

ED05

Tarea

ED05 – Tarea

Índice:

1	Selección de sustantivos.	3
1.1	Relación Clases con sus Atributos y Métodos.	3
1.2	Obtención de relaciones:	7
1.3	Primer Refinamiento	17
1.4	Documentación.....	17
1.5	Incluir el Proyecto VP-UML en un proyecto de NetBeans	18

1 Selección de sustantivos.

1.1 Relación Clases con sus Atributos y Métodos.

Clase	Atributos		Métodos	Comentarios
EmpresaAdmin	listadoEmpleados	ArrayList	Constructor	Nadie tiene acceso a los atributos salvo la propia Clase.
	listadoComunidades	ArrayList		
	nombre	String		
	CIF	String		
	direccion	String		
Comunidad	codigoComunidad	String	Constructor Getters&Setters	
	direccion	String		
	listadoVecinos	Array		Cada inmueble pertenece a un vecino, y cada vecino pertenece a una Comunidad, por lo tanto, el listado de vecinos tiene un tamaño fijo, no es necesaria una estructura dinámica.
	saldo	float		
	codigoAdmin	String		Se identifica al administrador de la Comunidad mediante el código de empleado, que Administrador hereda de Empleado. Por el codigoAdmin podemos saber qué empresa es la que gestiona la Comunidad.
	listadoFacturas	LinkedHashSet		Estructura dinámica que no admita repetidos y se ordene por entrada con mucha eficiencia al acceso.
	listadoActas	LinkedHashSet		

Usuario (Abstracta)	nombre	String	Constructor Getters&Setters*	*Los vecinos pueden modificar sus datos personales
	DNI	String		
	correo	String	verFacturas	Los usuarios navegan por la web para ver las Facturas de la Comunidad
	direccion	String	verActas	Los usuarios deben poder visualizar las Actas de la Comunidad
Vecino	listadoInmueble	ArrayList	Constructor	De los vecinos nos interesa su código de inmueble.Un mismo vecino puede ser propietario de varios inmuebles, por lo tanto se debe poder asignar varios inmuebles a un mismo vecino.
	numCuentaBancaria	String	Getters&Setters	
			visualizarInmuebles	Los vecinos pueden visualizar sus pisos. Se recorre el ArrayList listadoInmueble, llamando al método toString de Inmueble para obtener la información de cada Inmueble del Vecino.
			pagarDeuda	Los vecinos pueden pagar la deuda de cada Inmueble con el numCuentaBancaria
Presidente			convocaJunta	Las reuniones de vecinos las convoca un Vecino que es Presidente de la Comunidad, que avisa al Administrador.

Trabajador	cargo	String	Constructor	
	codigoEmpleado	String	Getters&Setters	
			pagarFactura	Podrán pagar las facturas de las empresas contratadas para hacer servicios en la Comunidad con el saldo de la Comunidad.
			contrataServicio	Los trabajadores pueden solicitar servicios a empresas para que realicen un servicio en la Comunidad.
Administrador			Constructor	
			mandaCartaJunta	El Administrador de Fincas es quien convoca oficialmente la reunión mandando una carta a cada vecino.
			conteoVecinos	Es el encargado de realizar el conteo de vecinos que asisten a la Junta.
			ConteoVotosItem	Es el encargado de contar el número de votos a favor y en contra de cada punto del orden del día (Item).
Inmueble	saldoInmueble	float	Constructor Getters&Setters	Si el saldo es negativo, es decir, el inmueble es moroso, el vecino asociado no tiene derecho a voto en la reunión
	codInmueble	String	toString	
	tipo	Tipoinmueble		El tipo de Inmueble puede ser piso, garaje o trastero.

Tipoinmueble <<Enumerado>>				El tipo de Inmueble puede ser piso, garaje o trastero.
Proveedor	nombreProveedor	String	generarFactura	De las facturas nos interesa el nombre de empresa y CIF, que son atributos del Proveedor.
	CIF	String	Constructor Getters&Setters	
Factura	proveedor	Proveedor	Constructor Getters&Setters	De las facturas nos interesa el nombre de empresa y CIF, que son atributos del Proveedor.
	pagada	boolean		
	descripcion	String		
	fecha	Date		
	importe	float		
	numFactura	int		
	codigoComunidad	String		
Acta	codigoActa	String	Constructor Getters&Setters	
	fecha	Date	infoActa	Los usuarios deben poder visualizar las Actas de la Comunidad
	listadoVecinosAsisten	ArrayList		
	listadoItems	ArrayList		
Item	codigoItem	String	Constructor Getters&Setters	
	aprobado	boolean		

Incluye:

Extracción de los sustantivos en la descripción del problema.

Selección de sustantivos como objetos/clases del sistema.

Obtención de los atributos de los objetos.

Obtención de los métodos.

Añadir getters, setters y constructores.

