”:



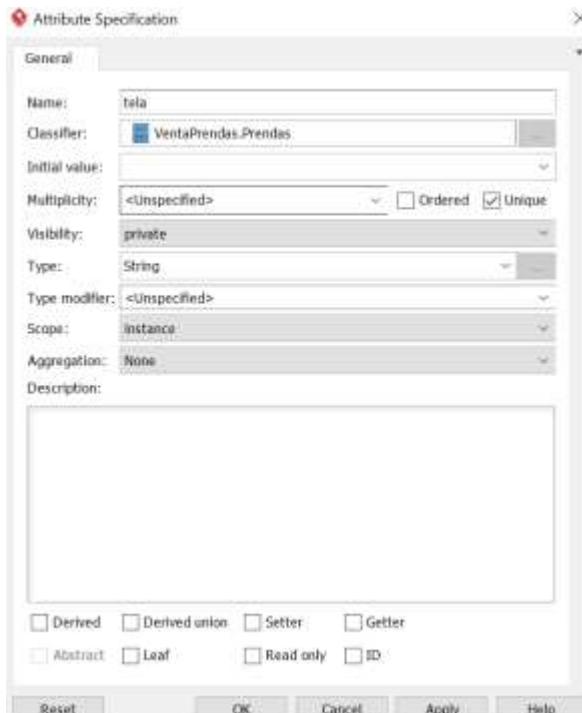
Declaración de atributos:



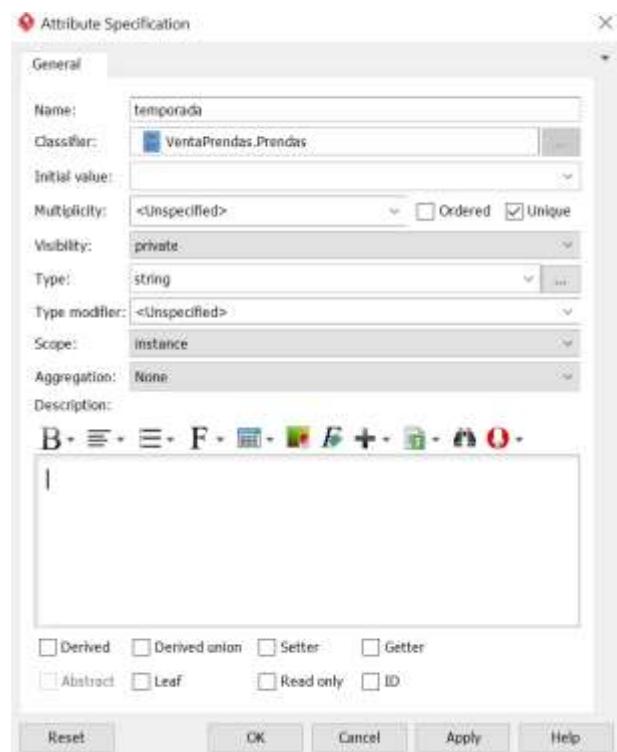
Declaración atributo “modelo”(String):



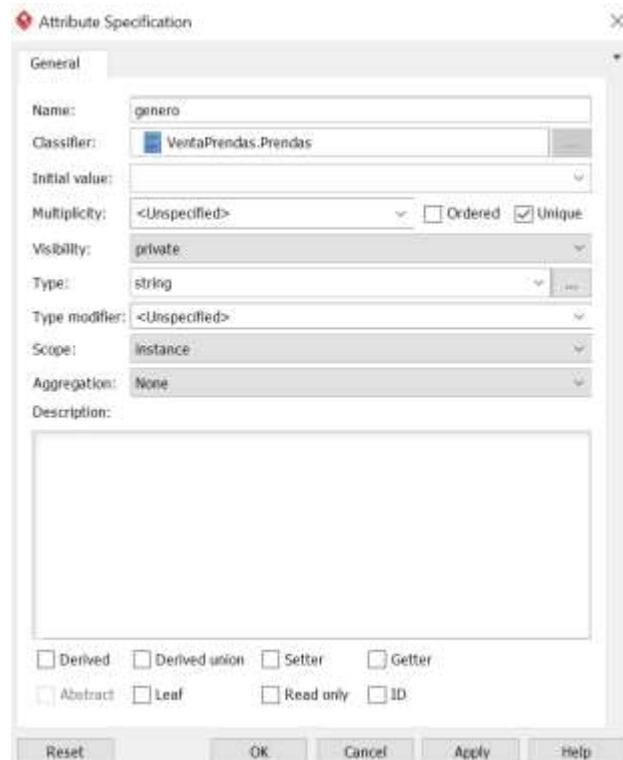
Declaración atributo “Tela”(String):



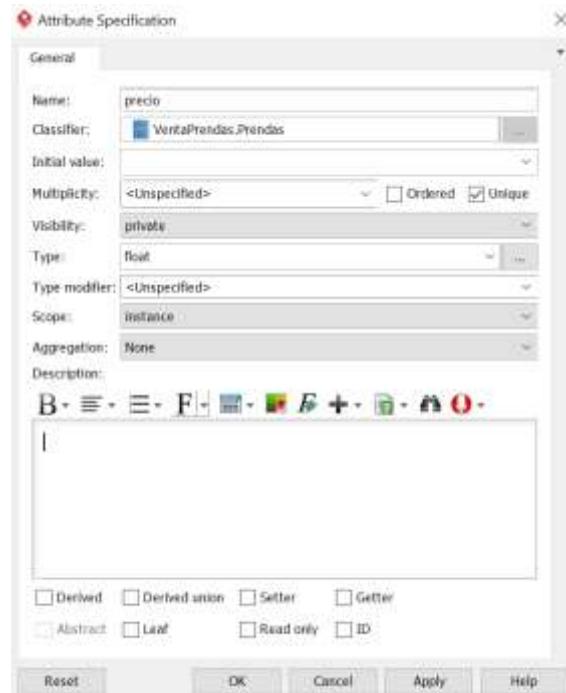
Declaración atributo “temporada”(String):



Declaración atributo “genero”(String):

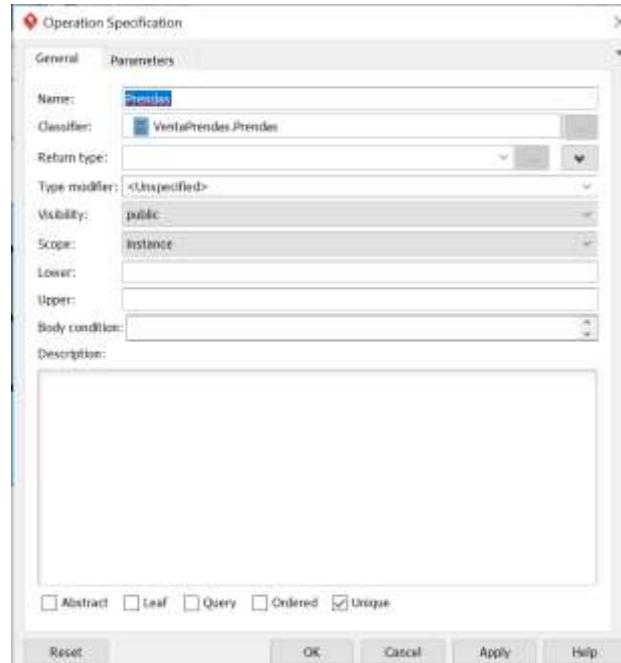


Declaración atributo “precio”(float):

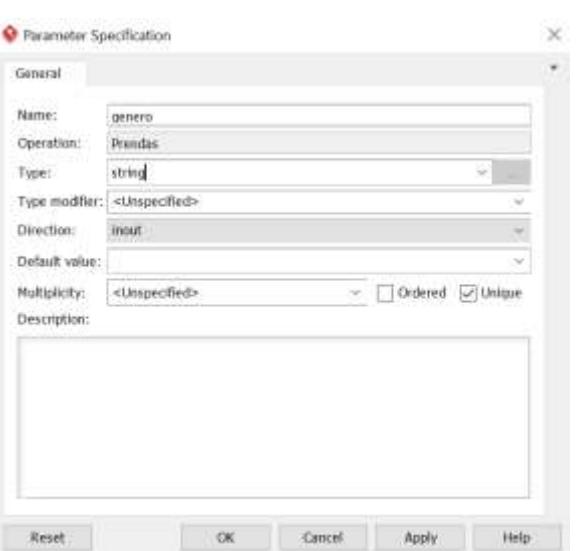
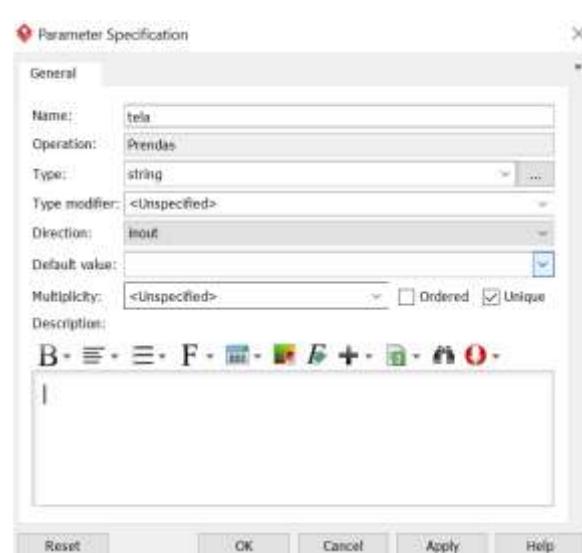


Declaración métodos:

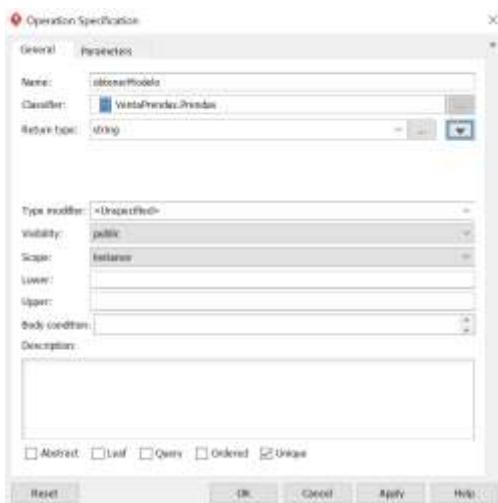
Declaración constructor:



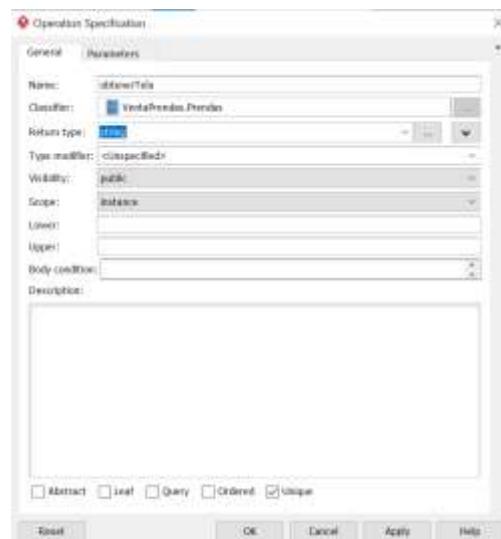
Declaración parámetros del constructor:



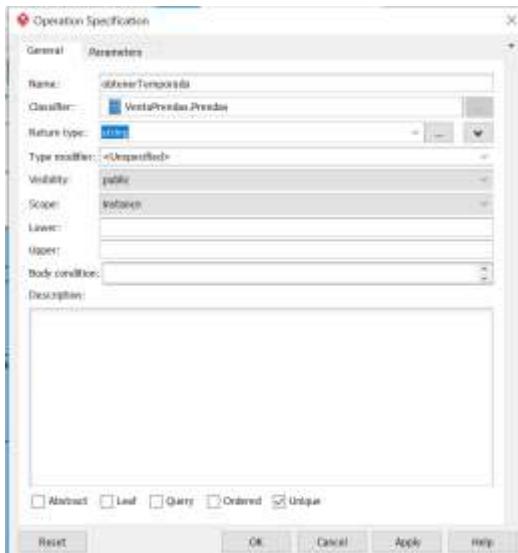
Declaración operación obtenerModelo:



Declaración operación obtenerTela:



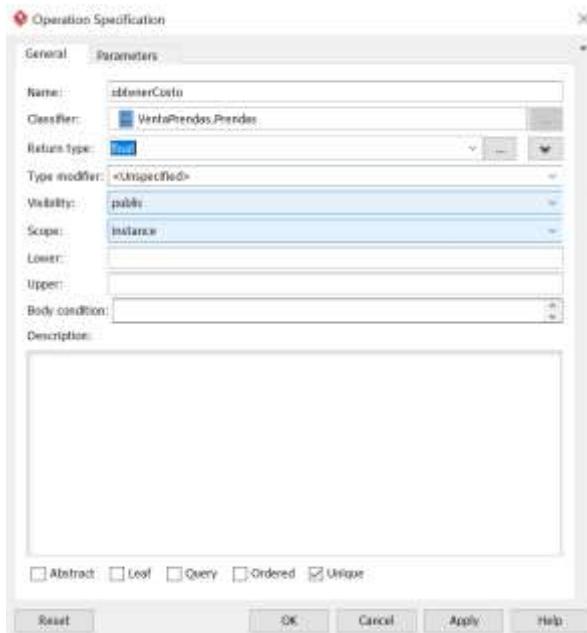
Declaración operación obtenerTemporada:



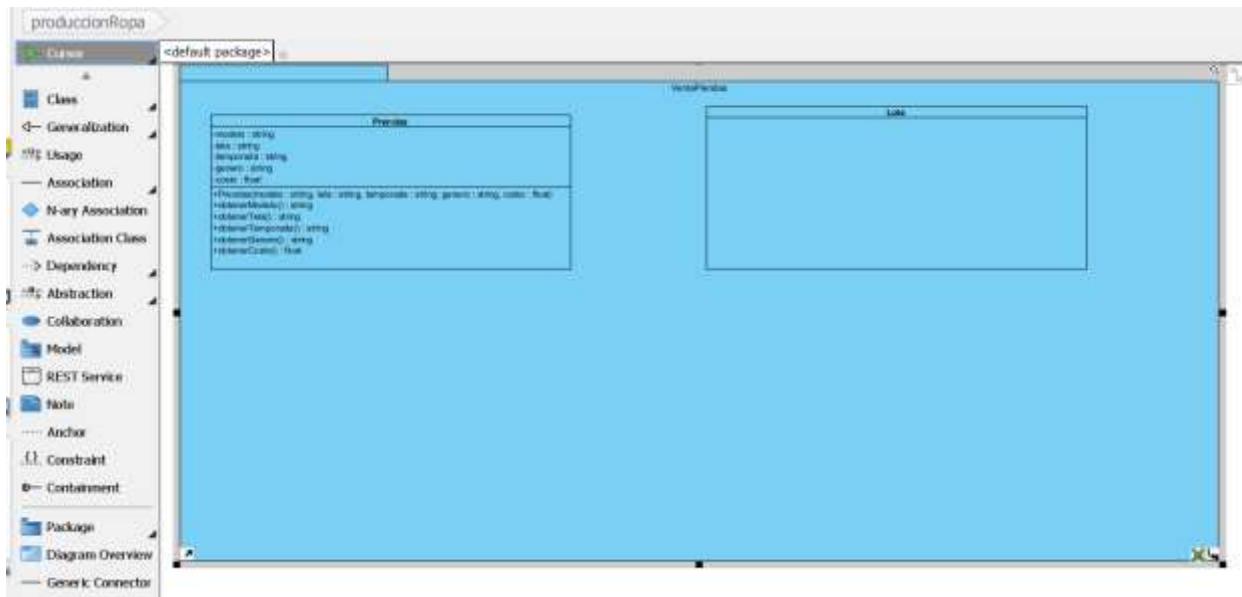
Declaración operación obtenerGenero:



Declaración operación obtenerCosto:

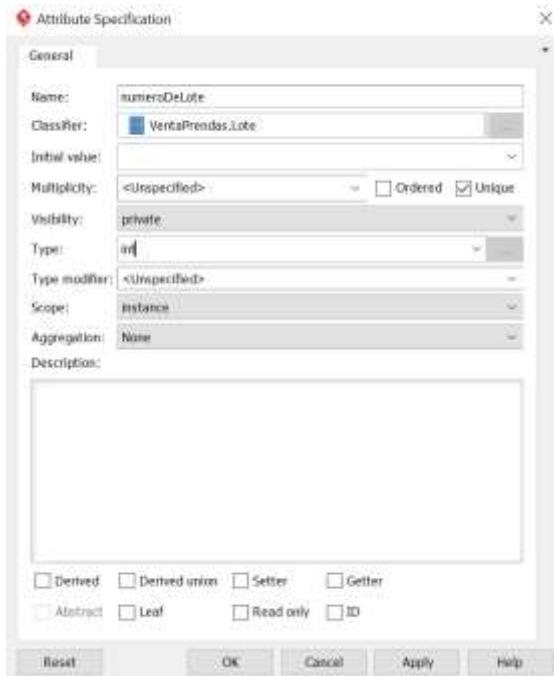


Creación de clase "lote":

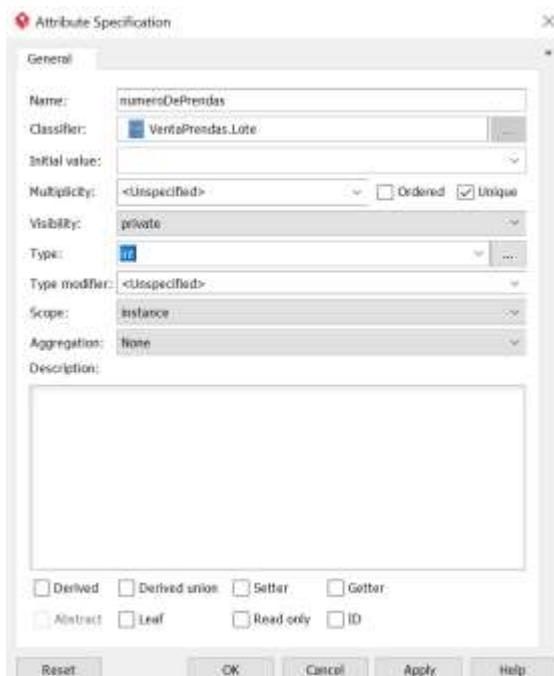


Declaración de Atributos:

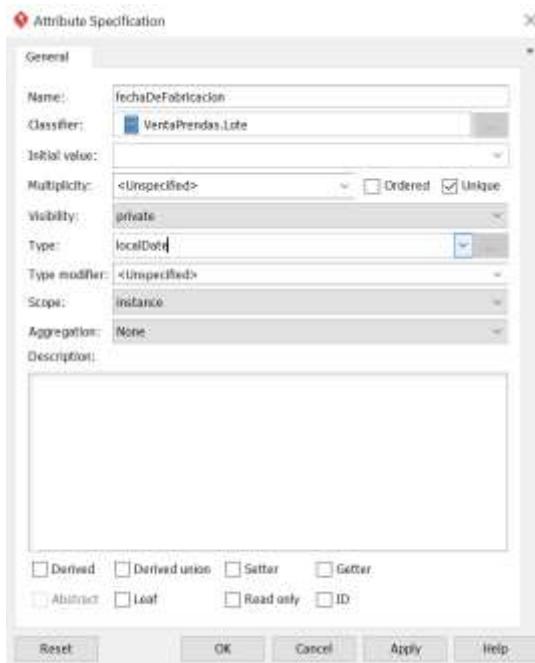
Declaración atributo “numeroDeLote”(int):