1.2 Obtención de relaciones:

Item – Acta: Asociación

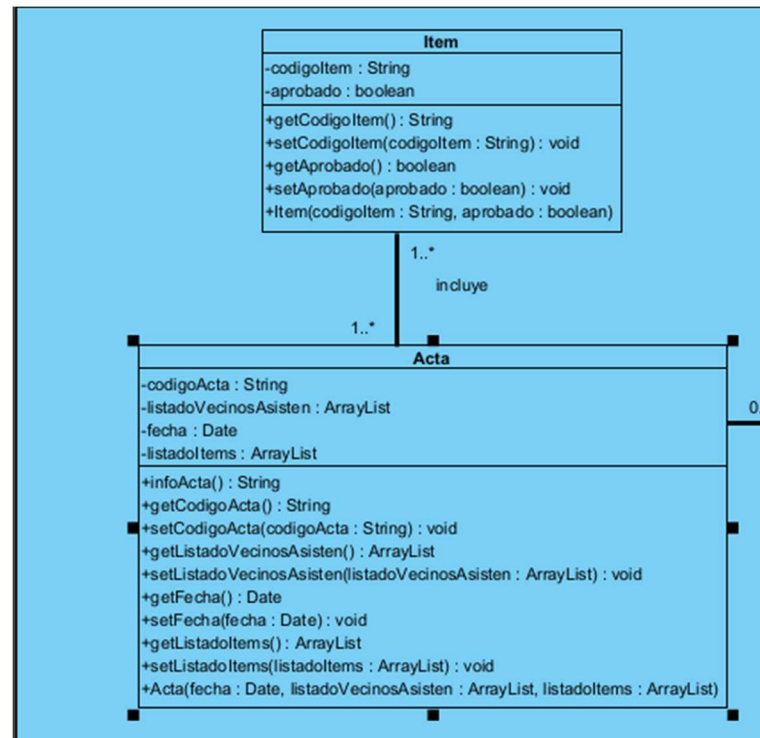
En cada junta de vecinos se crea un Acta en la que se tratan unos Items. Los ítems siguen existiendo aunque desaparezca el Acta y tienen entidad propia, no pertenecen al Acta, si no que el tema existe por si mismo.

En una Acta se han de incluir al menos un Item, ya que si no, no tendría sentido hacer una Junta, si no hay nada que tratar.

En una Acta se pueden incluir muchos Items, cuando haya muchos puntos a tratar.

Un Item al menos ha de incluirse en un Acta, si no no tiene sentido crearlo, ya que son puntos a tratar en Actas.

Un Item se puede incluir en varias Actas, cuando no se pueda resolver en una directamente, se tendrá que tratar en varias hasta que quede resuelto.

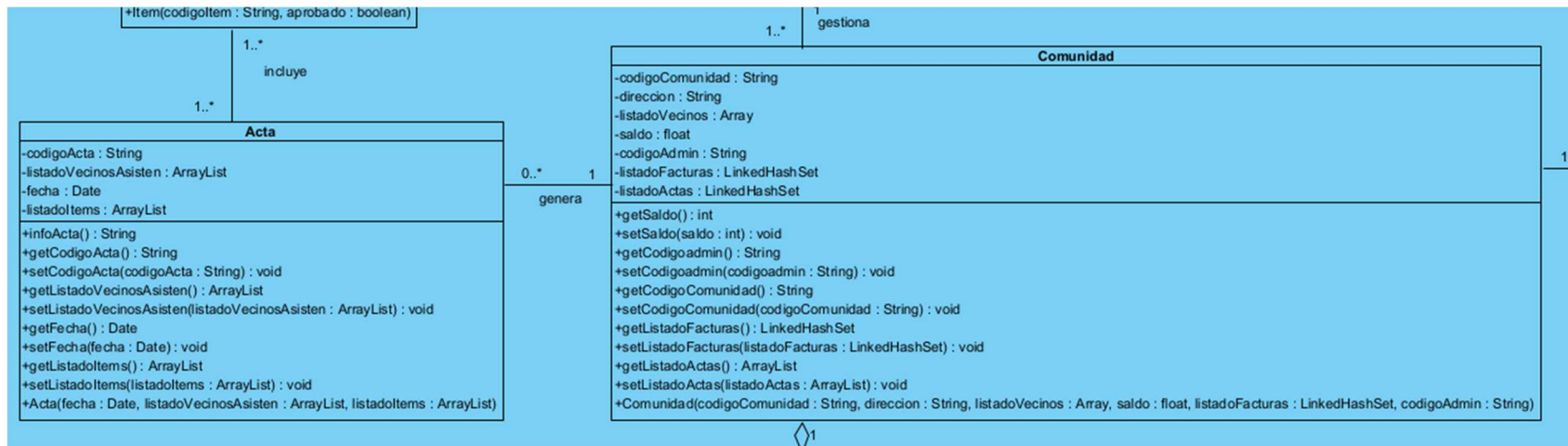


Comunidad – Acta: Asociación

Una Comunidad genera Actas que no dejan de existir si desaparece la Comunidad y tampoco pertenecen a la misma.

Una Comunidad puede no generar ningún Acta, si se acaba de establecer por ejemplo, o puede generar muchas Actas, si ya se han hecho muchas Juntas.

Un Acta sólo puede pertenecer a una Comunidad.



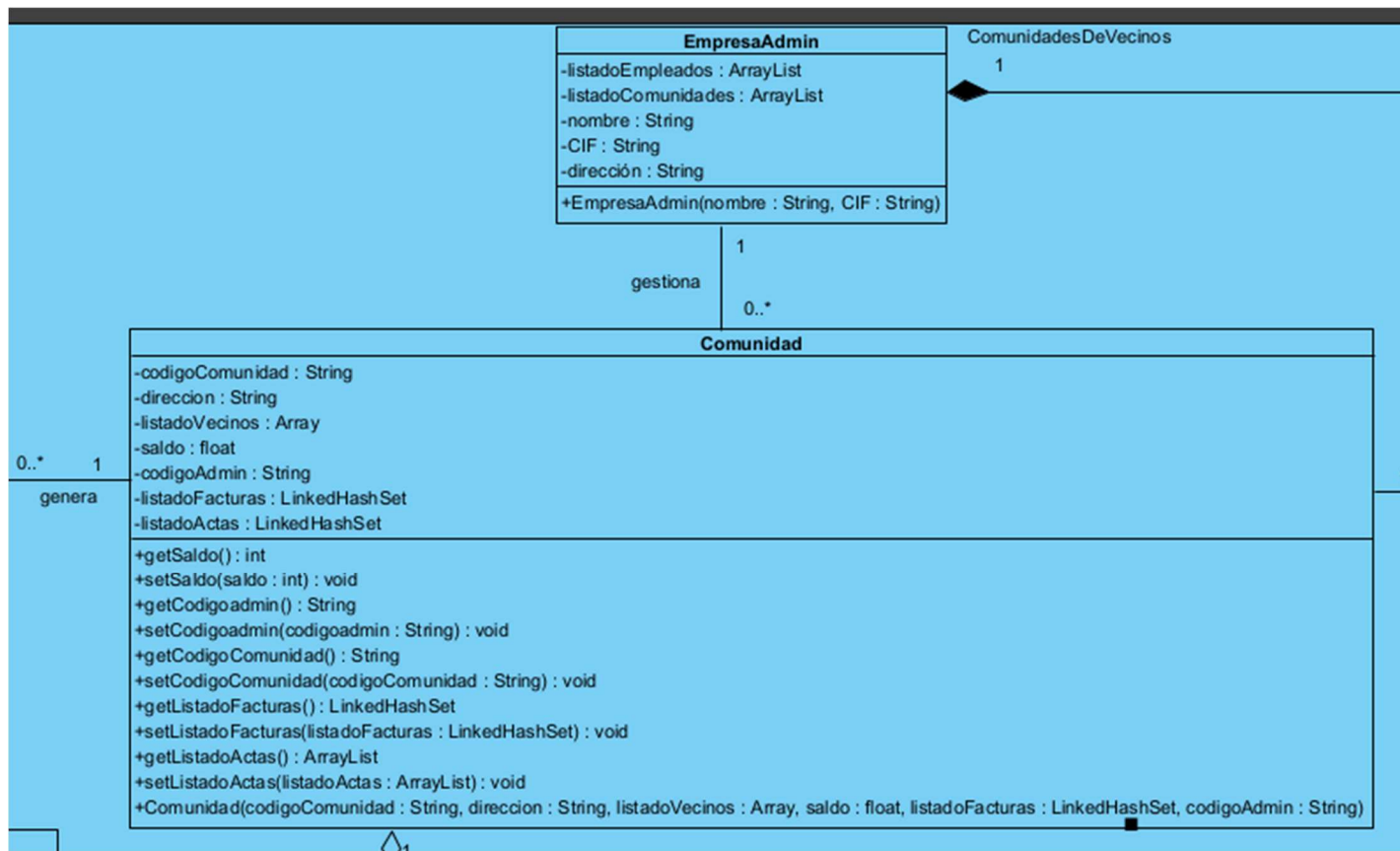
Comunidad- EmpresaAdmin: Asociación

La Empresa Administradora gestiona Comunidades, ambas Clases tienen entidad propia y la existencia de una no depende de la otra, así mismo, una no pertenece a la otra.

Una Comunidad es gestionada por una Empresa Administradora.

Una Empresa Administradora puede gestionar 0 Comunidades cuando se instituye, pero aún no tiene clientes.

Una Empresa Administradora puede gestionar muchas Comunidades, cuando ha conseguido muchos clientes.



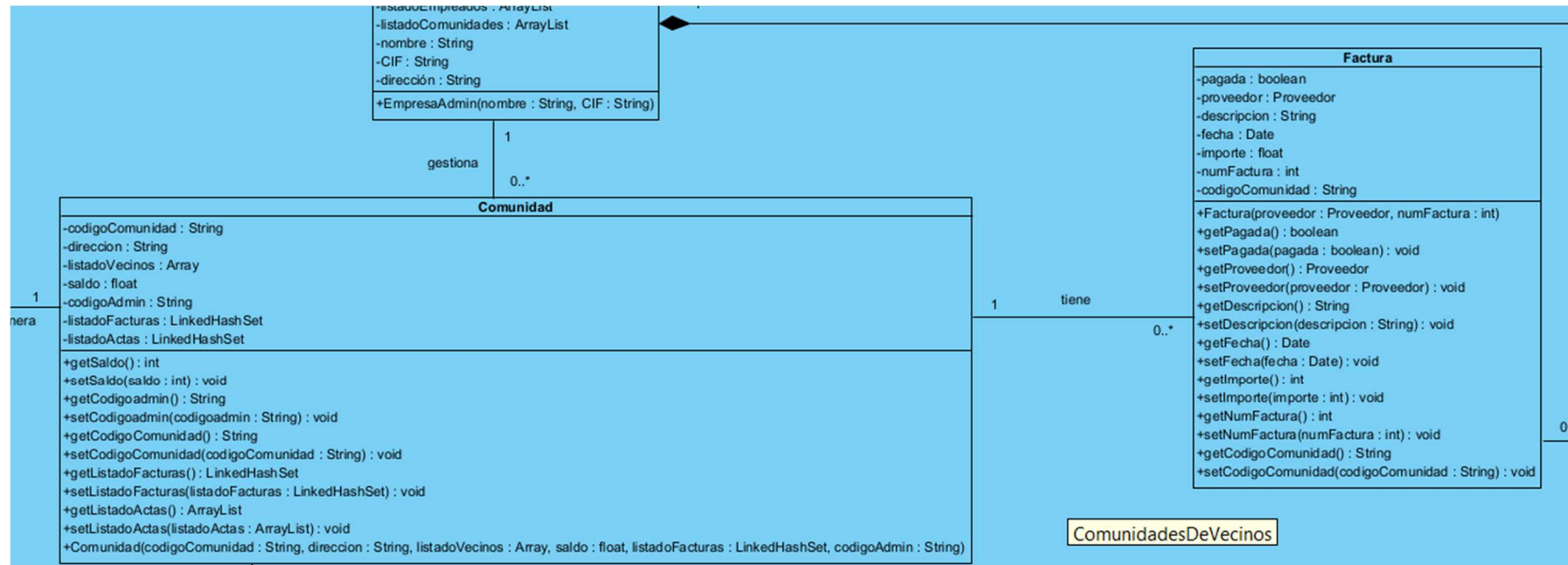
Comunidad-Factura: Asociación

Una Comunidad tiene Facturas, si la Comunidad desaparece las Facturas siguen existiendo, y ambas Clases tienen entidad propia, las Facturas no forman parte de la Comunidad.