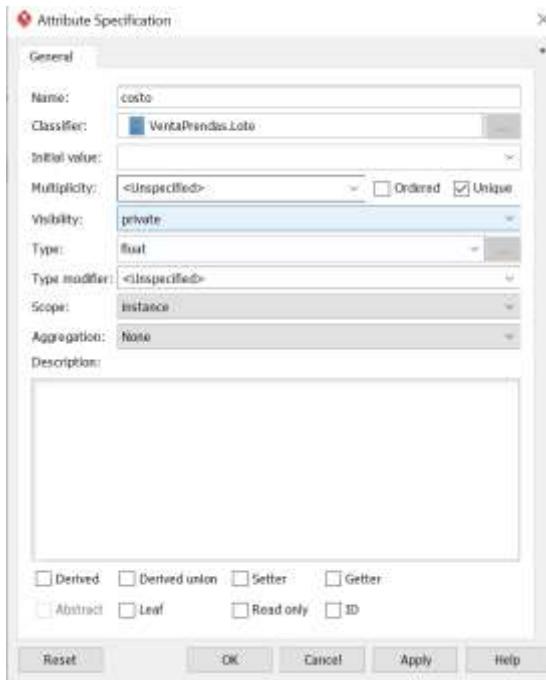
Declaración atributo “numeroDePrendas”(int):



Declaración atributo “fechaDeFabricacion”(localDate):



Declaración atributo “costo”(float):

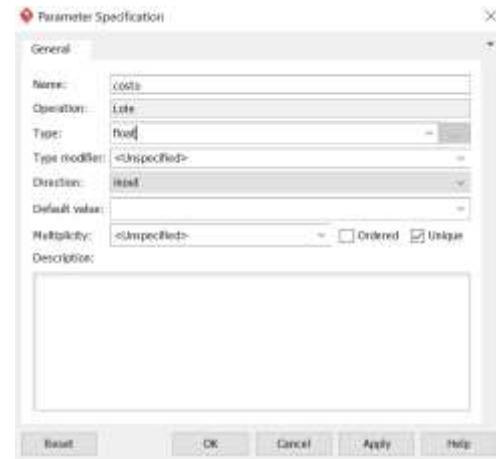
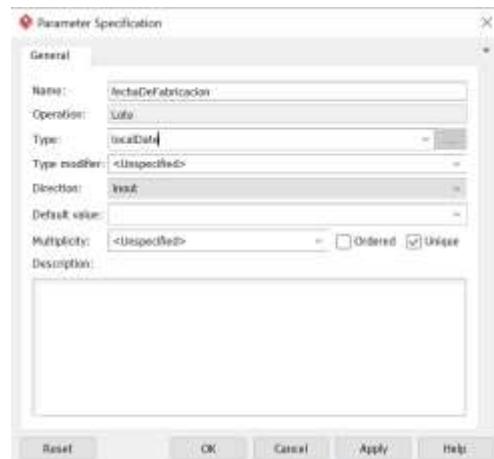
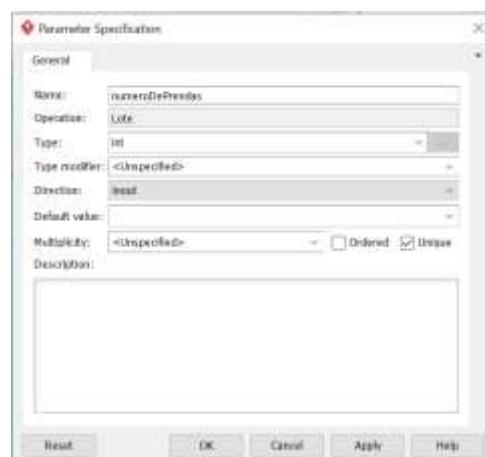
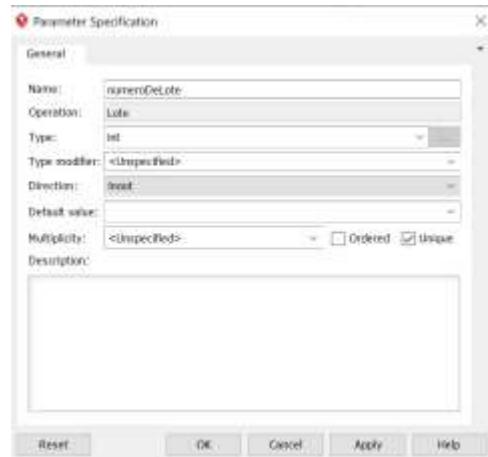


Declaración métodos:

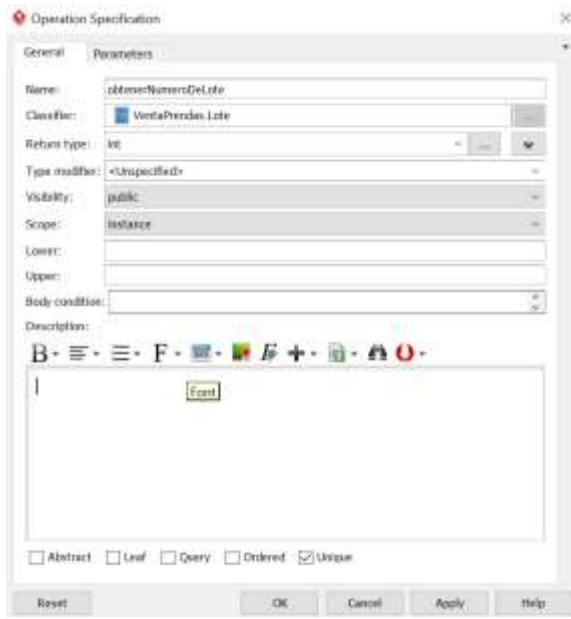
Declaración de constructor:



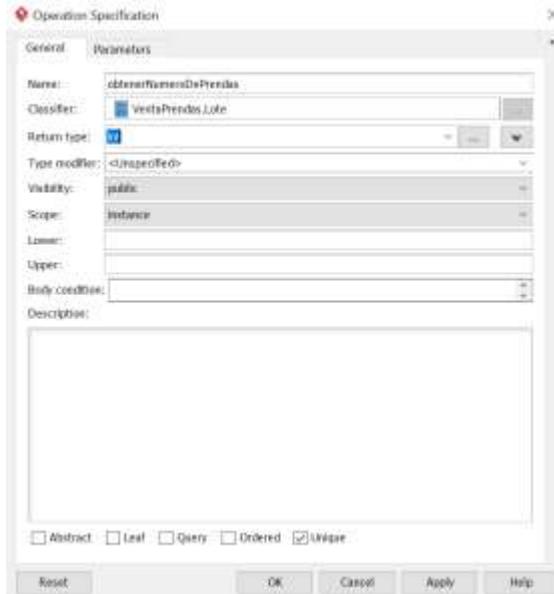
Declaración parámetros constructor:



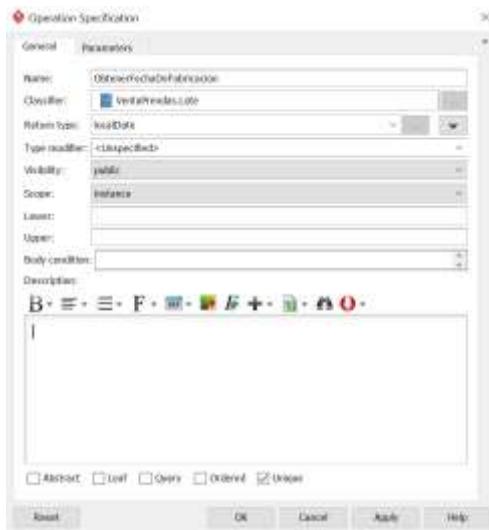
Declaración operación obtenerNumeroDeLote:



Declaracion operación obtenerNumeroDePrendas:



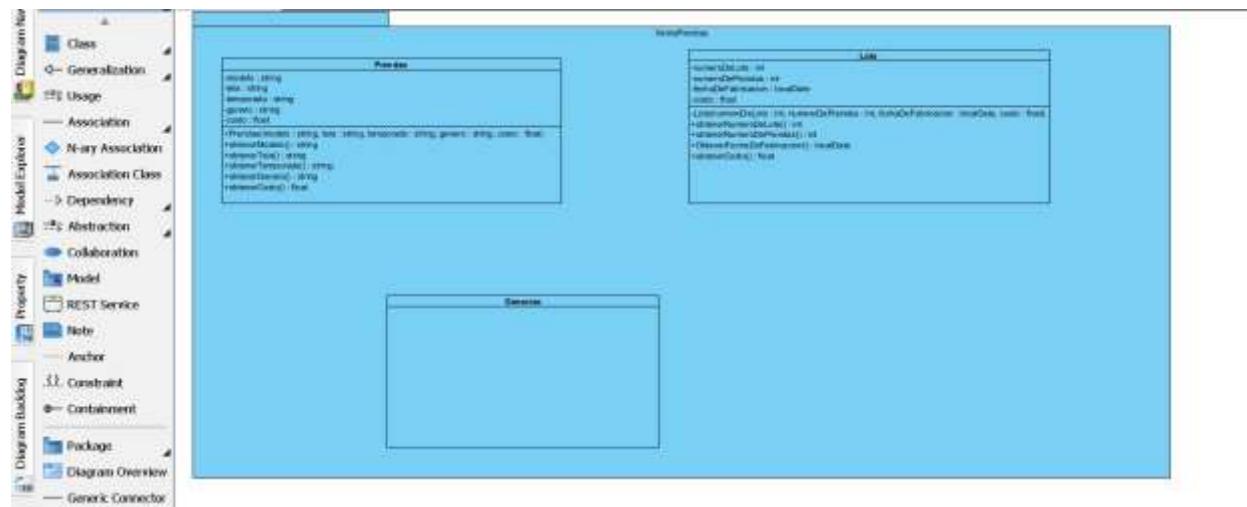
Declaración operación obtenerFechaDeFabricacion:



Declaración operación obtenerCosto:

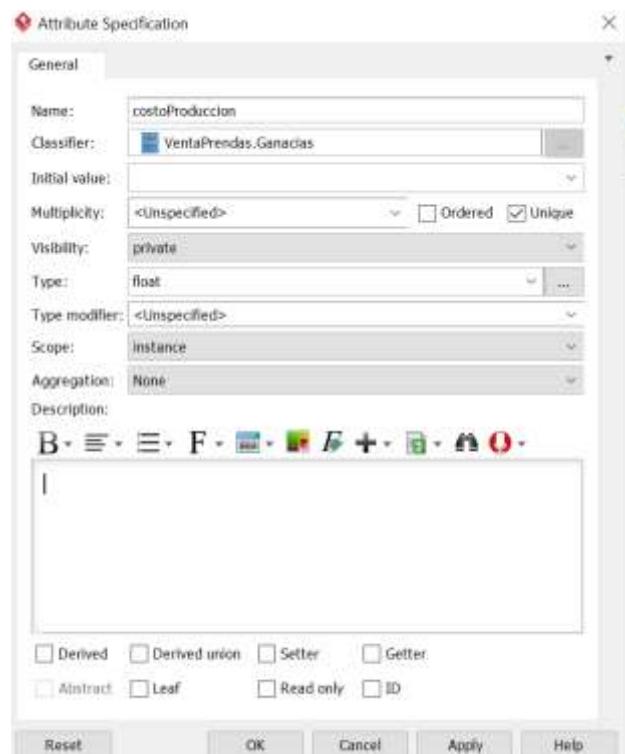


Creación de clase “Ganancias”:

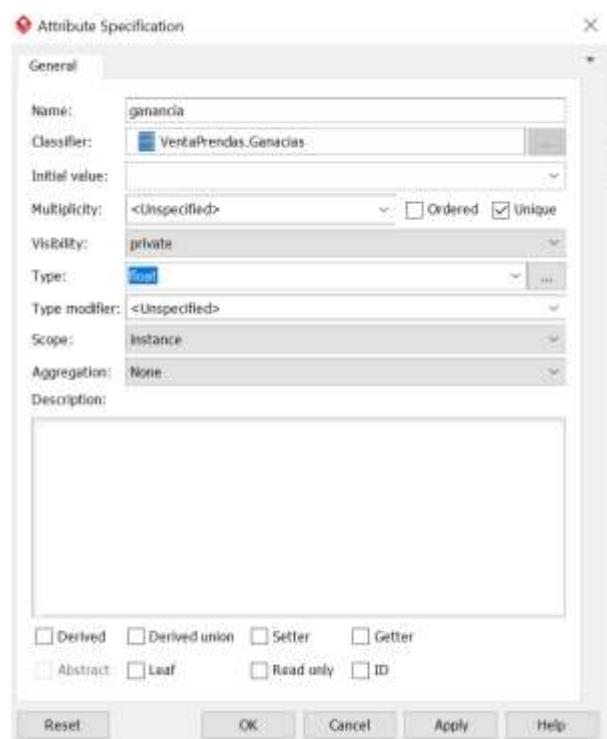


Declaración de Atributos:

Declaración atributo “costoProduccion”(float):



Declaración atributo “ganancia”(float):

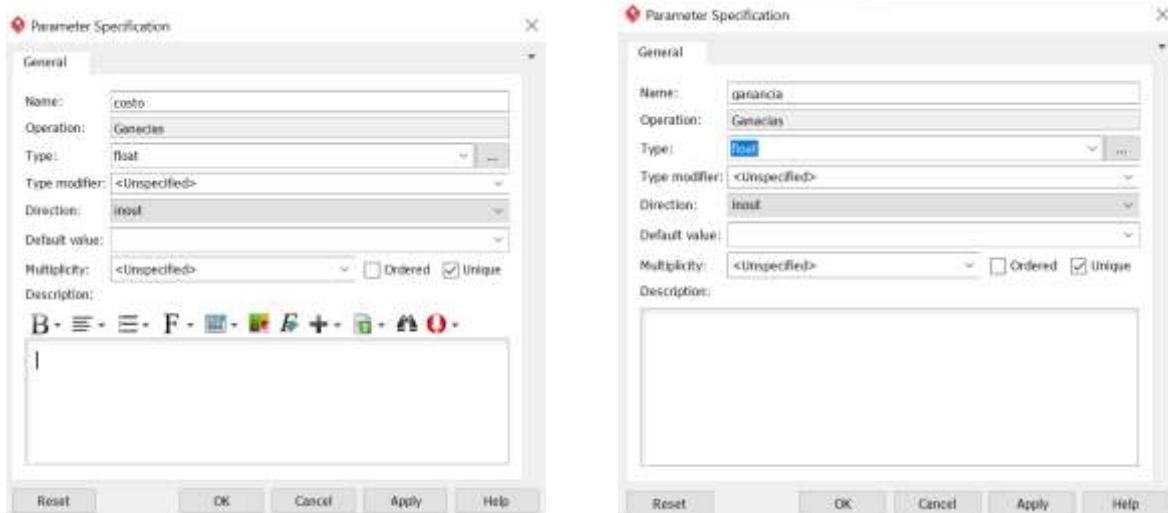


Declaración métodos:

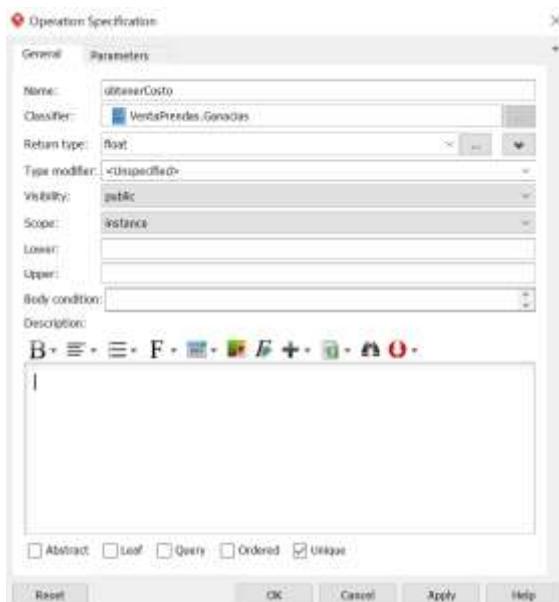
Declaración método constructor:



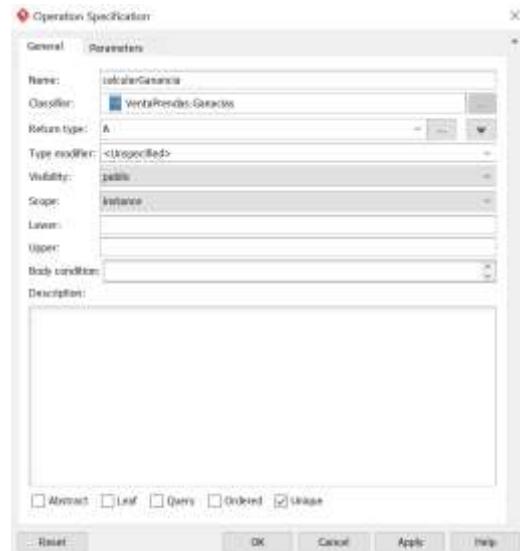
Declaracion parámetros constructor:



Declaración método obtenerCosto:



Declaración metodo calcularGanancia:



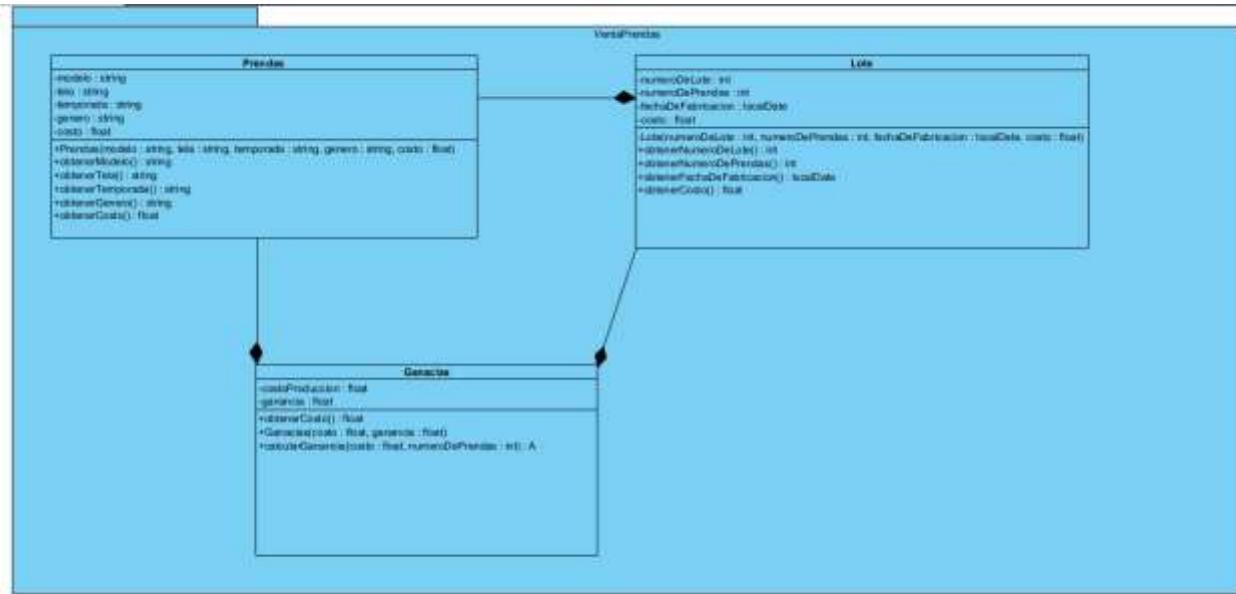
Parámetros calcular ganancia:

Relaciones clases:

Para el primer caso costo es completamente dependiente de las clases "Prendas" y "Lote" ya que sin esta no tendría caso como calcular la ganancia de ambas.