Una Comunidad puede no tener ninguna factura en el momento de su constitución.

Una Comunidad puede tener muchas Facturas por muchos servicios distintos.

Una Factura sólo puede pertenecer a una Comunidad, ya que las Facturas se emiten a una única entidad. Si el mismo servicio implica a varias Comunidades, cada una de ellas tendrá su Factura independiente.



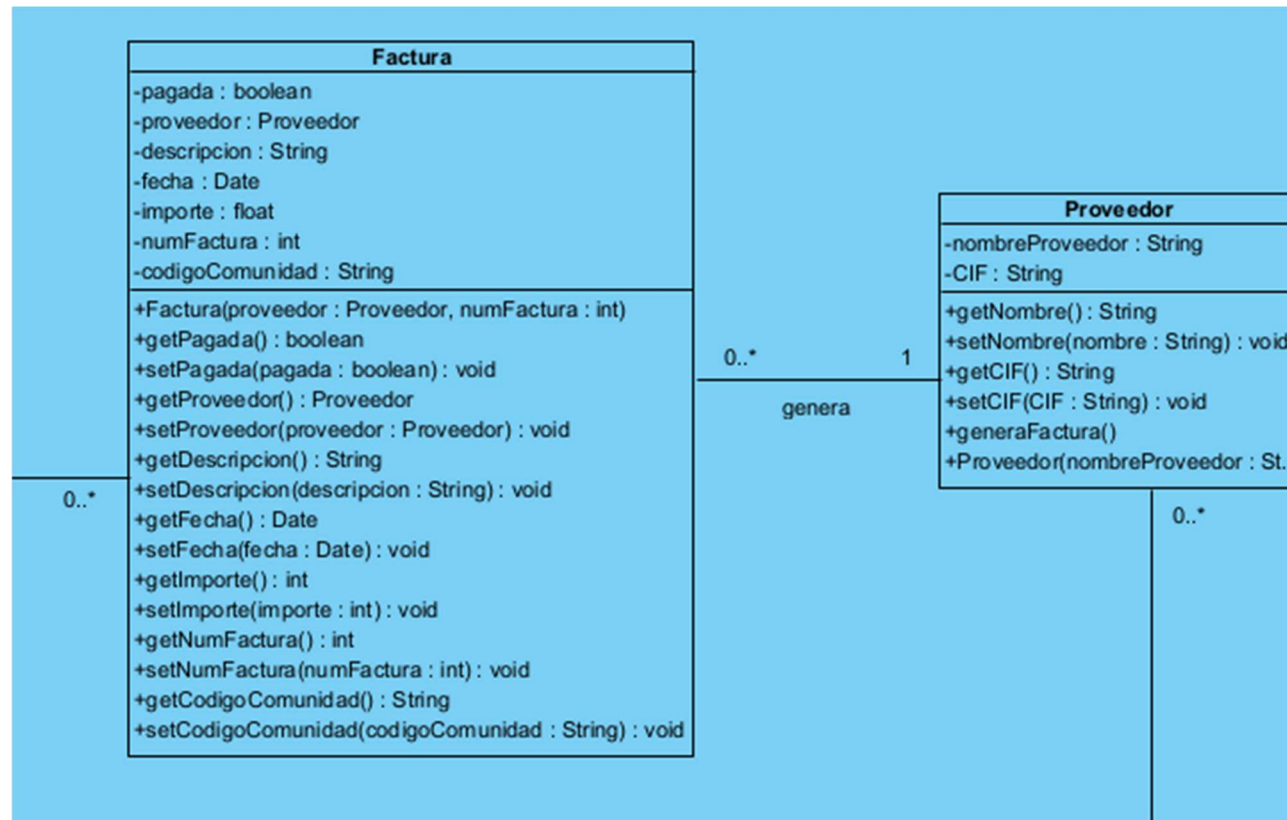
Factura-Proveedor: Asociación

Los Proveedores generan Facturas por servicios realizados, son entidades independientes, ya que la desaparición del Proveedor no implica la desaparición de la Factura, y la Factura no pertenece a la entidad Proveedor, no es una parte de este.

Un proveedor puede no haber generado ninguna Factura, por ejemplo cuando se crea.

Un Proveedor puede haber generado muchas Facturas, tras realizar muchos servicios.

Una Factura es generada por un Proveedor, si un trabajo lo realizan varios Proveedores, cada uno de ellos generará una Factura independiente con su servicio correspondiente.



Proveedor-Trabajador: Asociación

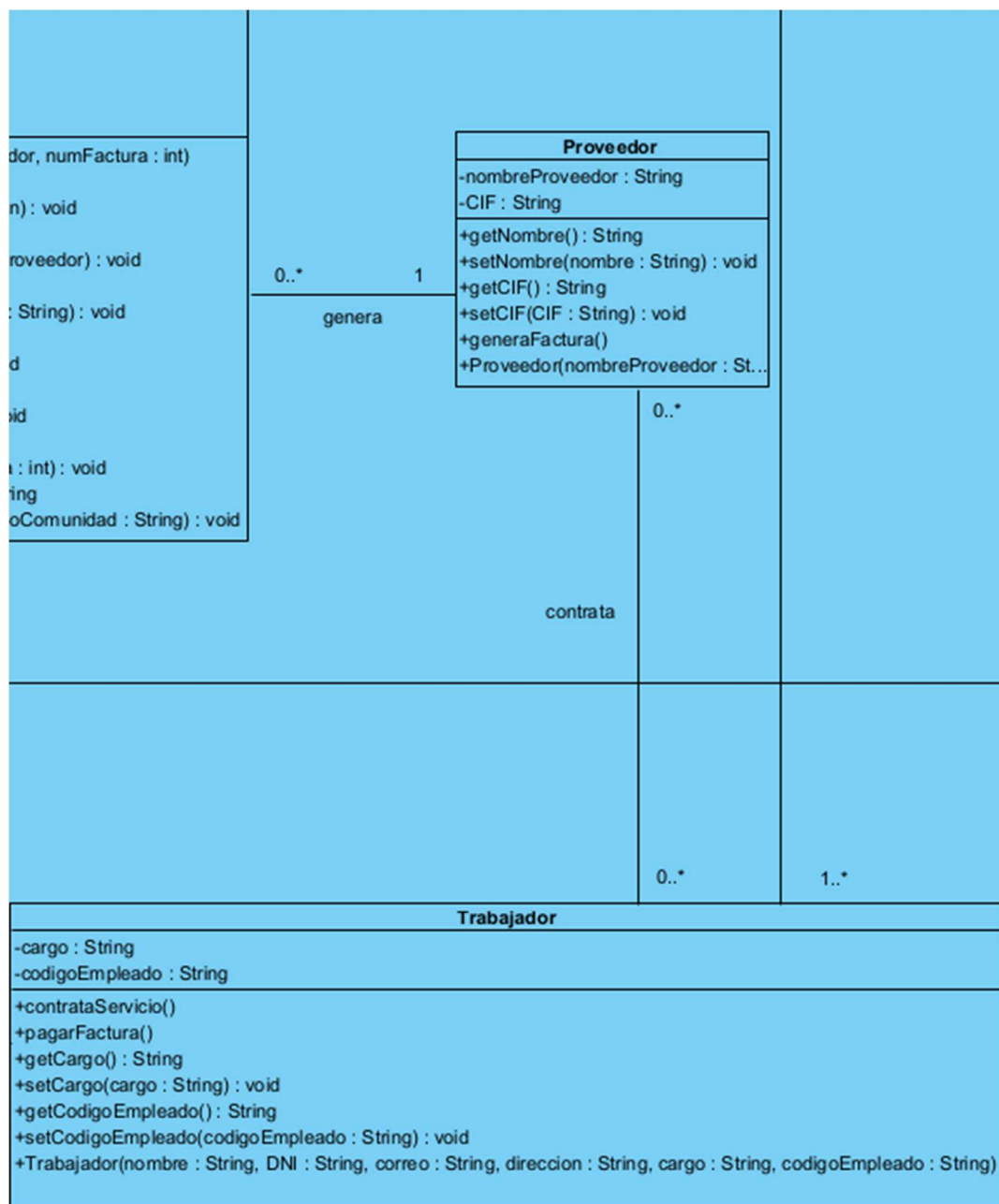
Un Trabajador de la Empresa Administradora contrata a un Proveedor de servicios, son entidades independientes, la desaparición de uno no hace que desaparezca el otro, y ninguno de ellos es parte del otro.

Un Proveedor puede no ser contratado por ningún Trabajador.

Un Proveedor puede ser contratado por muchos Trabajadores.

Un Trabajador puede no contratar a ningún Proveedor, si no necesita llevar a cabo ningún trabajo que requiera un servicio.

Un Trabajador puede contratar a muchos Proveedores, cuando necesita llevar a cabo trabajos que requieran de muchos servicios.

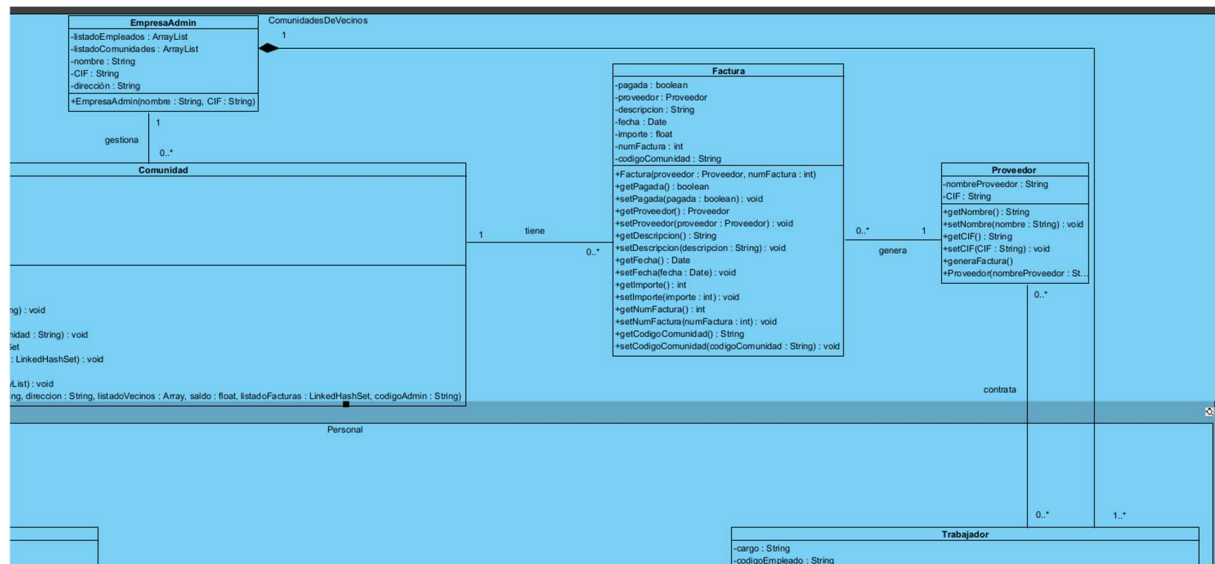


EmpresaAdmin – Trabajador: Composición

Un Trabajador es una Persona que trabaja en la Empresa Administradora, si la Empresa Administradora desaparece la Persona deja de ser un Trabajador de la misma.