Por lo cual elegiremos la opción de composición ya que es completamente dependiente del otro;

Para el segundo caso el lote es una extensión de varias prendas por lo cual seria inútil que hubiera un lote sin prendas.



Ejercicio 2:

Una empresa agrícola ha decidido sistematizar el control de la producción frutal que tiene, para lo cual de cada fruta se describe su nombre, la extensión del terreno (hectáreas) dedicadas a su cultivo y producción, el tiempo de cosecha (Cada fruta puede tener 1 o más periodos de cosecha), la cantidad estimada a cosechar por hectárea en cada periodo (cada periodo por condiciones de estación climática puede tener diferentes cantidades estimadas), costo de producción promedio por tonelada y el precio de venta promedio por tonelada. Por cada fruta se puede agregar o eliminar un periodo de cosecha, se puede obtener el costo total de un periodo, así como las ganancias estimadas en un periodo.

Análisis:

ATRIBUTO	TIPO DE DATO	CARDINALIDAD	OBLIGATORIEDAD	
Nombre	String	1	Si	
AreaDelTerreno	float	1	Si	frutas
Periodo	Periodo		Si	
CostoDeProduccion	float		Si	
CostoDeVenta	float		Si	

ATRIBUTO	TIPO DE DATO	CARDINALIDAD	OBLIGATORIEDAD	
temporada	String	1	si	
cantidadPorHectarea	int	1	si	
tiempodecosecha	float	1	Si	Periodo

frutas

OPERACION	TIPO DE DATO	ARGUMENTO	ALCANCE
contructor		Nombre, areaDelTerreno, periodo, CostoDeProduccion,CostoDeVenta	publico
ObtenerNombre	String		public
ObtenerAreaDelTerreno	float	Largo,Ancho	public

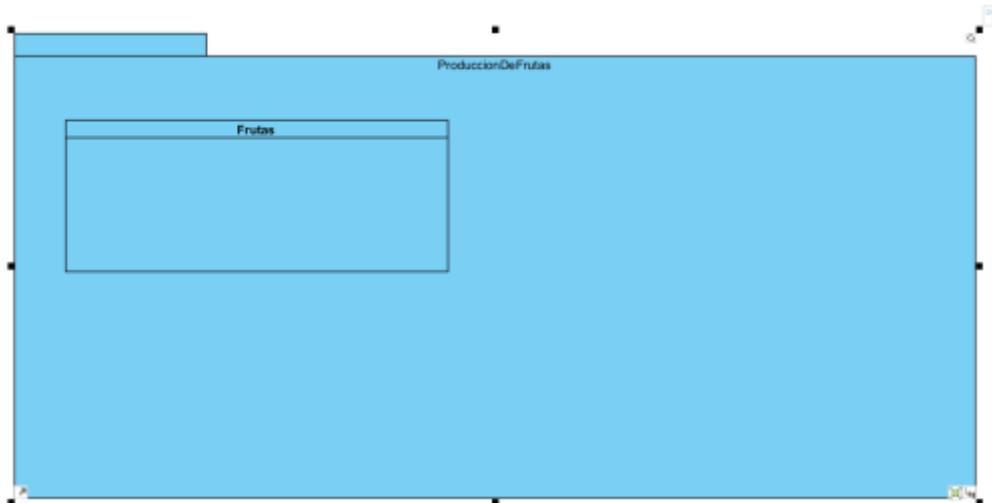
ObtenerPeriodos	Periodo[]	public
ObtenerCostoDeProduccion	float	public
ObtenerCostoDeVenta	float	public
CalcularGanancias	float	CostoDeProduccion, CostoDeVenta,Periodo
agregarPeriodo	void	public
eliminarPeriodo	void	public

Periodo

OPERACION	TIPO DE DATO	ARGUMENTO	ALCANCE
contructor		Temporada,cantidadPorHectarea,TiempoDeCosecha	publico
ObtenerTemporada	String		public
ObtenerCantidadPorHectar ea	float	Largo,Ancho	public
TiempoDeCosecha	Periodo[]		public

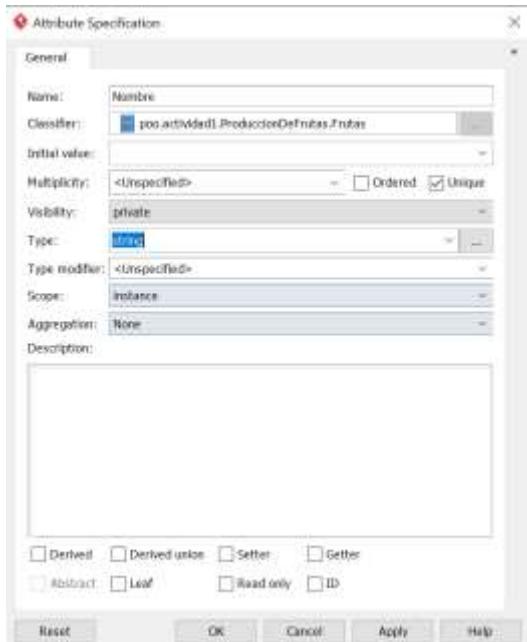
Diagrama uml:

Creación de clase “Fruta”:

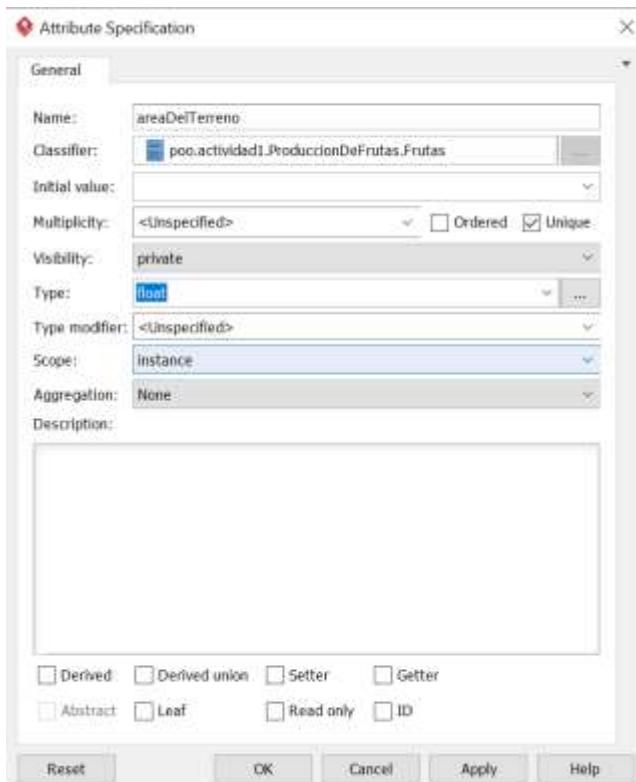


Declaración atributos:

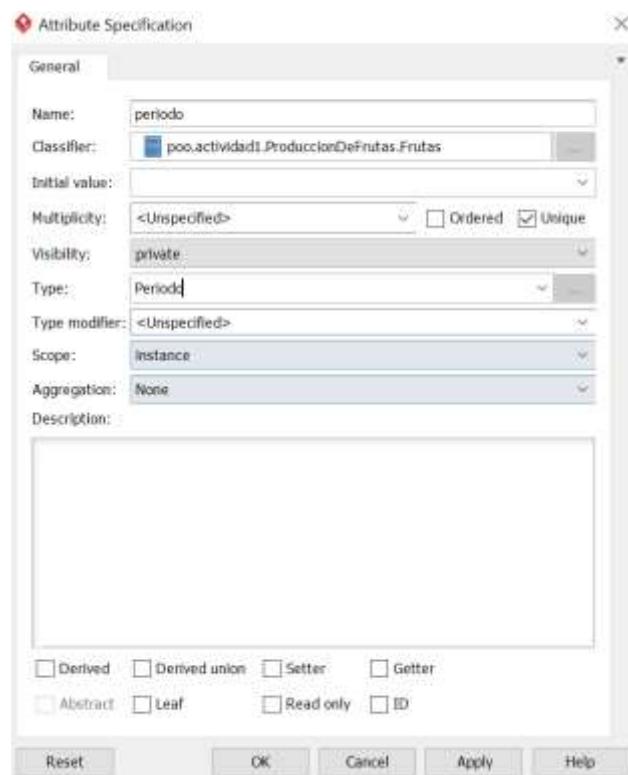
Declaración atributo “nombre” (String):



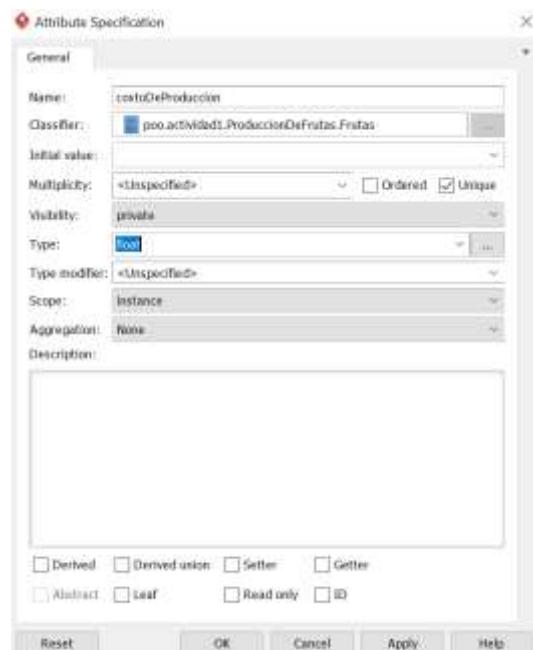
Declaración atributo “areaDelTerreno” (float):



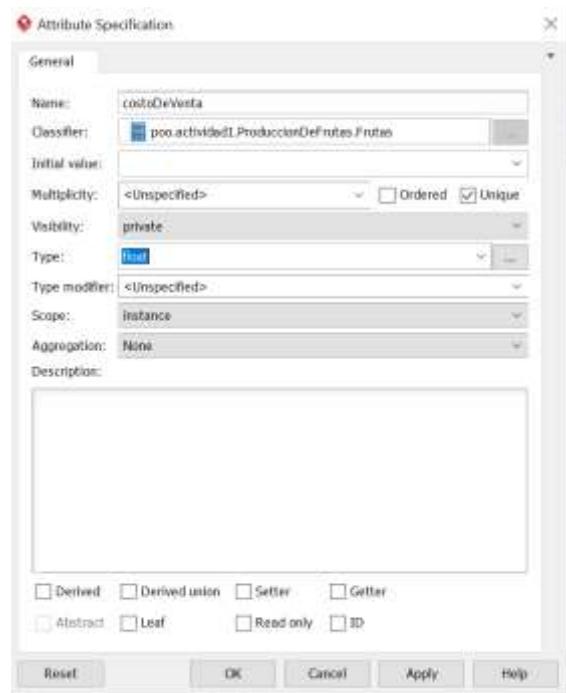
Declaración atributo “Periodo” (Periodo):



Declaración atributo “costoDeProduccion” (float):



Declaración atributo “costoDeVenta” (float):



Declaración métodos:

Declaración constructor:

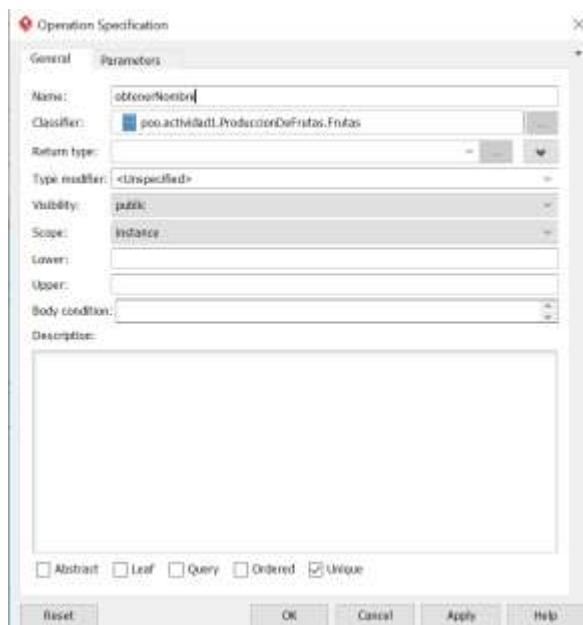


Declaración parámetros constructor:

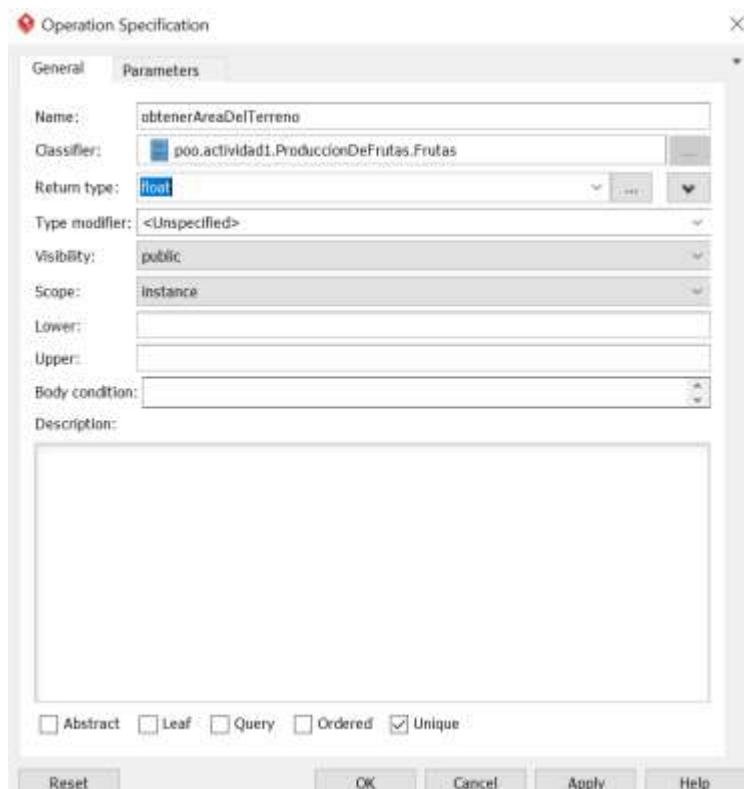
The image displays five separate instances of the 'Parameter Specification' dialog box, each with a 'General' tab selected. The parameters are as follows:

- Parameter 1:** Name: nombre, Operation: Frutas, Type: String, Type modifier: <Unspecified>, Direction: inout, Default value: (empty), Multiplicity: <Unspecified>, Description: (empty). Checkboxes: Ordered (unchecked), Unique (checked).
- Parameter 2:** Name: areaDelTerreno, Operation: Frutas, Type: float, Type modifier: <Unspecified>, Direction: inout, Default value: (empty), Multiplicity: <Unspecified>, Description: (empty). Checkboxes: Ordered (unchecked), Unique (checked).
- Parameter 3:** Name: periodo, Operation: Frutas, Type: Periodo, Type modifier: <Unspecified>, Direction: inout, Default value: (empty), Multiplicity: <Unspecified>, Description: (empty). Checkboxes: Ordered (unchecked), Unique (checked).
- Parameter 4:** Name: costosDeProduccion, Operation: Frutas, Type: float, Type modifier: <Unspecified>, Direction: inout, Default value: (empty), Multiplicity: <Unspecified>, Description: (empty). Checkboxes: Ordered (unchecked), Unique (checked).
- Parameter 5:** Name: costoDeVenta, Operation: Frutas, Type: float, Type modifier: <Unspecified>, Direction: inout, Default value: (empty), Multiplicity: <Unspecified>, Description: (empty). Checkboxes: Ordered (unchecked), Unique (checked).

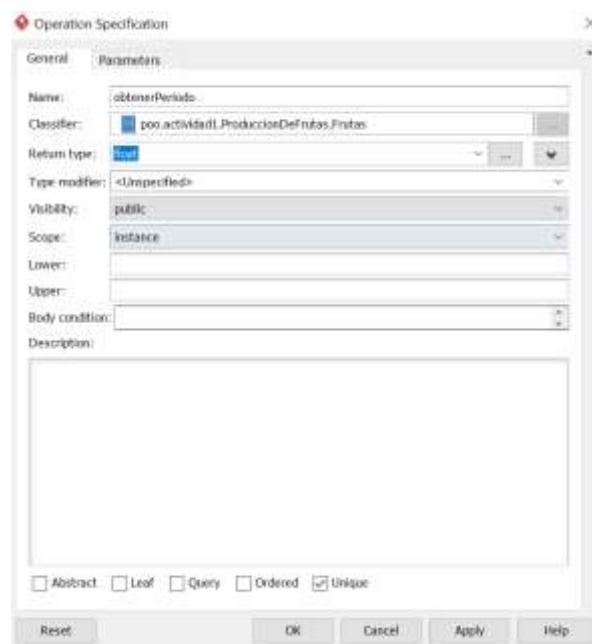
Declaración método “ObtenerNombre” (String):



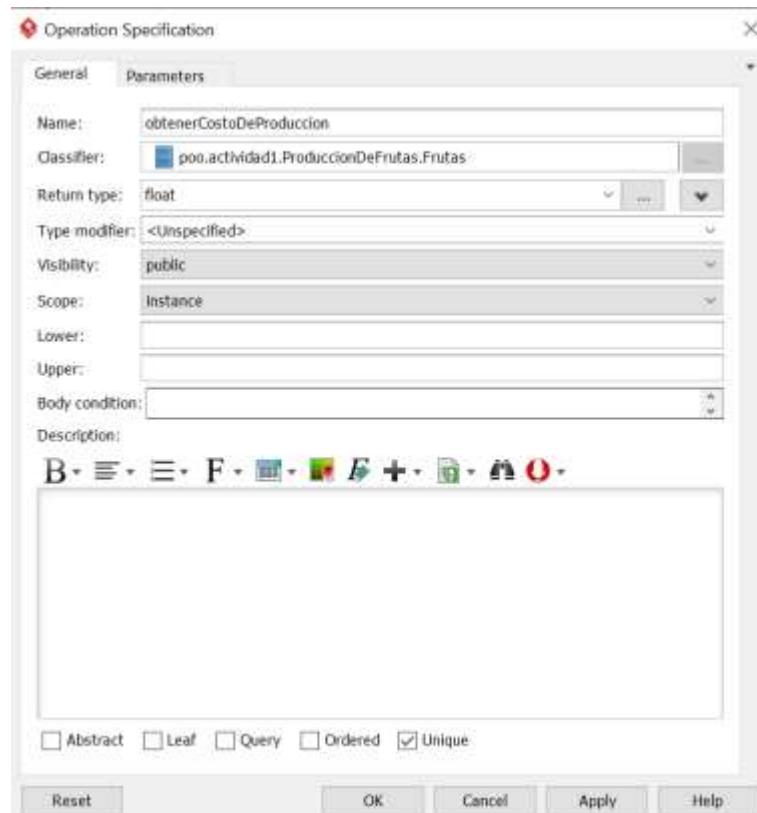
Declaración método “obtenerAreaDelTerreno” (float):



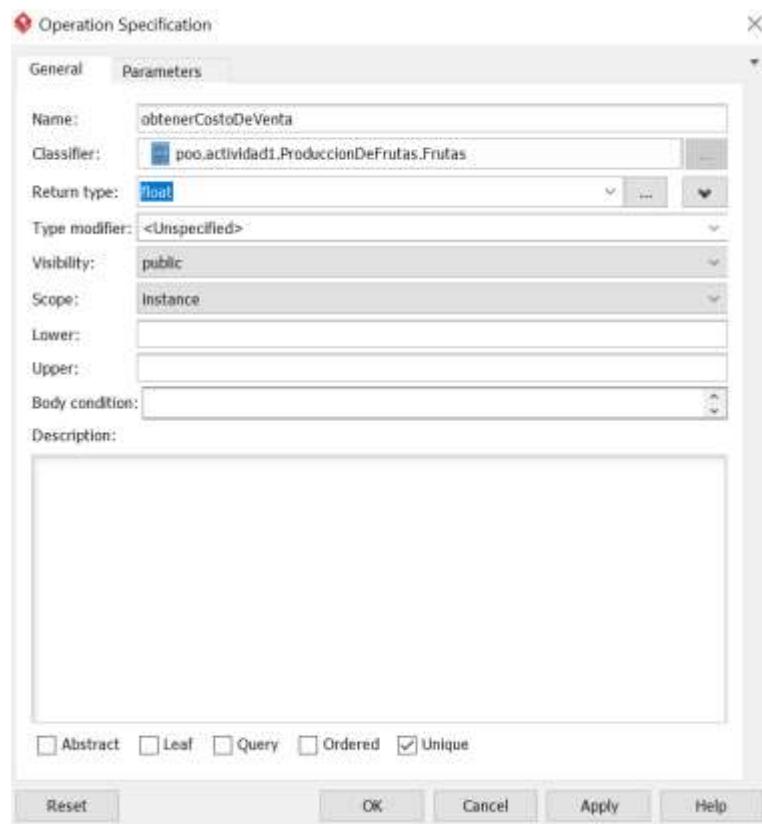
Declaración método “ObtenerPeriodo” (Periodo):



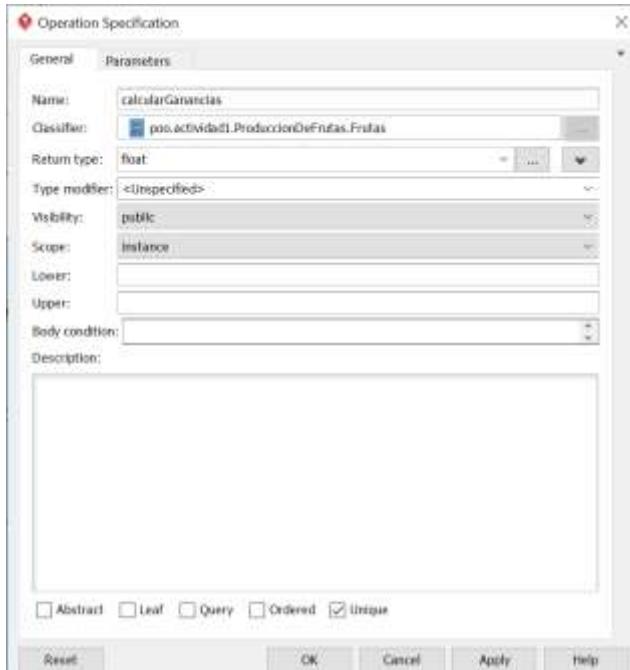
Declaración método “ObtenerCostoDeProduccion” (float):



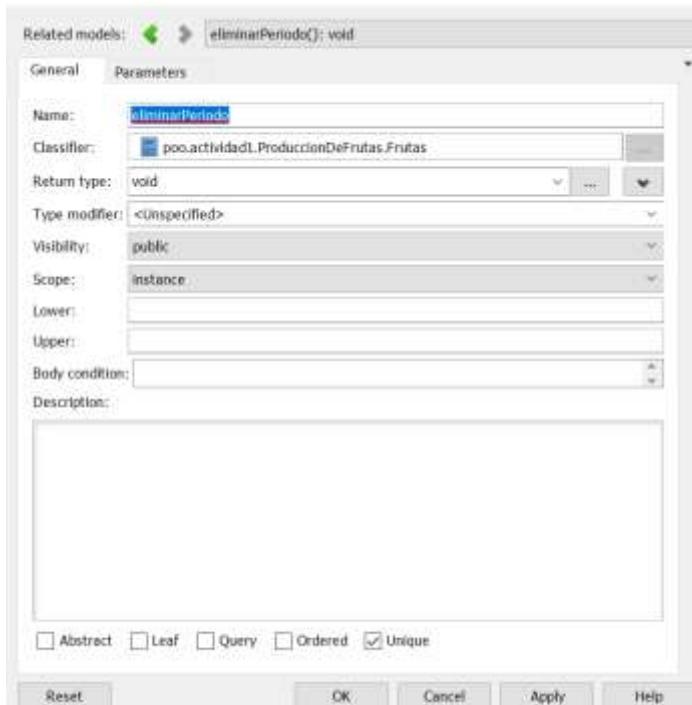
Declaración método “ObtenerCostoDeVenta” (float):



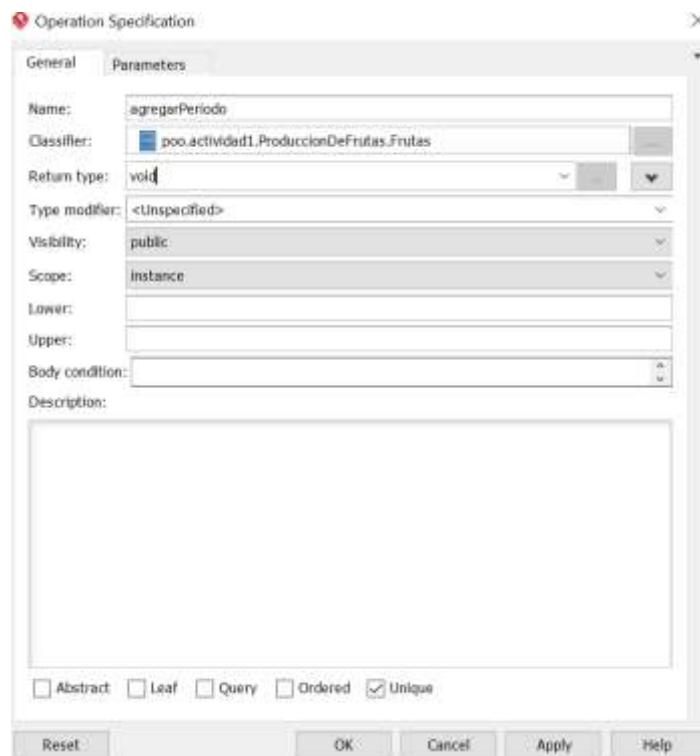
Declaración método “calcularGanancias” (float):



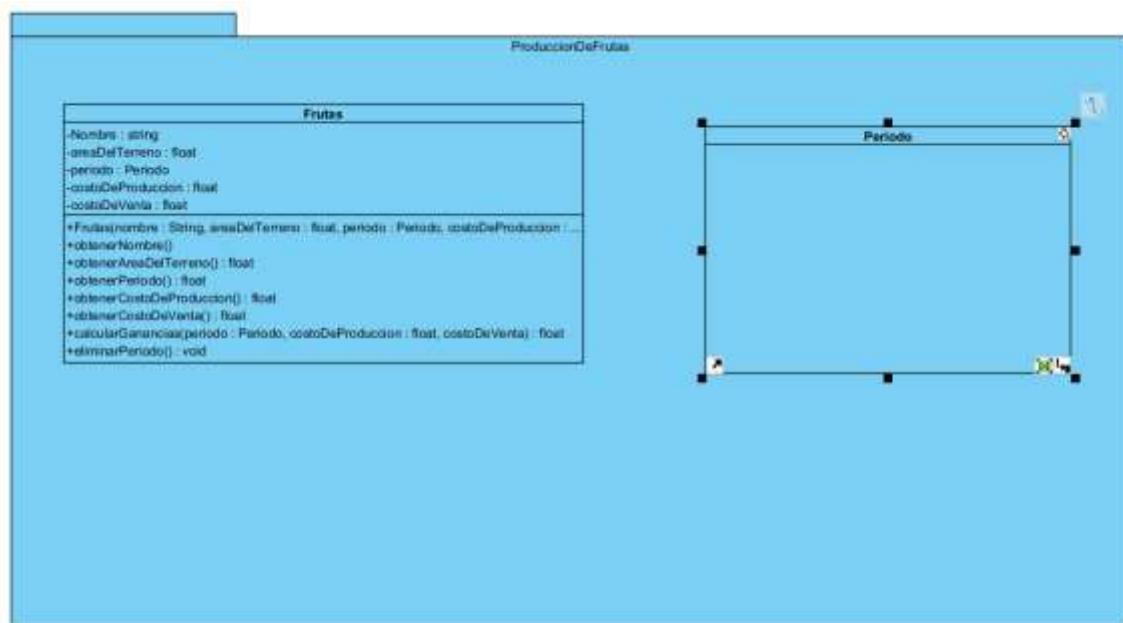
Declaración método “eliminarPeriodo” (void):



Declaración método “agregarPeriodo” (void):

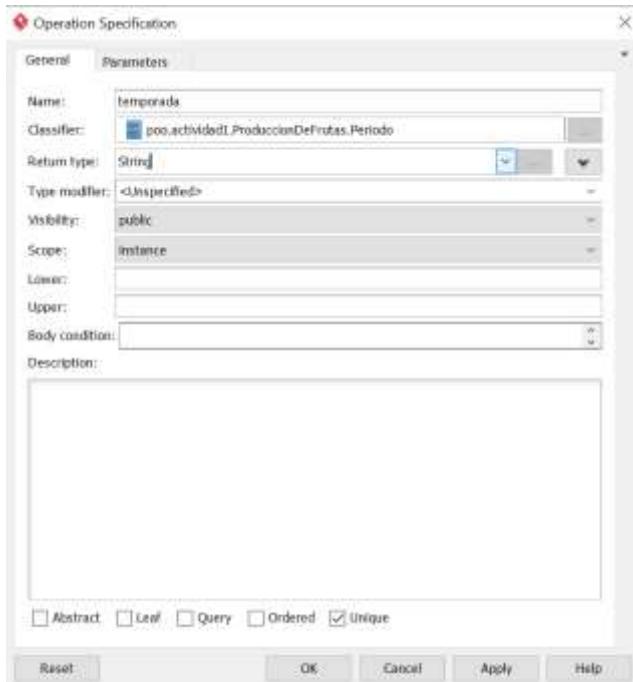


Creación de clase "Periodo":



Declaración atributos:

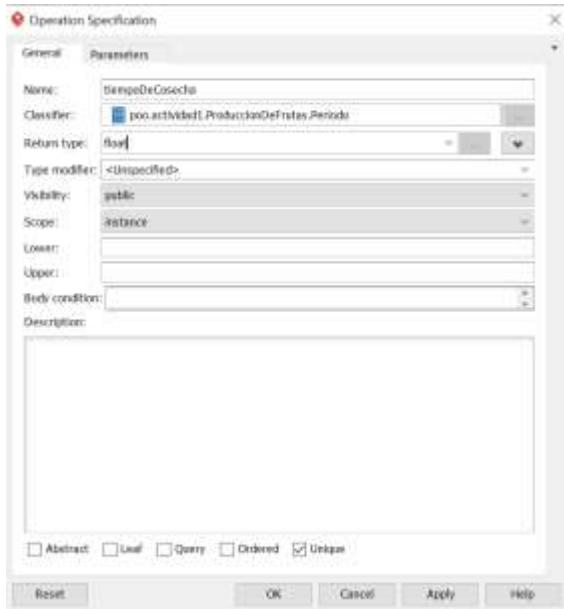
Declaración atributo “Temporada”:



Declaración atributo “cantidadPorHectarea”:

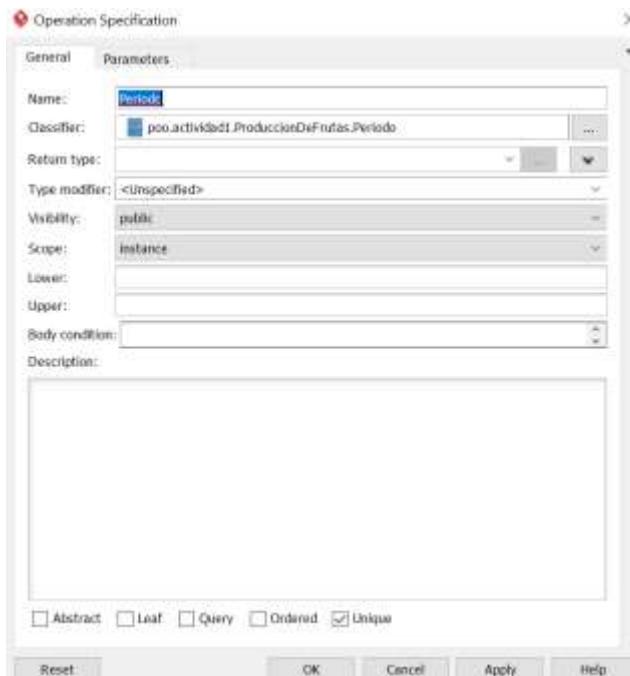


Declaración atributo “tiempoDeCosecha”:

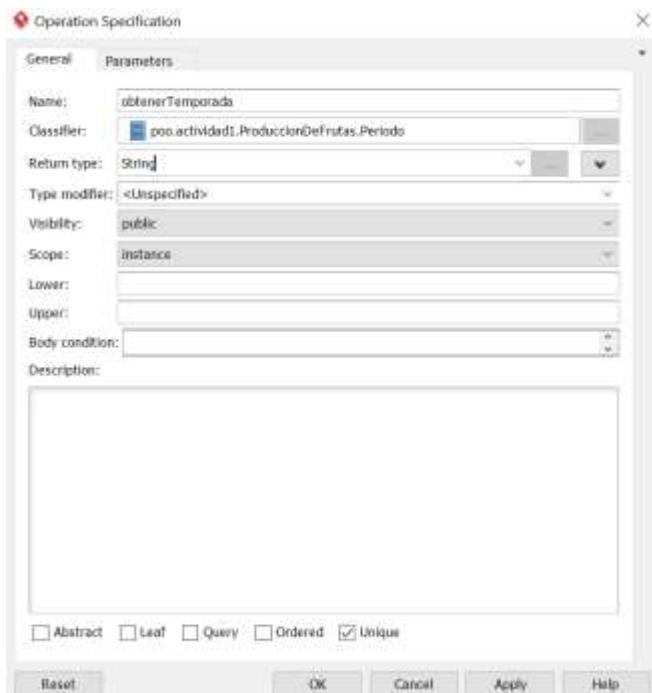


Declaración métodos:

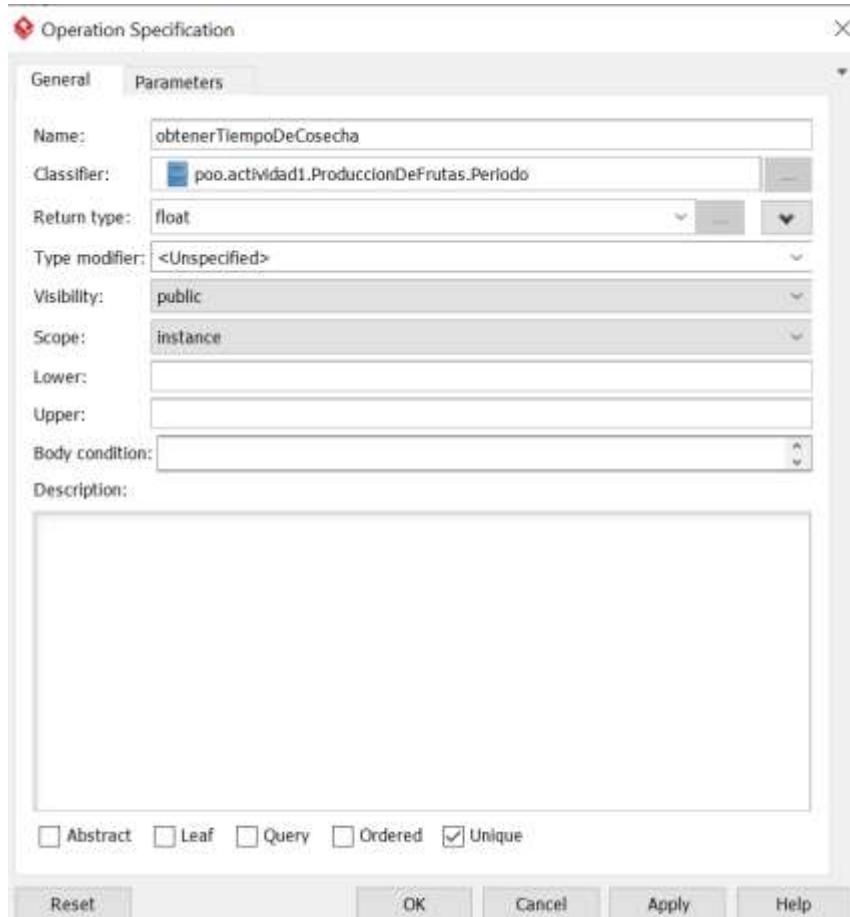
Declaración constructor:



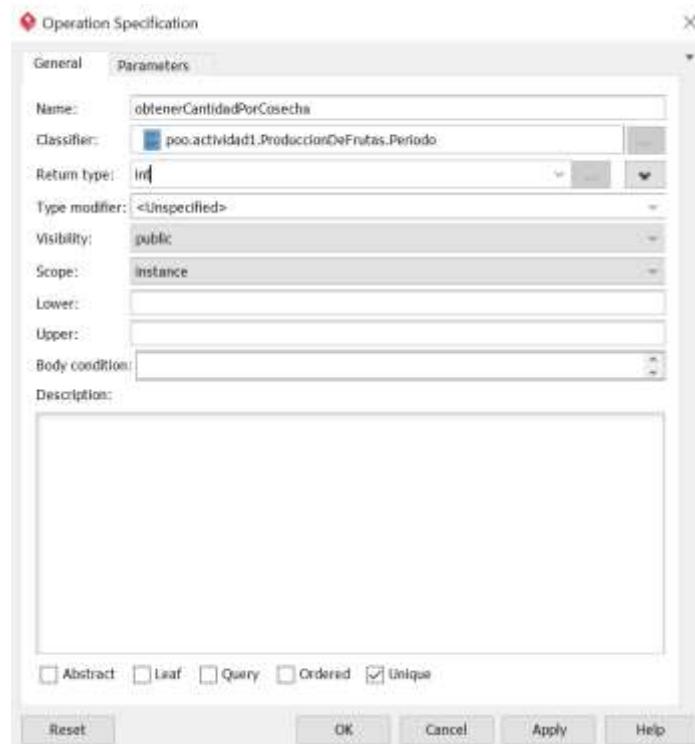
Declaración método obtenerTemporada:



Declaración método obtenerTiempoDeCosecha:

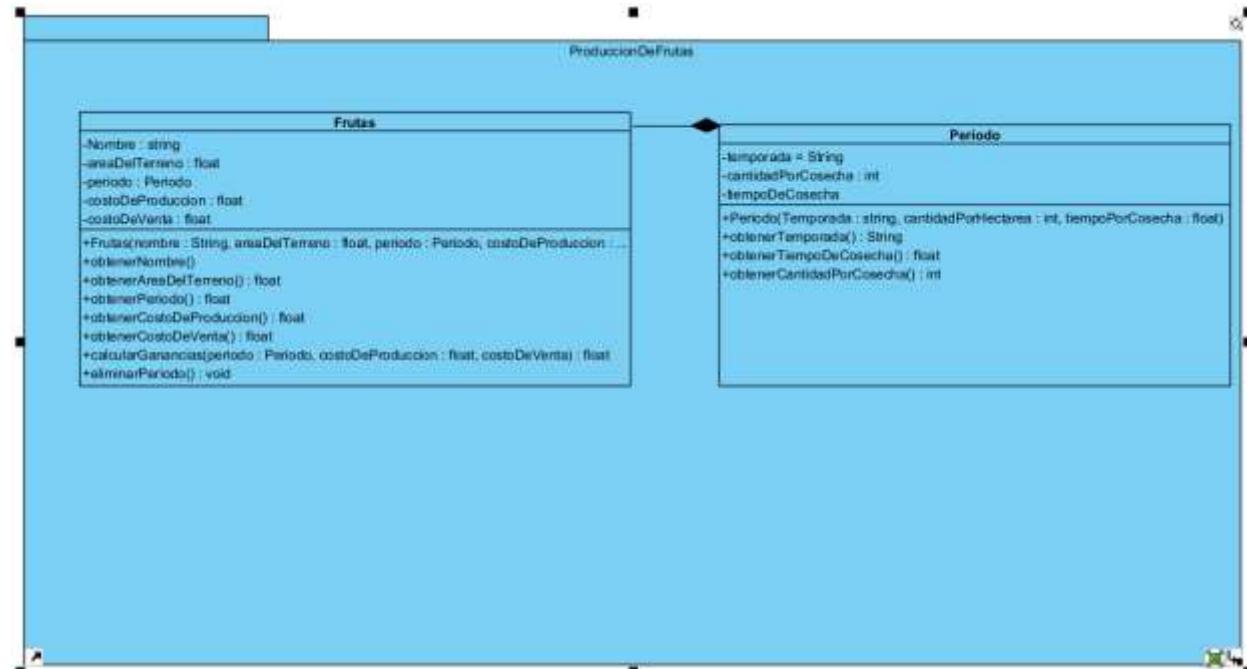


Declaración método obtenerCantidadPorCosecha:



Relación de clases:

Es una composición ya que el periodo es completamente dependiente.