Un Trabajador sólo puede trabajar en una Empresa Administradora.

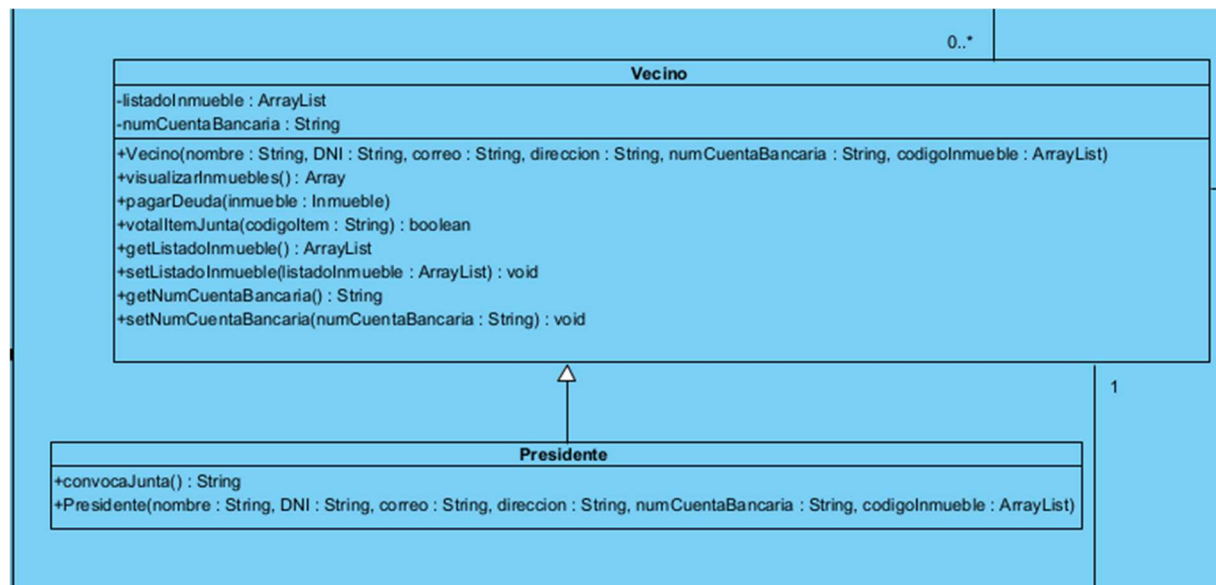
Una Empresa Administradora puede tener mínimo 1 empleado, al constituir la el que la crea por ejemplo, o muchos empleados, cuando ya tiene volumen de clientes e ingresos suficientes.



Los Usuarios del sistema pueden ser Vecinos o Trabajadores, Vecinos y Trabajadores heredan Atributos y Métodos de la Clase Usuario.

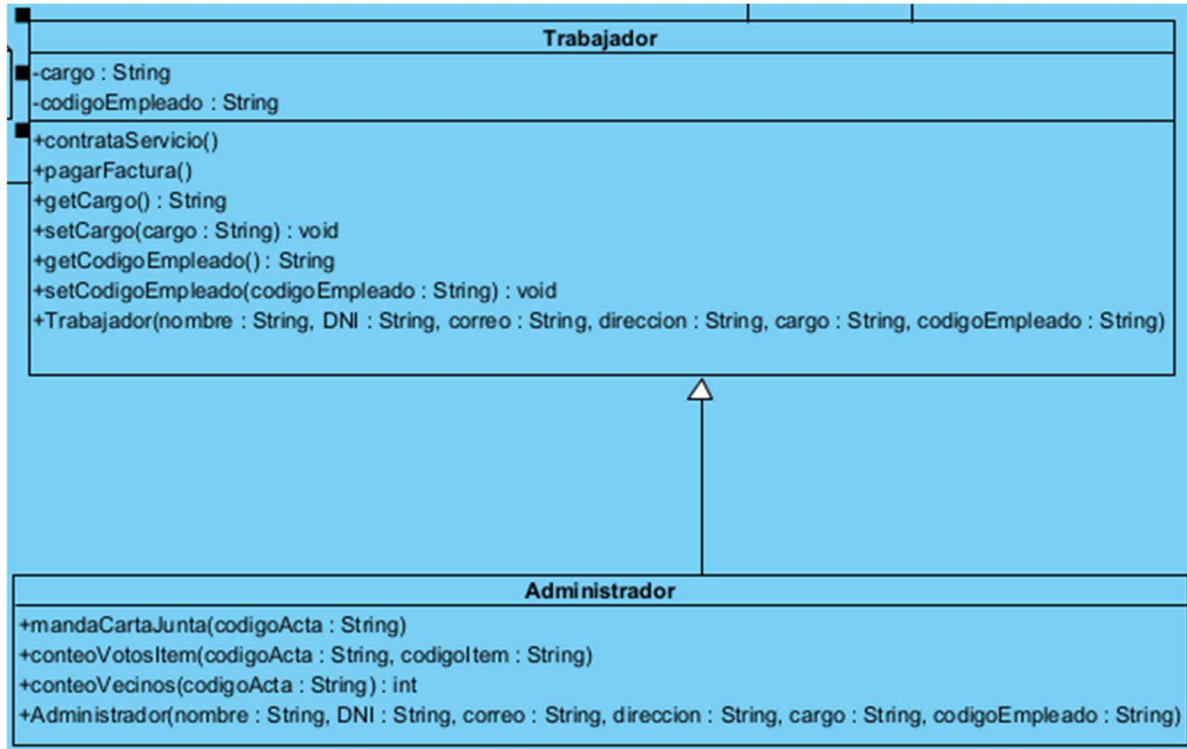
[illegible]

Hay un vecino que es el Presidente. El Presidente hereda todos los Atributos y Métodos de Vecino y a su vez tiene Métodos Propios, es decir, acciones que sólo puede llevar a cabo el Presidente.