Ejercicio 3:

Problemática:

El observatorio astronómico de San Pedro Mártir, de acuerdo a sus observaciones necesita un catálogo de los cuerpos celestes descubiertos para lo cual de cada cuerpo celeste se describe la ubicación en el espacio estelar en relación a las posiciones geográficas espaciales (expresado en grado de latitud (norte o sur) y grados de longitud (este o oeste)) periodo de ubicación (mes(es) donde se observó en dicha posición), distancia aproximada a la tierra (años luz o km/segundos), el nombre que se le asignó, la composición del cuerpo estelar (gases, sólidos, líquidos). De cada cuerpo celeste se puede obtener el desplazamiento con respecto a otra observación del mismo cuerpo en un periodo distinto.

Análisis:

CuerpoCeleste

ATRIBUTO	TIPO DE DATO	CARDINALIDAD	OBLIGATORIEDAD
NOMBRE	String	1	Sí
COMPOSICION	String	1	Sí
DISTANCIAATIERRA	double	1	Sí
UNIDADDISTANCIA	String	1	Sí

Observacion

ATRIBUTO	TIPO DE DATO	CARDINALIDAD	OBLIGATORIEDAD
FECHAOBSEVACION	Date	1	Sí
PERIODOMESES	String	1	Sí

PosicionEspacial

ATRIBUTO	TIPO DE DATO	CARDINALIDAD	OBLIGATORIEDAD
LATITUDGRADOS	double	1	Sí

DIRECCIONLATITUD	String	1	Sí
LONGITUDGRADOS	double	1	Sí
DIRECCIONLONGITUD	String	1	Sí

- Operaciones para clase "CuerpoCeleste":

Operación	Tipo de dato	Argumentos	Alcance
Constructor		nombre, composicion, distanciaATierra, unidadDistancia	público
calcularDesplazamiento	double	obs1: Observacion, obs2: Observacion	público
agregarObservacion	void	observacion: Observacion	público
obtenerNombre	String		público

Operaciones para clase "Observación":

OPERACIÓN	TIPO DE DATO	ARGUMENTOS	ALCANCE
CONSTRUCTOR		fechaObservacion, periodoMeses, posicion: PosicionEspacial	público
OBTENERFECHA	Date		público
OBTENERPOSICION	PosicionEspacial		público

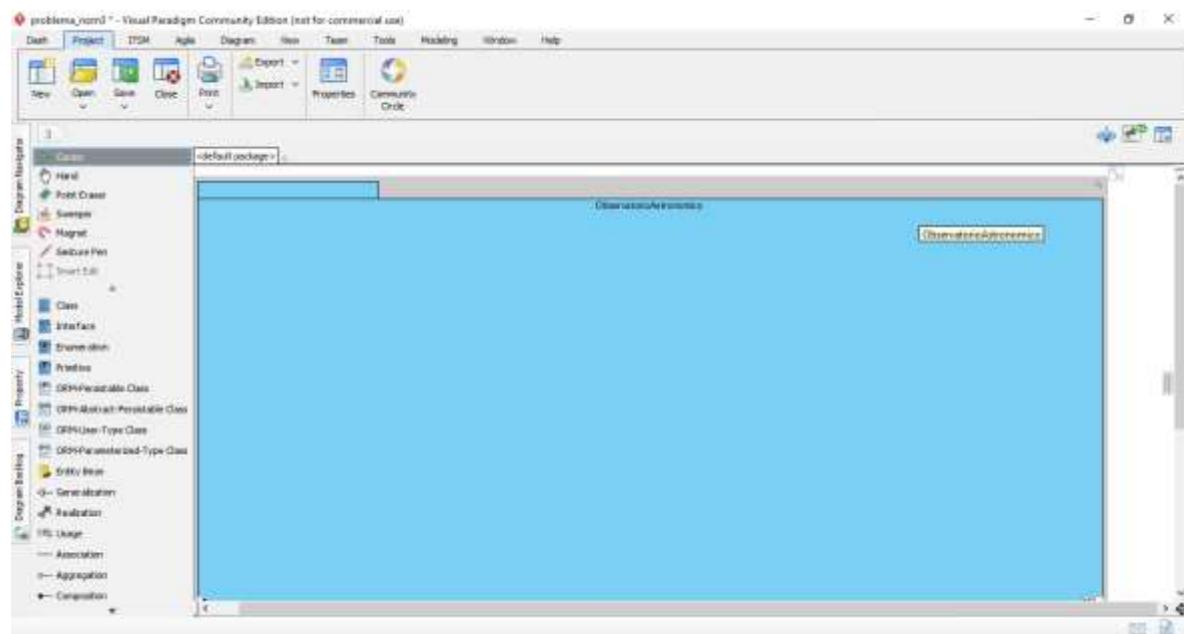
Nota Elihu: El constructor no lleva tipo de dato en la columna "Tipo de dato" porque es especial, "siempre retorna una instancia de la clase que está construyendo".

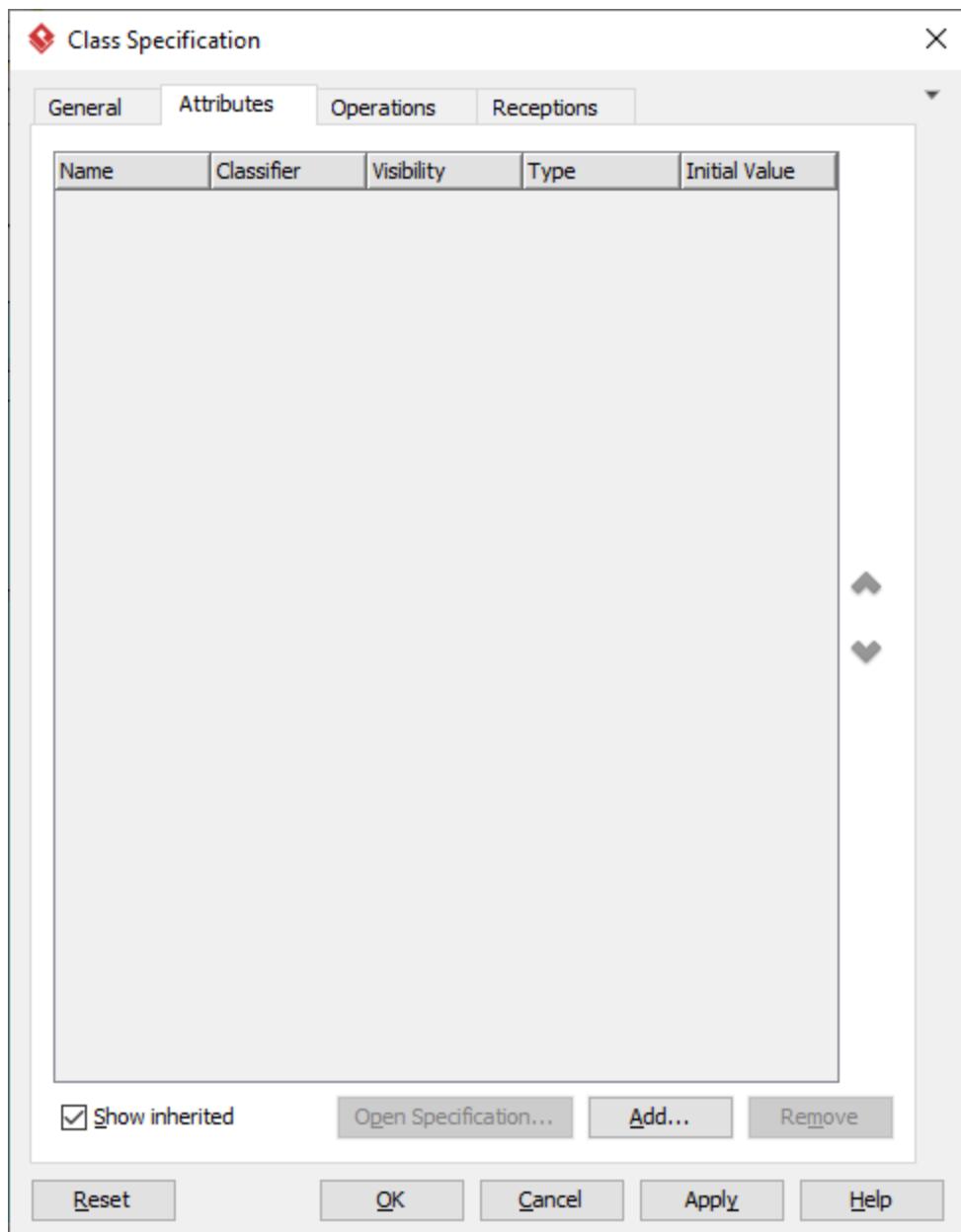
Operaciones para clase "PosicionEspacial":

OPERACIÓN	TIPO DE DATO	ARGUMENTOS	ALCANCE
CONSTRUCTOR		latitudGrados, direccionLatitud, longitudGrados, direccionLongitud	público
CALCULARDISTANCIA	double	otraPosicion: PosicionEspacial	público

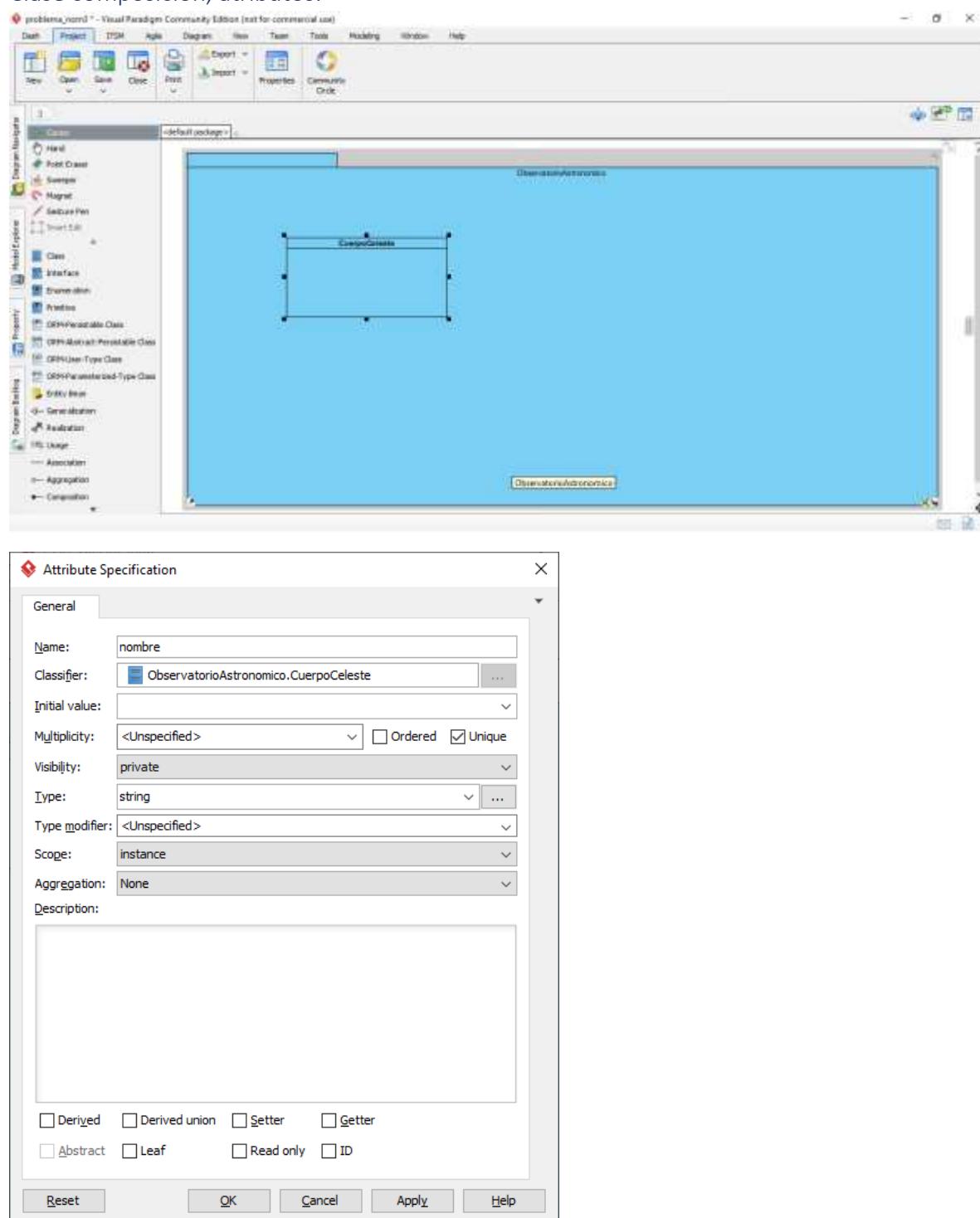
Diagrama Uml:

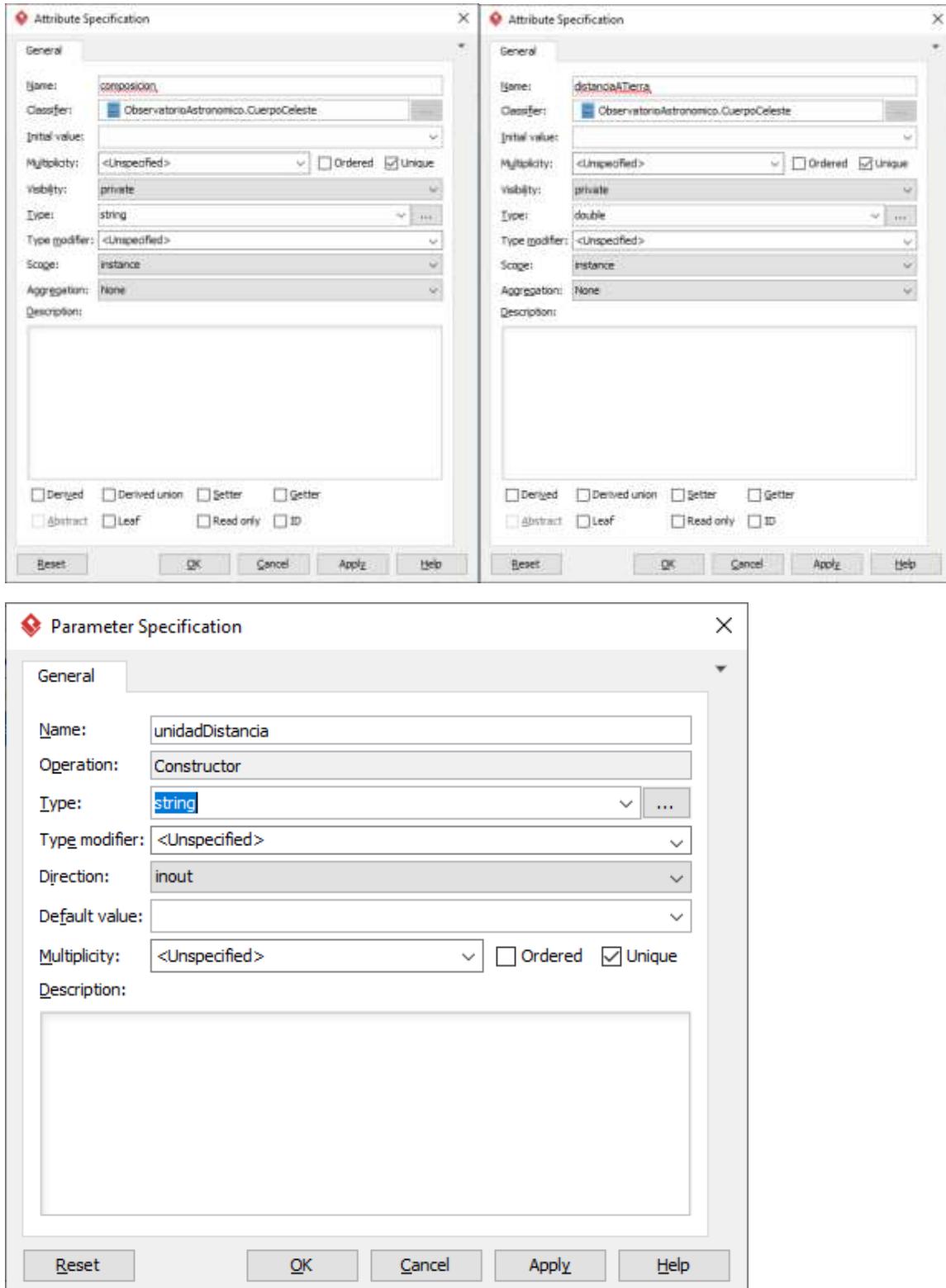
Creación de la clase:





Clase composición, atributos:





Creación clase observación, atributos:

Class Specification Dialog (Left):

- Name: Observacion
- Parent: ObservatorioAstronomico
- Visibility: public
- Description:
- Buttons: Reset, OK, Cancel, Apply, Help

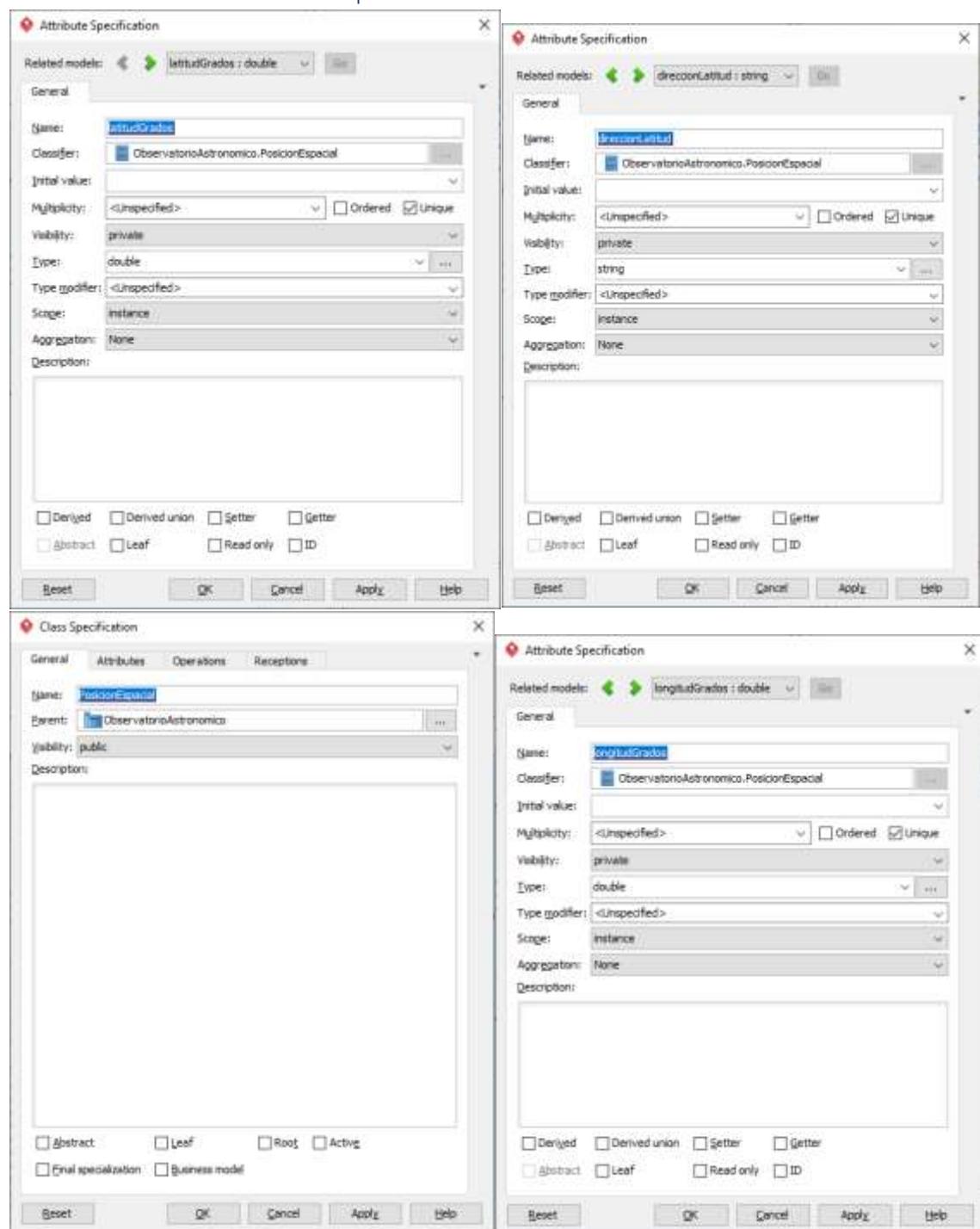
Attribute Specification Dialog (Top Right):

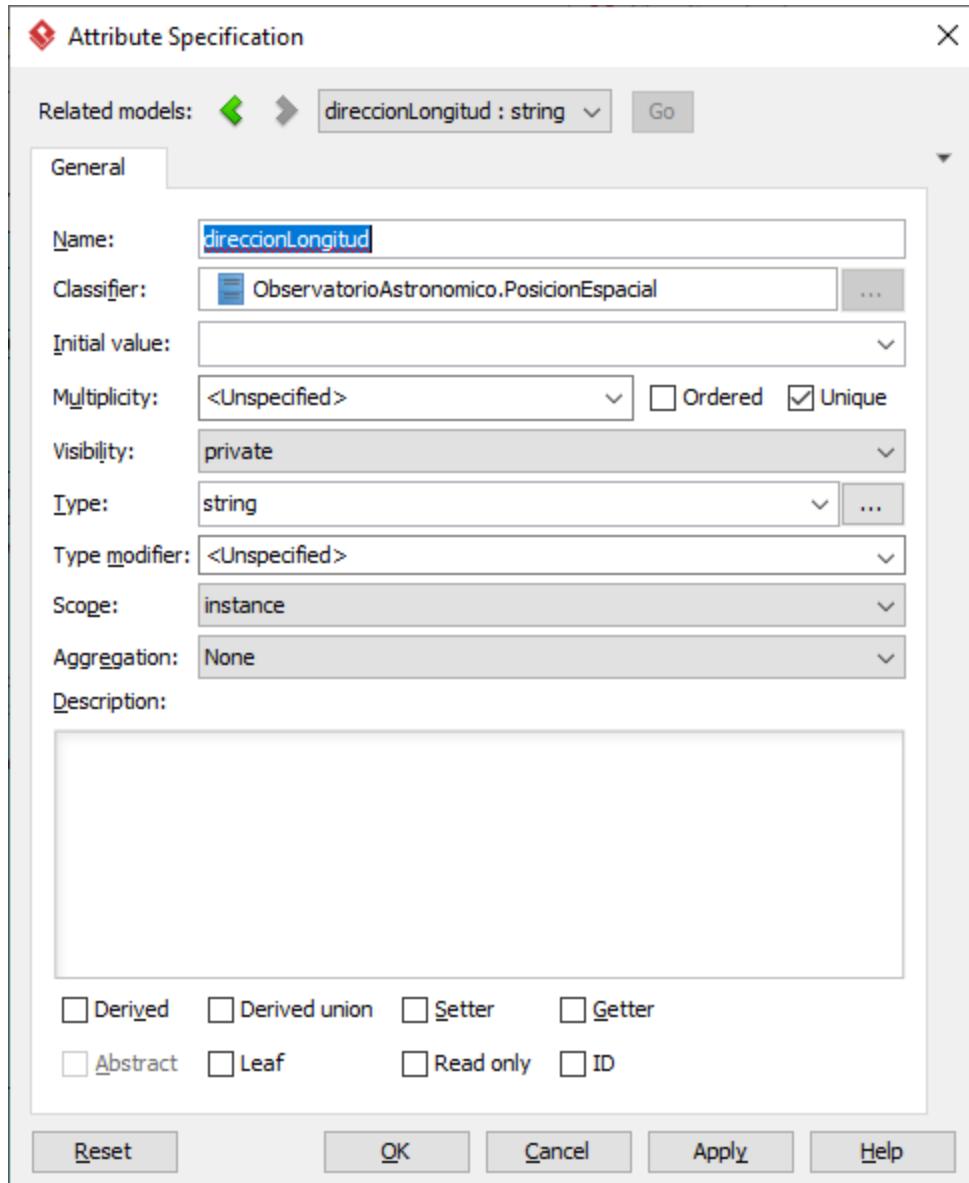
- Related models: fechaObservacion : Date
- General tab settings:
 - Name: fechaObservacion
 - Classifier: ObservatorioAstronomico.Observacion
 - Initial value:
 - Multiplicity: <Unspecified>
 - Visibility: public
 - Type: Date
 - Type modifier: <Unspecified>
 - Scope: instance
 - Aggregation: None
 - Description:
- Checkboxes at the bottom:
 - Derived
 - Derived union
 - Setter
 - Getter
 - Abstract
 - Leaf
 - Read only
 - ID
- Buttons: Reset, OK, Cancel, Apply, Help

Attribute Specification Dialog (Bottom Right):

- Related models: periodoMeses : string
- General tab settings:
 - Name: periodoMeses
 - Classifier: ObservatorioAstronomico.Observacion
 - Initial value:
 - Multiplicity: <Unspecified>
 - Visibility: private
 - Type: string
 - Type modifier: <Unspecified>
 - Scope: instance
 - Aggregation: None
 - Description:
- Checkboxes at the bottom:
 - Derived
 - Derived union
 - Setter
 - Getter
 - Abstract
 - Leaf
 - Read only
 - ID
- Buttons: Reset, OK, Cancel, Apply, Help

Creación de la clase Posicion espacial:





OPERACIONES DE LAS CLASES :

CLASE 1:

The screenshot shows two UML tool windows side-by-side.

Class Specification Window (Left):

- General tab selected.
- Operations tab selected.
- Shows a list of operations for the class "CuerpoCeleste".

Name	Classifier	Visibility	Return type
Constructor	CuerpoCeleste	public	
calcularDesplaz...	CuerpoCeleste	public	double
agregarObservac...	CuerpoCeleste	public	void
obtenerNombre	CuerpoCeleste	public	string
obtenerComposic...	CuerpoCeleste	public	string
obtenerDistancia...	CuerpoCeleste	public	double
obtenerUnidadDi...	CuerpoCeleste	public	string
establecerNombre	CuerpoCeleste	public	void
establecerCompo...	CuerpoCeleste	public	void
establecerDistanc...	CuerpoCeleste	public	void
establecerUnidad...	CuerpoCeleste	public	void

Operation Specification Window (Right):

- Related models: Constructor(nombre : string, composicion : string, distanciaATierra : double, unid...
- General tab selected.
- Name: Constructor
- Classifier: ObservatorioAstronomico.CuerpoCeleste
- Return type: <Unspecified>
- Type modifier: <Unspecified>
- Visibility: public
- Scope: instance
- Lower:
- Upper:
- Body condition:
- Description:
- Checkboxes at the bottom: Abstract, Leaf, Query, Ordered, Unique (Unique is checked).

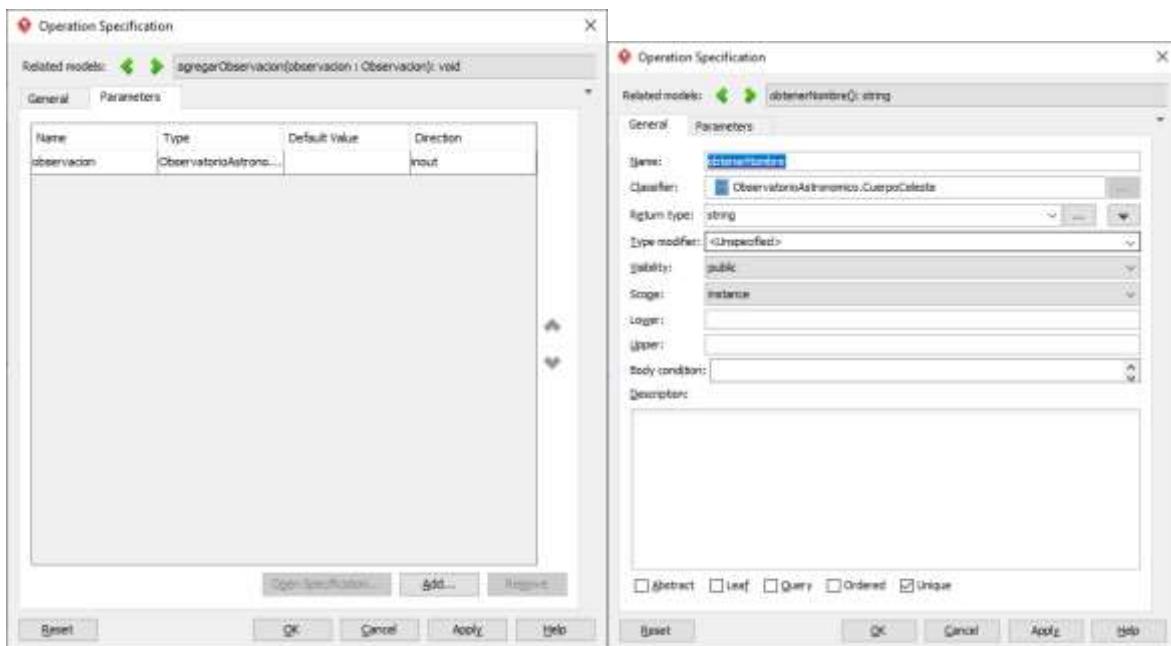
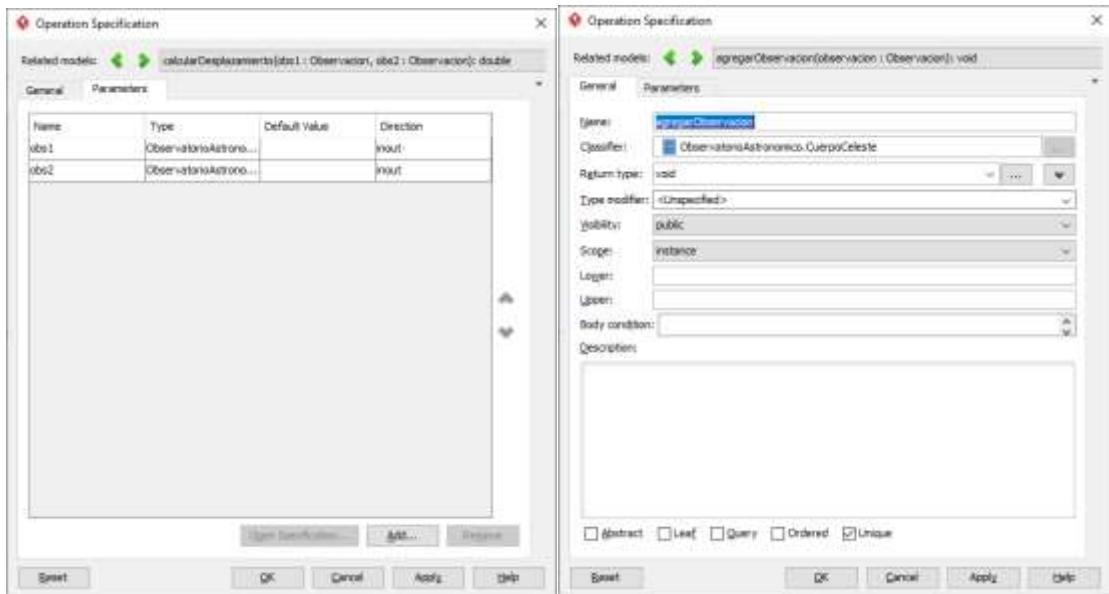
Operation Specification Window (Bottom Left):

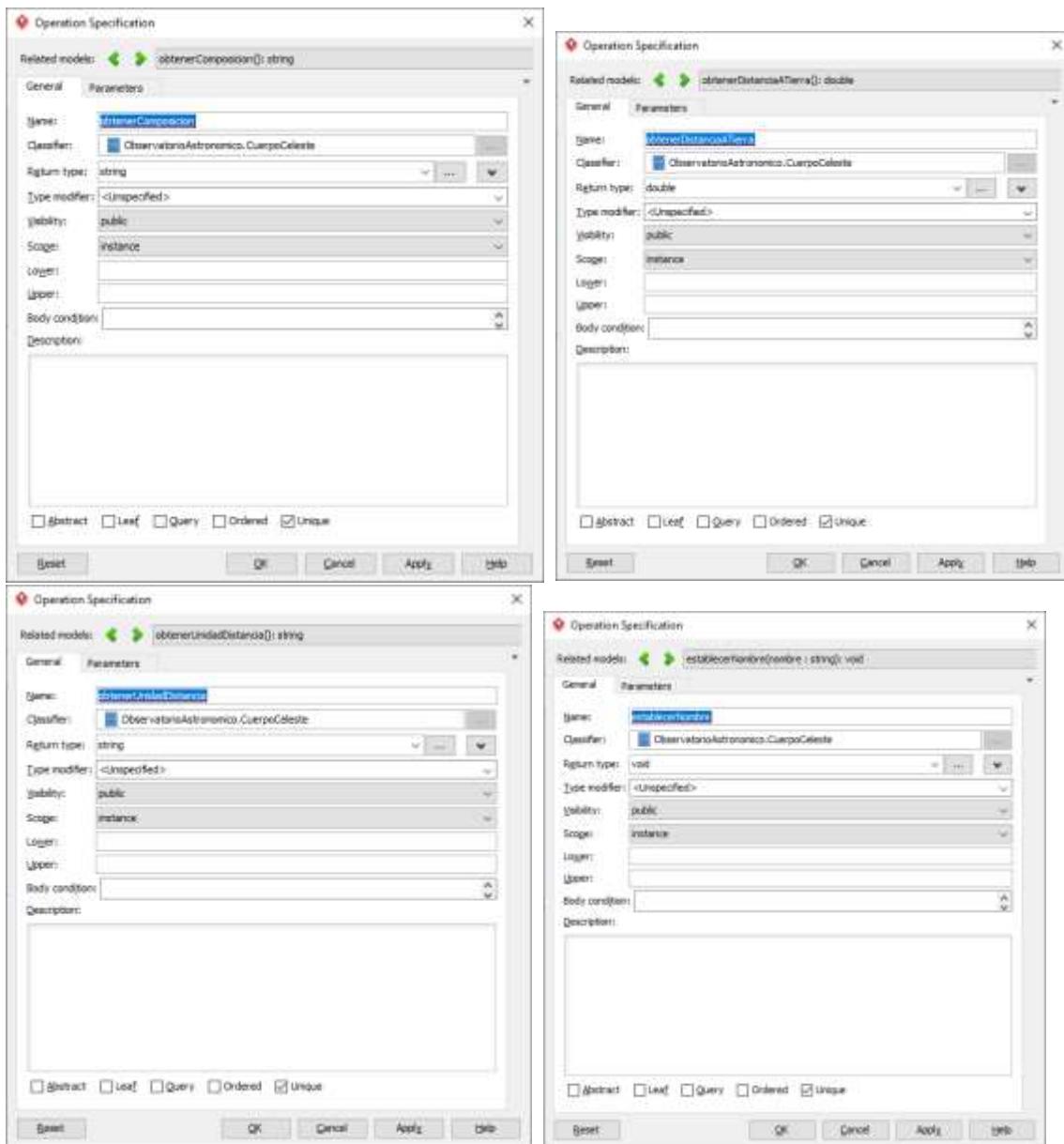
- Related models: Constructor(nombre : string, composicion : string, distanciaATierra : double, unid...
- General tab selected.
- Parameters tab selected.
- Shows parameters for the operation "calcularDesplazamiento".