Trabajador-Administrador:Herencia

Hay un Trabajador que es el Administrador. El Administrador hereda todos los Atributos y Métodos de Trabajador y a su vez tiene Métodos Propios, es decir, acciones que sólo puede llevar a cabo el Administrador.



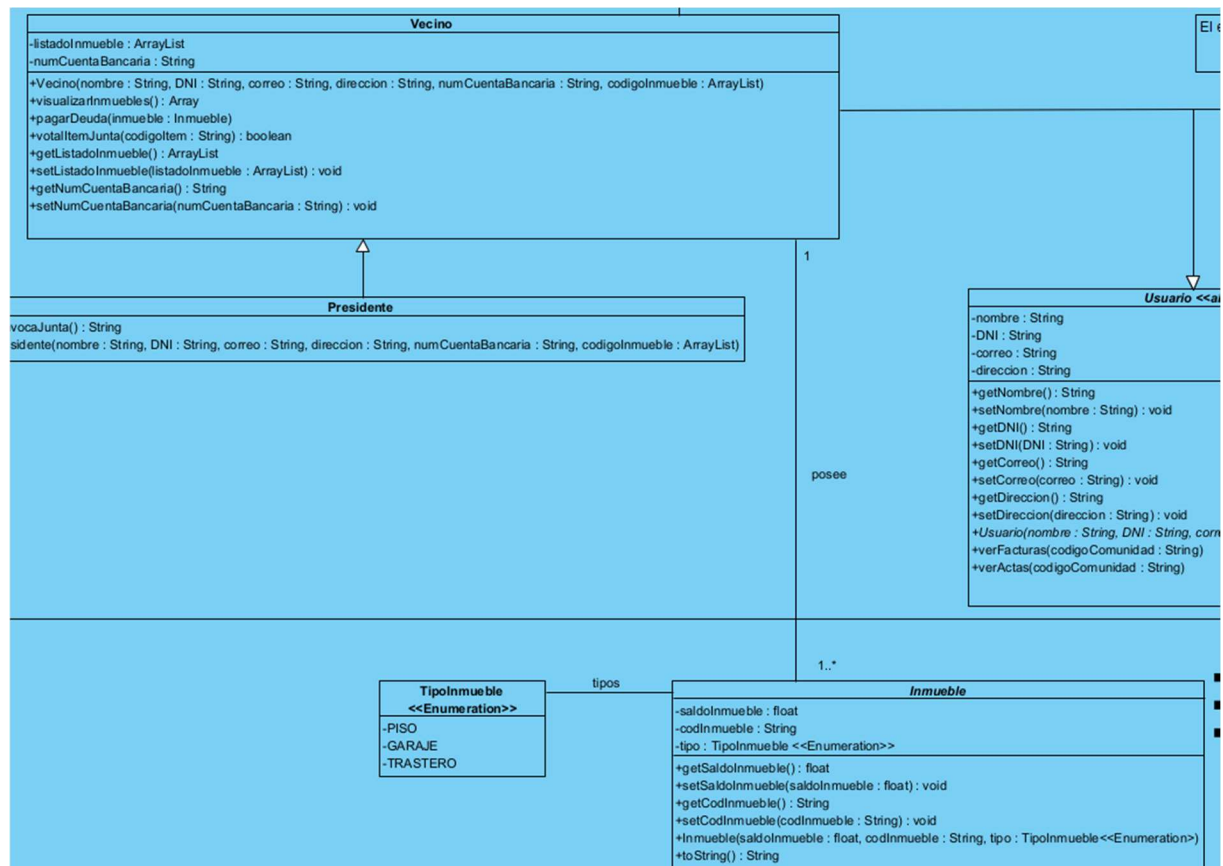
Vecino-Inmueble: Asociación

Un Vecino posee un Inmueble, son entidades independientes, ya que aunque desaparezca el Vecino el Inmueble sigue existiendo, y un Inmueble no es parte de Vecino.

Un Vecino ha de tener al menos 1 Inmueble, porque si no es propietario de ninguno no es Vecino.

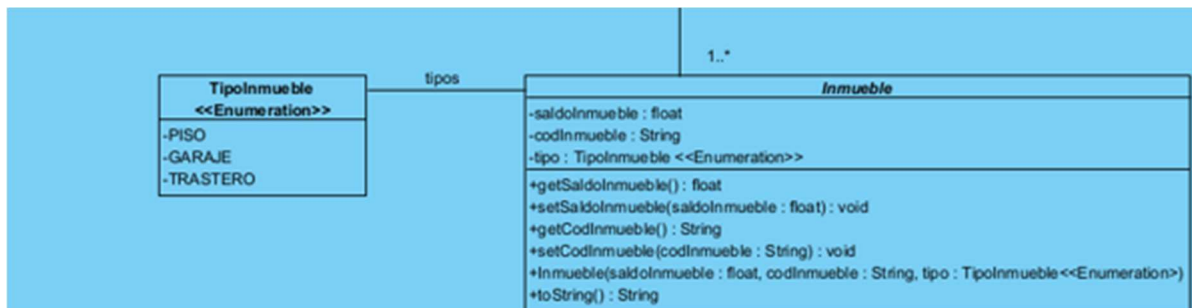
Un Vecino puede tener muchos Inmuebles.