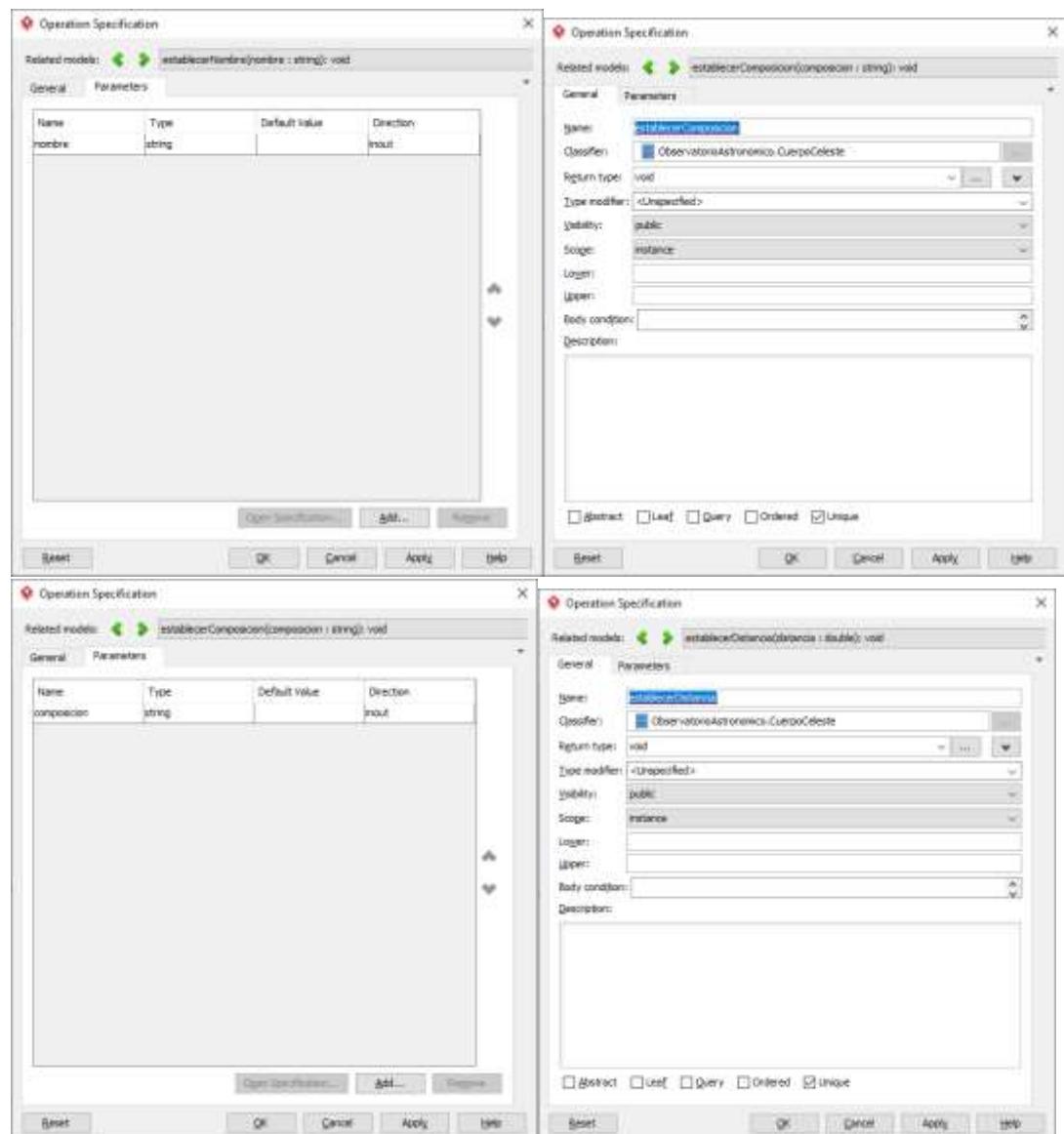
Name	Type	Default value	Direction
nombre	string		inout
composicion	string		inout
distanciaATierra	double		inout
unidaDistancia	string		inout

Operation Specification Window (Bottom Right):

- Related models: calcularDesplazamiento(obs1 : Observacion, obs2 : Observacion) : double
- General tab selected.
- Name: calcularDesplazamiento
- Classifier: ObservatorioAstronomico.CuerpoCeleste
- Return type: double
- Type modifier: <Unspecified>
- Visibility: public
- Scope: instance
- Lower:
- Upper:
- Body condition:
- Description:
- Checkboxes at the bottom: Abstract, Leaf, Query, Ordered, Unique (Unique is checked).







The image displays three overlapping UML Operation Specification dialog boxes, likely from a modeling tool like Rational Rose or similar.

Top Dialog (Operation Specification):

- Related models: establecerDistancia(distance : double); void
- General tab selected.
- Parameters table:

Name	Type	Default Value	Direction
distance	double		inout
- Buttons: Open Specification..., Add..., Remove, Reset, OK, Cancel, Apply, Help.

Middle Dialog (Operation Specification):

- Related models: establecerUnidadDistancia(unidad : string); void
- General tab selected.
- Parameters table:

Name	Type	Default Value	Direction
unidad	string		inout
- Buttons: Open Specification..., Add..., Remove, Reset, OK, Cancel, Apply, Help.

Bottom Dialog (Operation Specification):

- Related models: establecerUnidadDistancia(unidad : string); void
- Parameters tab selected.
- Fields:
 - Name: establecerUnidadDistancia
 - Classifier: ObservatorioAstronomico.CuerpoCeleste
 - Return type: void
 - Type modifier: <Unspecified>
 - Visibility: public
 - Scope: instance
 - Lower:
 - Upper:
 - Body condition:
- Checkboxes: Abstract, Leaf, Query, Ordered, Unique (Unique is checked).
- Buttons: Reset, OK, Cancel, Apply, Help.

OPERACIONES CLASE OBSERVACION:

The image displays two windows of the 'Operation Specification' dialog box from a UML modeling tool.

Top Window (General Tab):

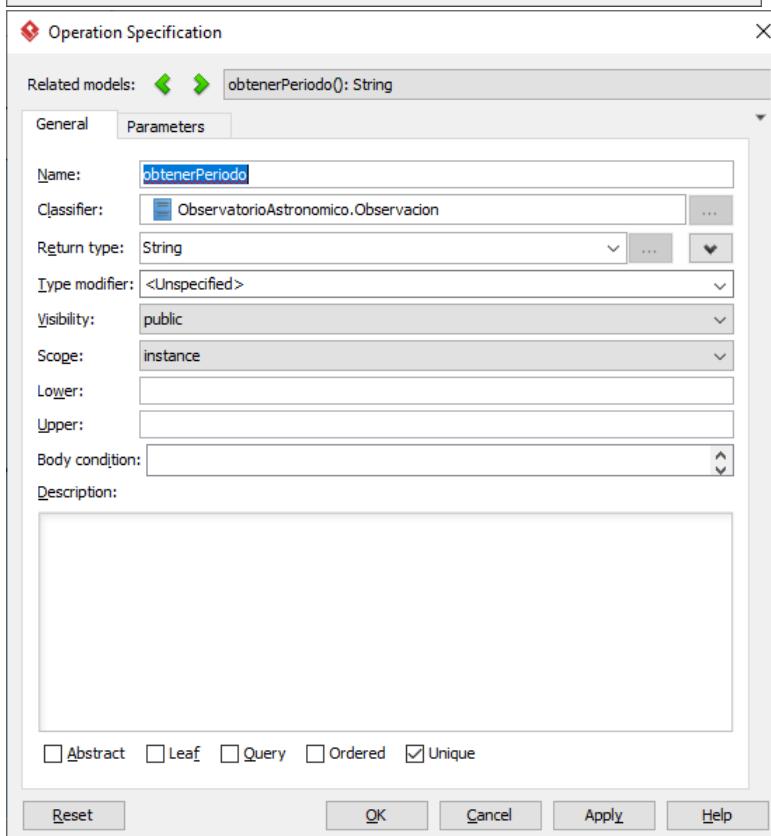
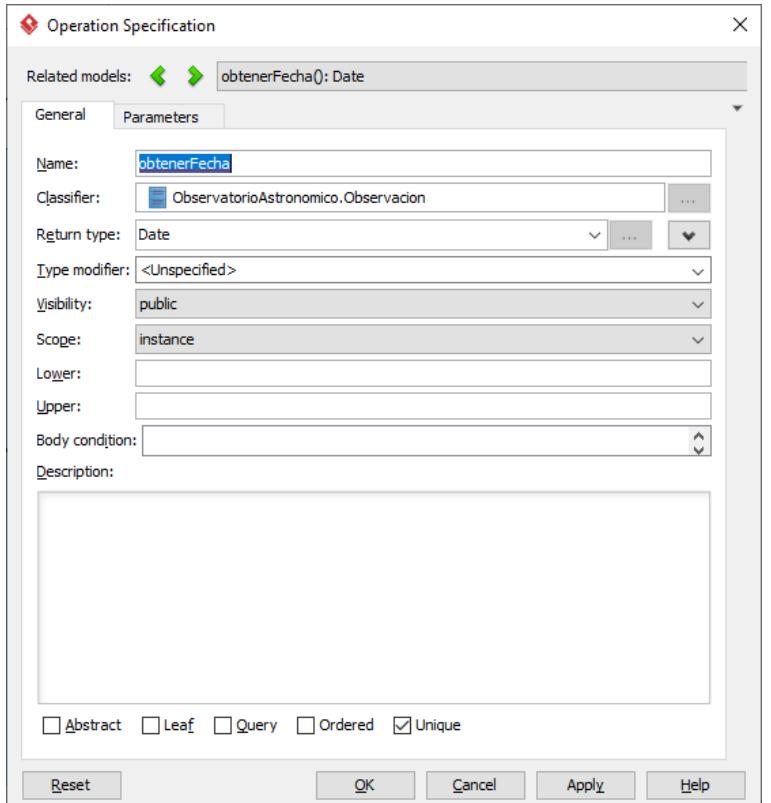
- Related models: Observacion (fechaObservacion : Date, periodoMeses : string, posicion : Posicion)
- General tab selected.
- Name: Observacion
- Classifier: ObservatorioAstronomico.Observacion
- Return type: <Unspecified>
- Type modifier: <Unspecified>
- Visibility: public
- Scope: instance
- Layer:
- Upper:
- Body condition:
- Description:
- Checkboxes at the bottom: Abstract (unchecked), Leaf (unchecked), Query (unchecked), Ordered (unchecked), Unique (checked).

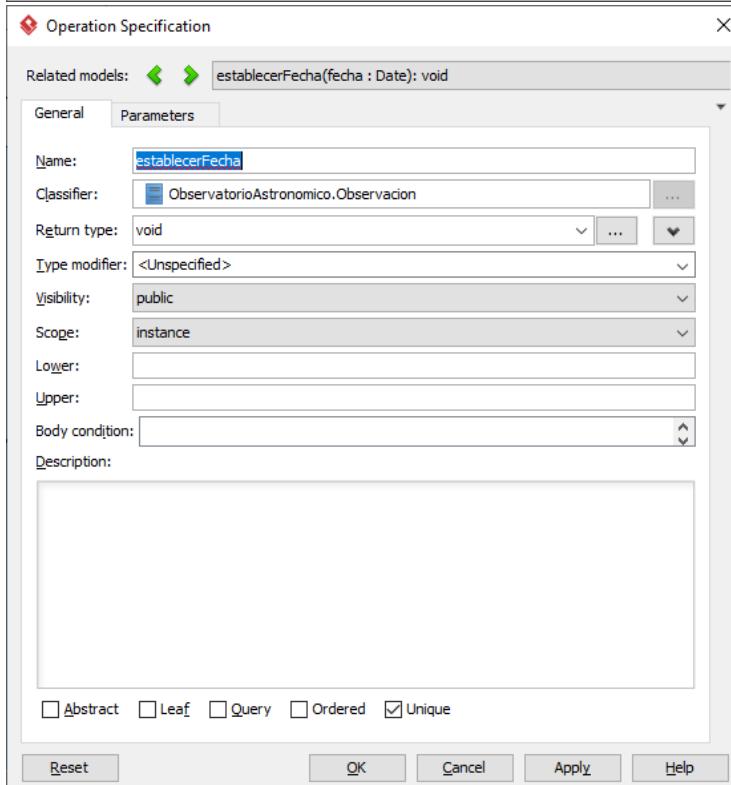
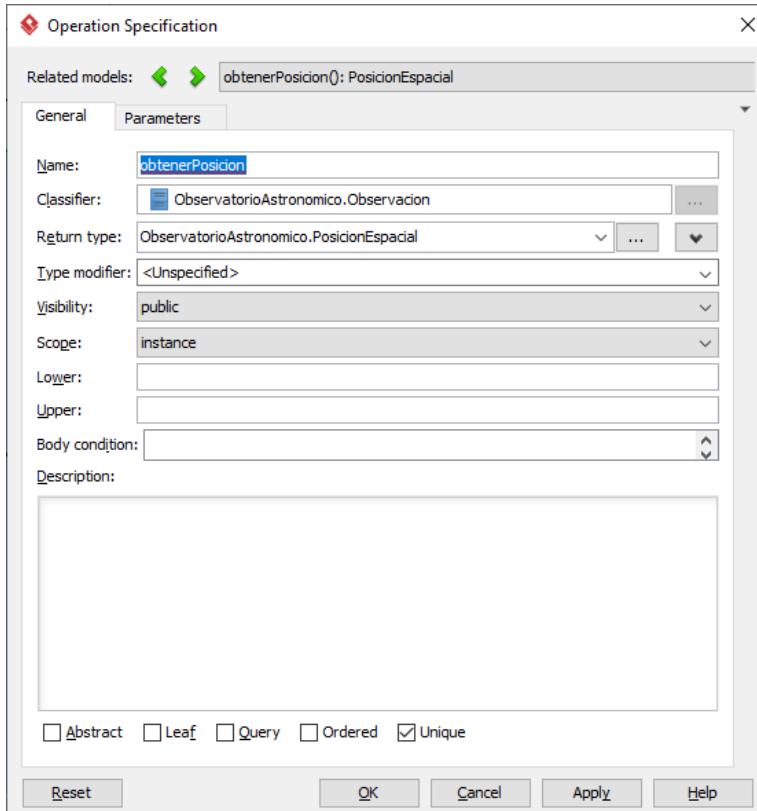
Bottom Window (Parameters Tab):

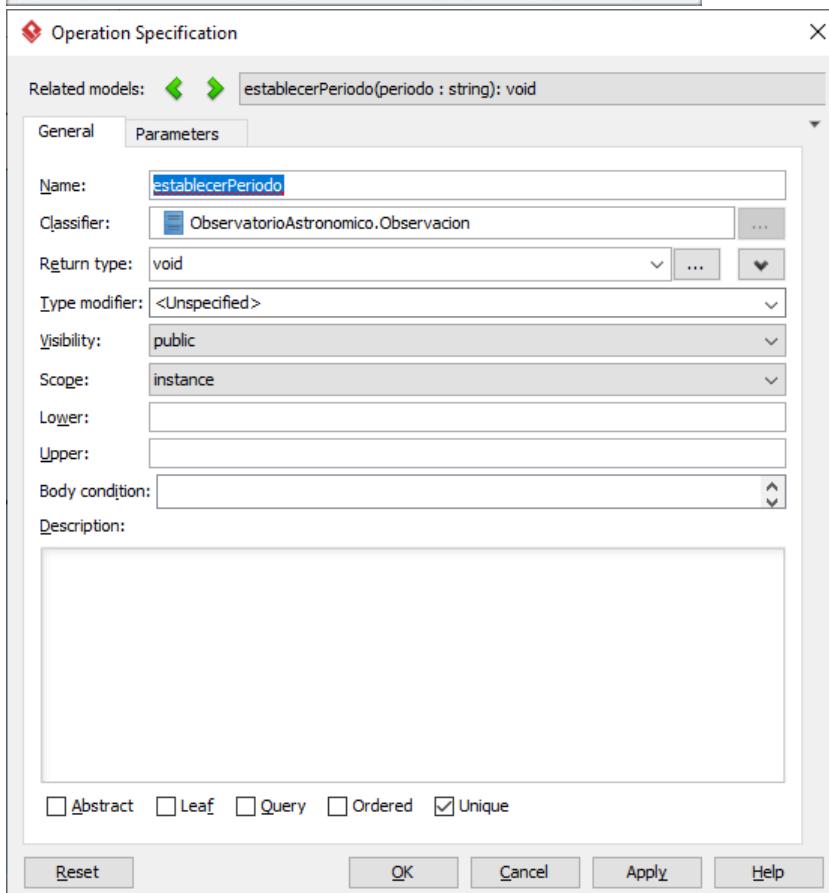
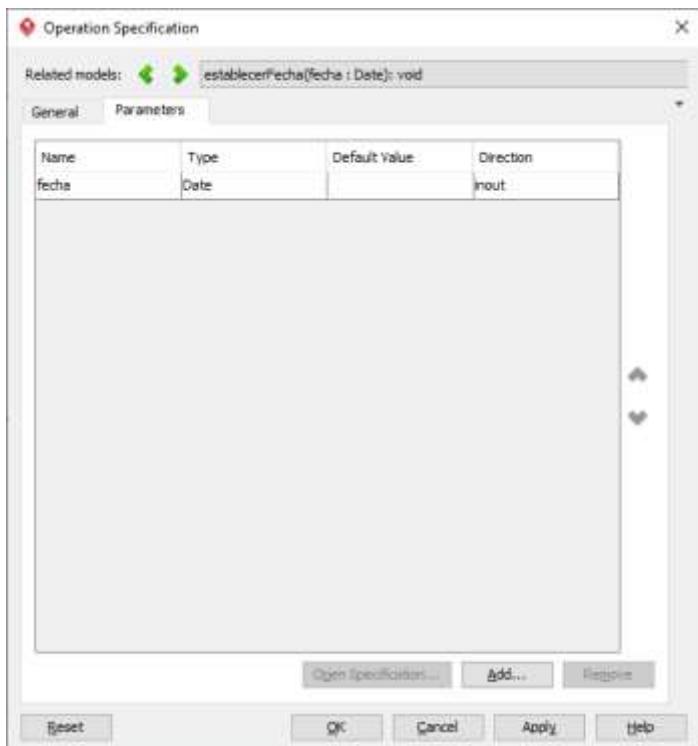
- Related models: Observacion (fechaObservacion : Date, periodoMeses : string, posicion : Posicion)
- Parameters tab selected.
- Table of parameters:

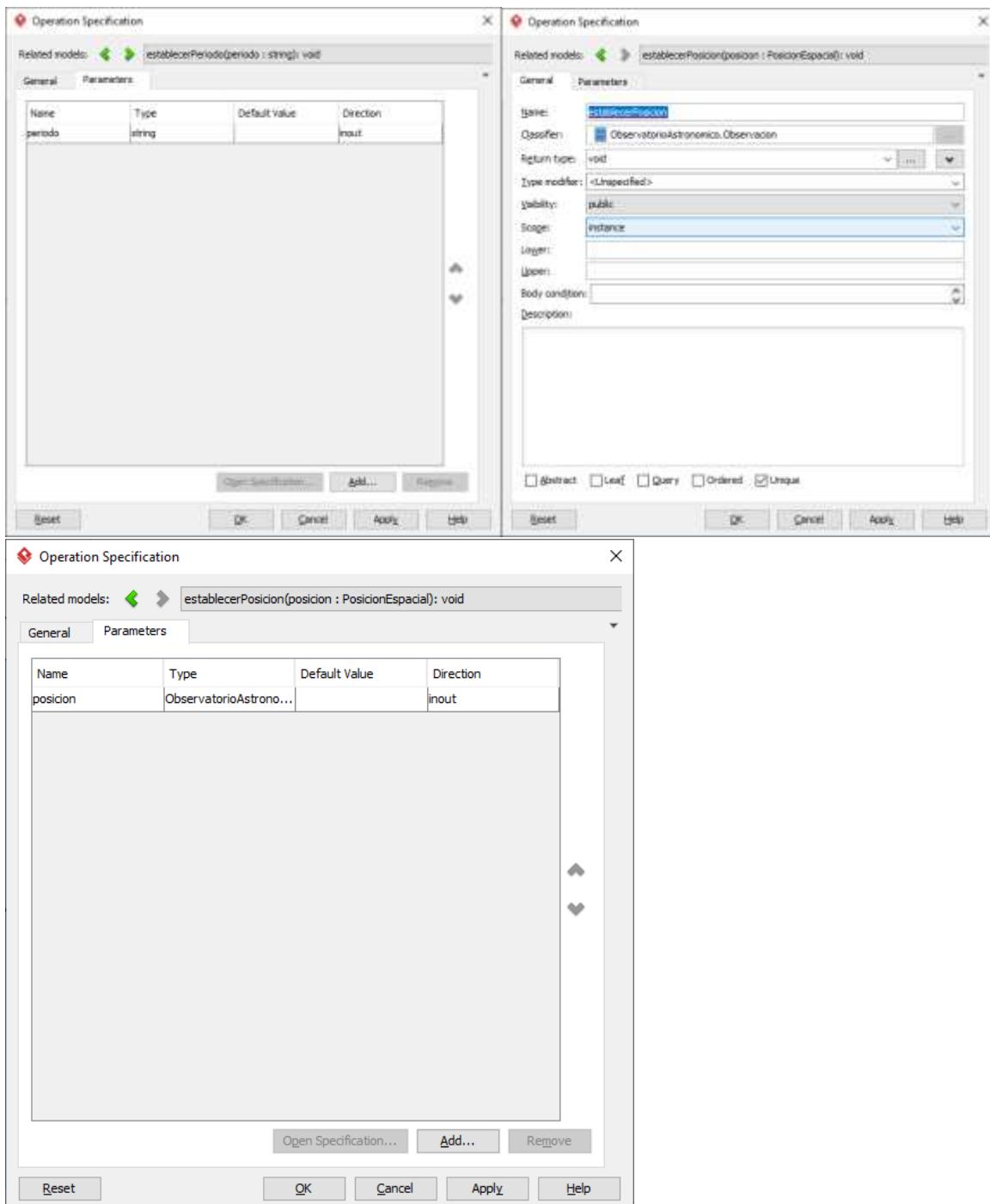
Name	Type	Default Value	Direction
fechaObservacion	Date		inout
periodoMeses	string		inout
posicion	ObservatorioAstronomico.Posicion		inout

- Buttons at the bottom: Open Specification..., Add..., Remove, Reset, OK, Cancel, Apply, Help.









OPERACIONES CLASE POSICION ESPACIAL:

The image displays two overlapping windows of the 'Operation Specification' dialog from a UML modeling tool.

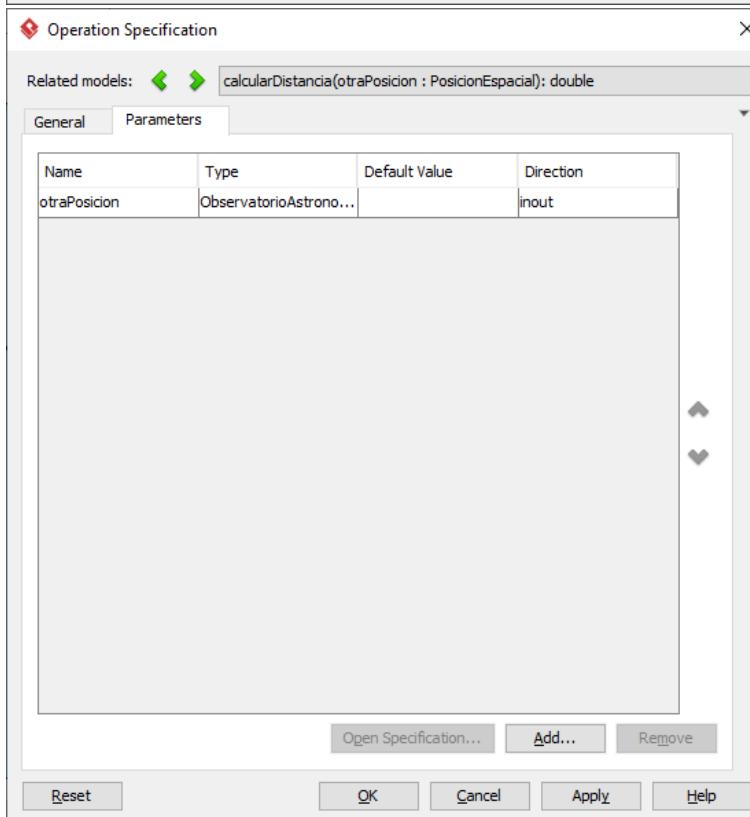
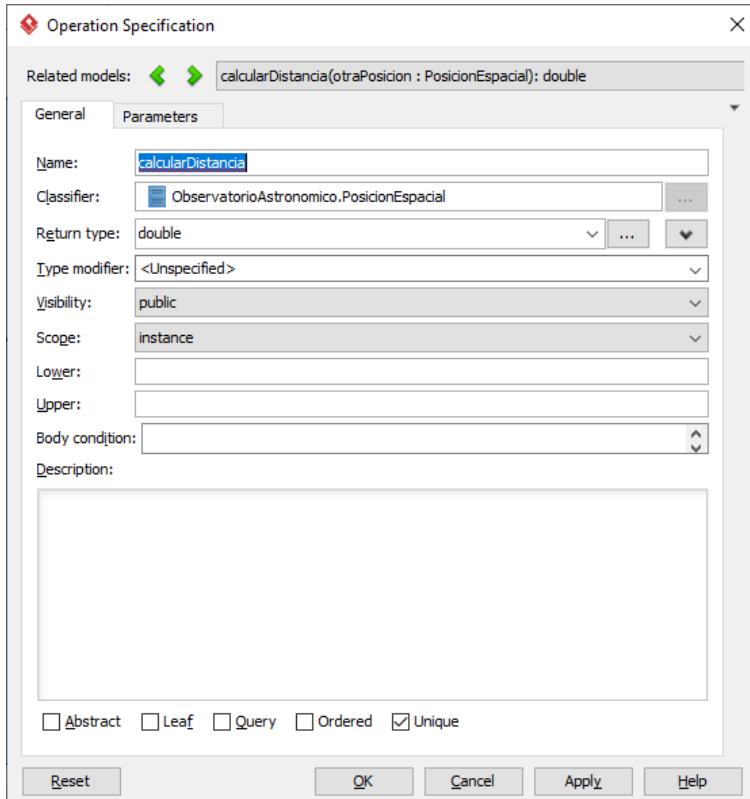
Top Window (General Tab):

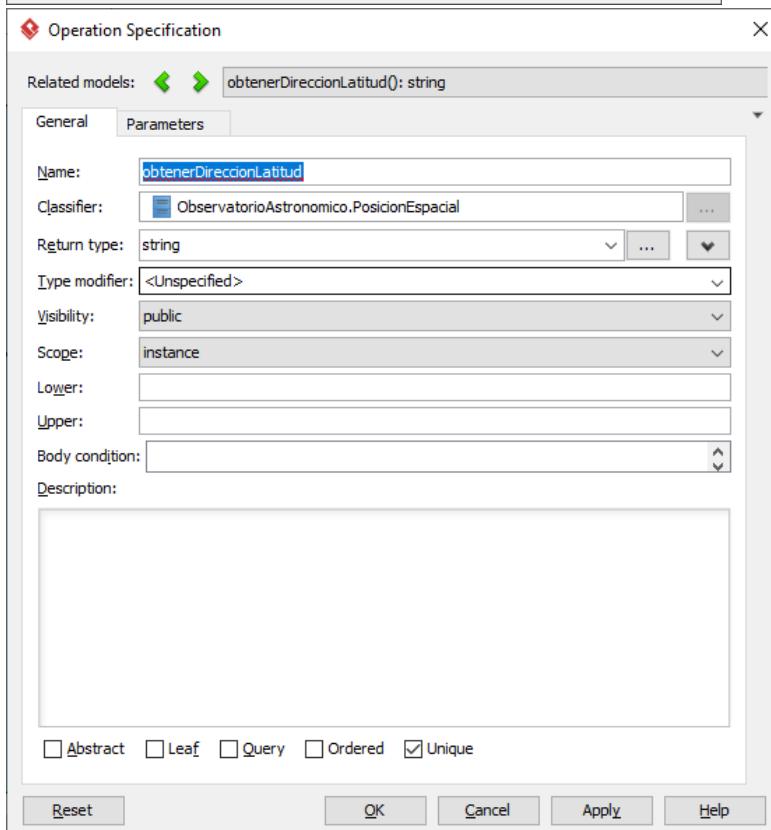
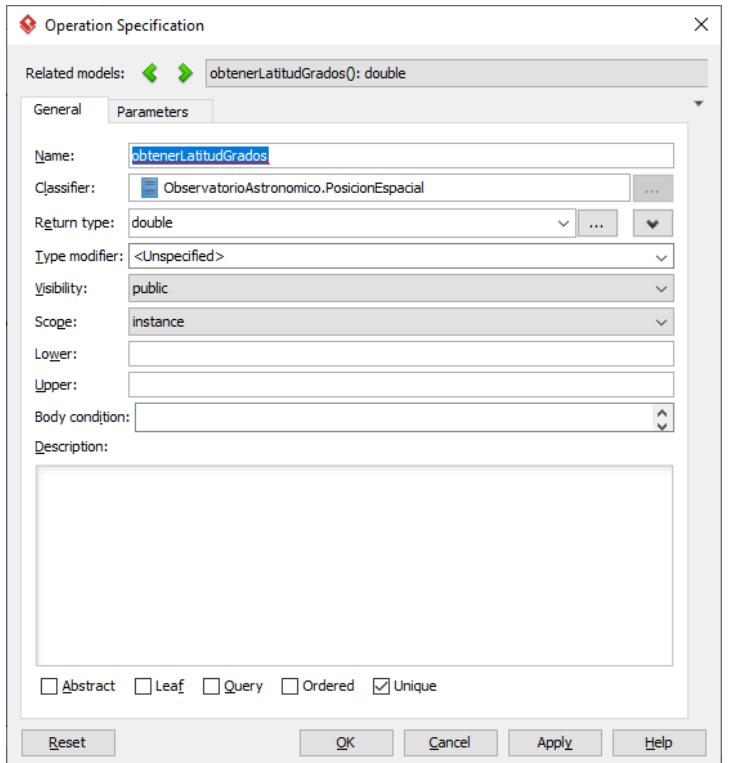
- Related models:** PosicionEspecial (latitudGrados : double, direccionLatitud : string, longitudGrados : double)
- General:**
 - Name:** PosicionEspecial
 - Classifier:** ObservatorioAstronomico.PosicionEspecial
 - Return type:** (empty)
 - Type modifier:** <Unspecified>
 - Visibility:** public
 - Scope:** instance
 - Layer:** (empty)
 - Upper:** (empty)
 - Body condition:** (empty)
- Description:** (empty text area)
- Checkboxes:** Abstract, Leaf, Query, Ordered, Unique (Unique is checked)
- Buttons:** Reset, OK, Cancel, Apply, Help

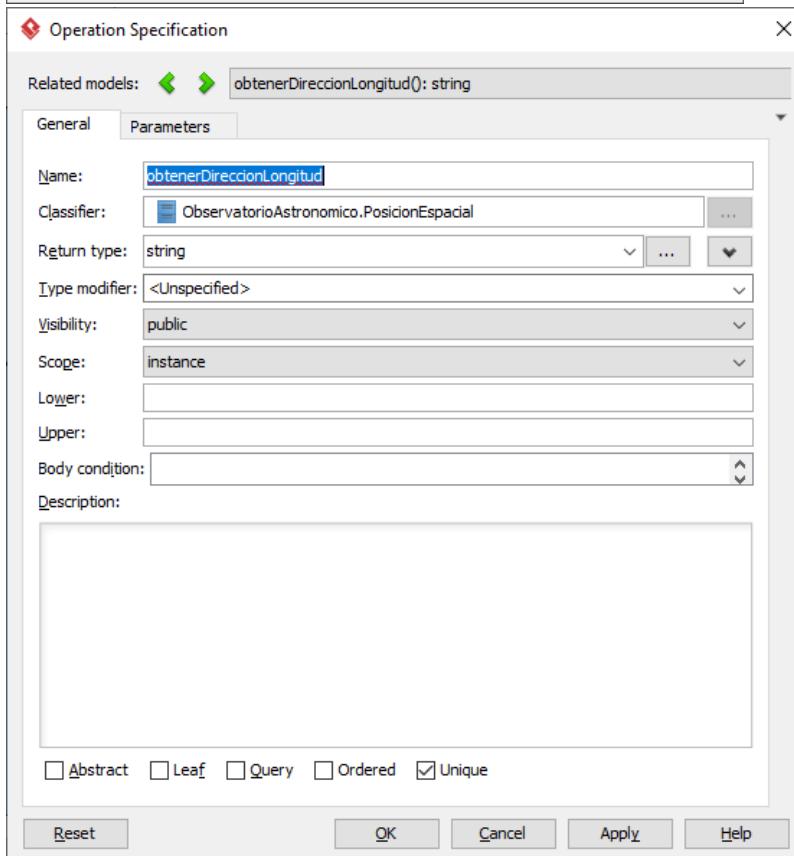
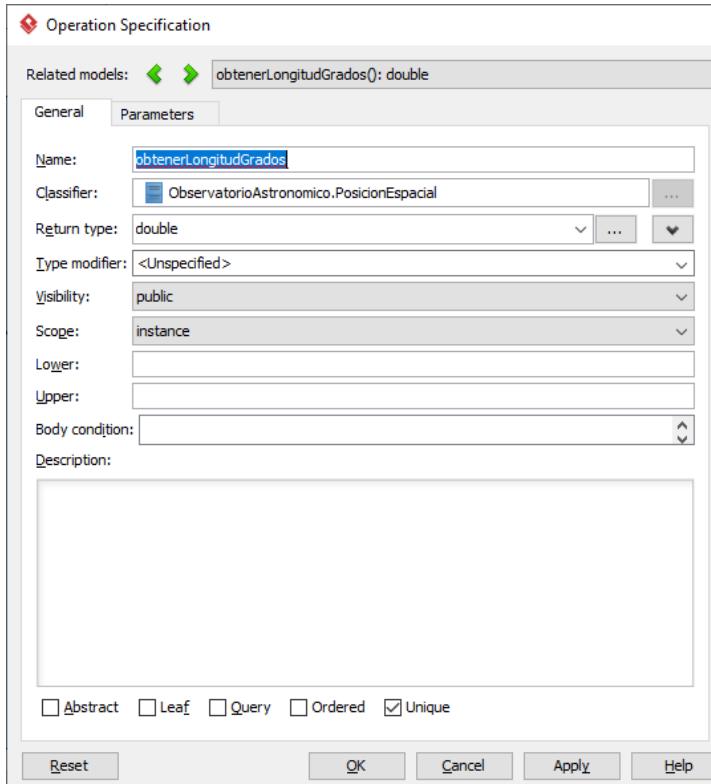
Bottom Window (Parameters Tab):

- Related models:** PosicionEspecial (latitudGrados : double, direccionLatitud : string, longitudGrados : double)
- Parameters:**

Name	Type	Default Value	Direction
latitudGrados	double		inout
direccionLatitud	string		inout
longitudGrados	double		inout
direccionLongitud	string		inout
- Buttons:** Open Specification..., Add..., Remove, Reset, OK, Cancel, Apply, Help







❖ Operation Specification

Related models: obtenerLatitudCompleta(): string

General Parameters

Name: **obtenerLatitudCompleta**

Classifier: ObservatorioAstronomico.PosicionEspacial

Return type: string

Type modifier: <Unspecified>

Visibility: public

Scope: instance

Lower:

Upper:

Body condition:

Description:

Abstract Leaf Query Ordered Unique

Reset OK Cancel Apply Help

❖ Operation Specification

Related models: obtenerLongitudCompleta(): string

General Parameters

Name: **obtenerLongitudCompleta**

Classifier: ObservatorioAstronomico.PosicionEspacial

Return type: string

Type modifier: <Unspecified>

Visibility: public

Scope: **instance**

Lower:

Upper:

Body condition:

Description:

Abstract Leaf Query Ordered Unique

Reset OK Cancel Apply Help

❖ Operation Specification

Related models: convertirACadena(): string

General Parameters

Name: convertirACadena
Classifier: ObservatorioAstronomico.PosicionEspacial
Return type: string
Type modifier: <Unspecified>
Visibility: public
Scope: instance
Lower:
Upper:
Body condition:
Description:

Abstract Leaf Query Ordered Unique

Reset OK Cancel Apply Help

❖ Operation Specification

Related models: establecerLatitud(latitud : double, direccion : string): void

General Parameters

Name: establecerLatitud
Classifier: ObservatorioAstronomico.PosicionEspacial
Return type: void
Type modifier: <Unspecified>
Visibility: public
Scope: instance
Lower:
Upper:
Body condition:
Description:

Abstract Leaf Query Ordered Unique

Reset OK Cancel Apply Help

The image displays three overlapping UML Operation Specification dialog boxes, likely from a modeling tool like Rational Rose or similar.

Top Left Dialog: Shows the operation `establecerLatitud(latitud : double, direccion : string): void`. The General tab is selected, showing parameters:

Name	Type	Default Value	Direction
latitud	double		inout
direccion	string		inout

Top Right Dialog: Shows the operation `establecerLongitud(longitud : double, direccion : string): void`. The General tab is selected, showing parameters:

Name	Type	Default Value	Direction
longitud	double		inout
direccion	string		inout

Bottom Dialog: Shows the operation `establecerLongitud(longitud : double, direccion : string): void`. The Parameters tab is selected, showing parameters:

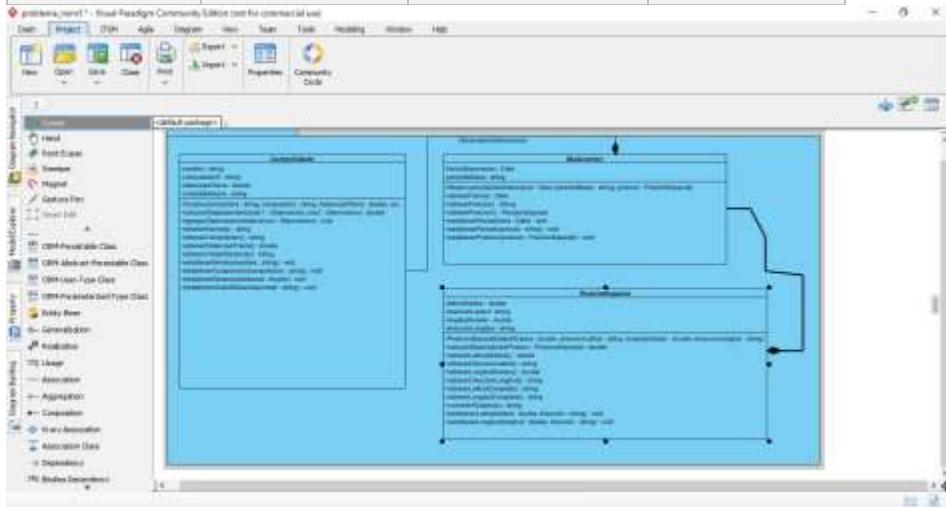
Name	Type	Default Value	Direction
longitud	double		inout
direccion	string		inout

RELACIÓN 1: CuerpoCeleste → Observacion

Clase Origen	Tipo Relación	Clase Destino	Cardinalidad
CuerpoCeleste	Composición	Observacion	1..*

RELACIÓN 2: Observacion → PosicionEspacial

Clase Origen	Tipo Relación	Clase Destino	Cardinalidad
Observacion	Composición	PosicionEspacial	1



URL repositorio:

<https://github.com/Jynsst/practica0>