Un Inmueble sólo puede ser de 1 Vecino, porque aunque el Inmueble tenga más de 1 Propietario, sólo 1 de ellos se podrá considerar Vecino, que es el que constará en Actas y tendrá opción a Voto, y es el que tendrá usuario y contraseña como usuario de la aplicación, aunque estas las pueda compartir con el resto de propietarios.



Inmueble-TipoInmueble: Asociación

Los Inmuebles pueden ser de 3 tipos, que serán un enumerado en Clase TipoInmueble.

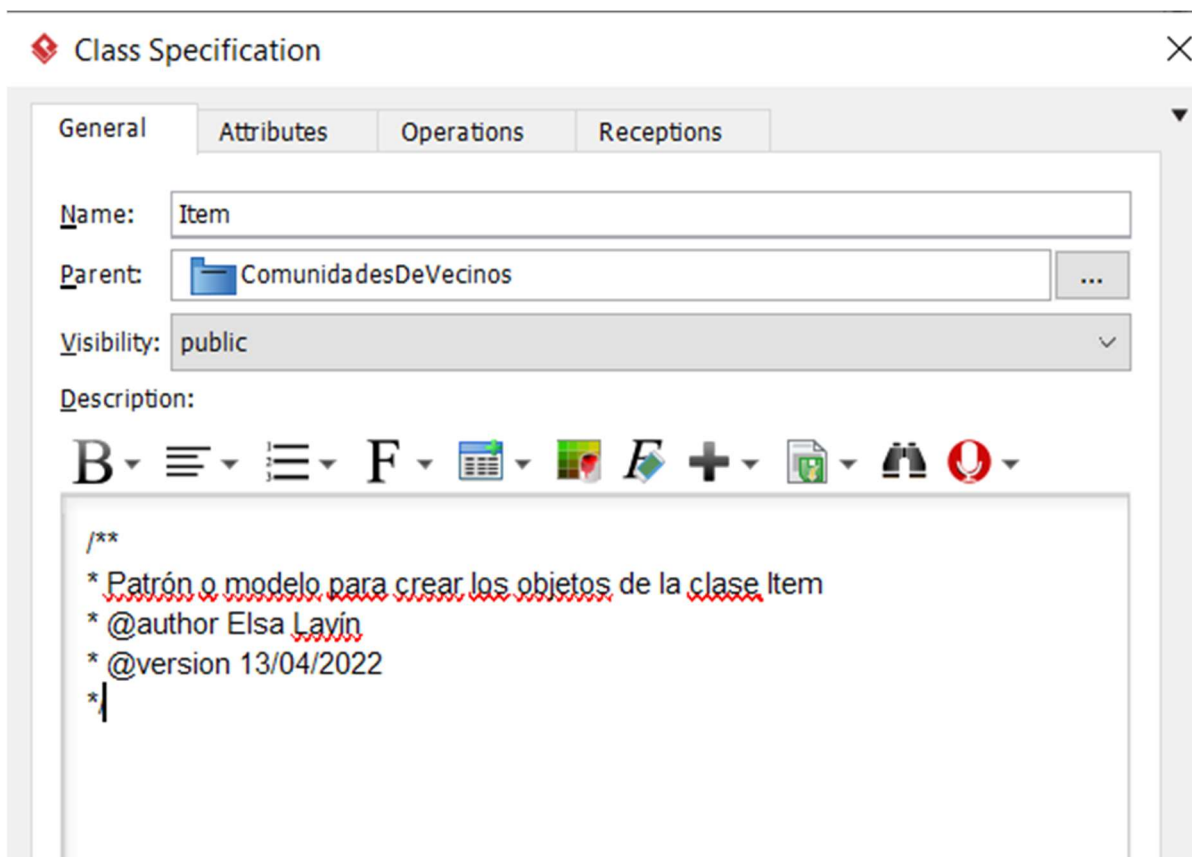


1.3 [Primer Refinamiento](#)

Como tratado en tutoría personal, el Refinamiento ha quedado realizado en la primera fase.

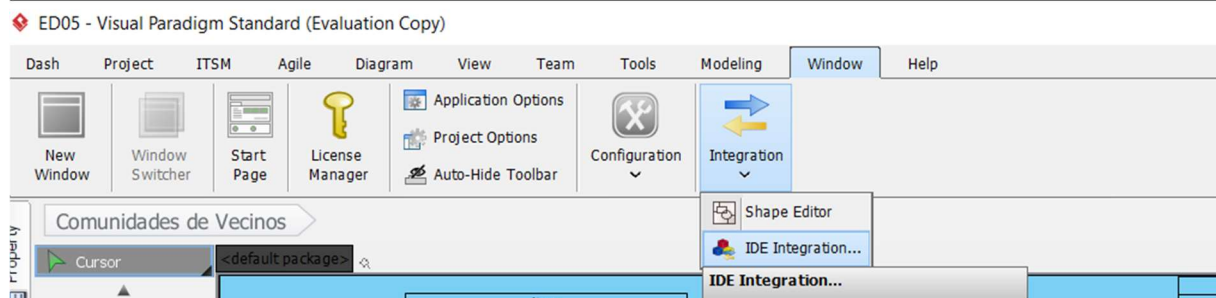
1.4 [Documentación](#)

Realizado en el propio diagrama con Visual Paradigm.

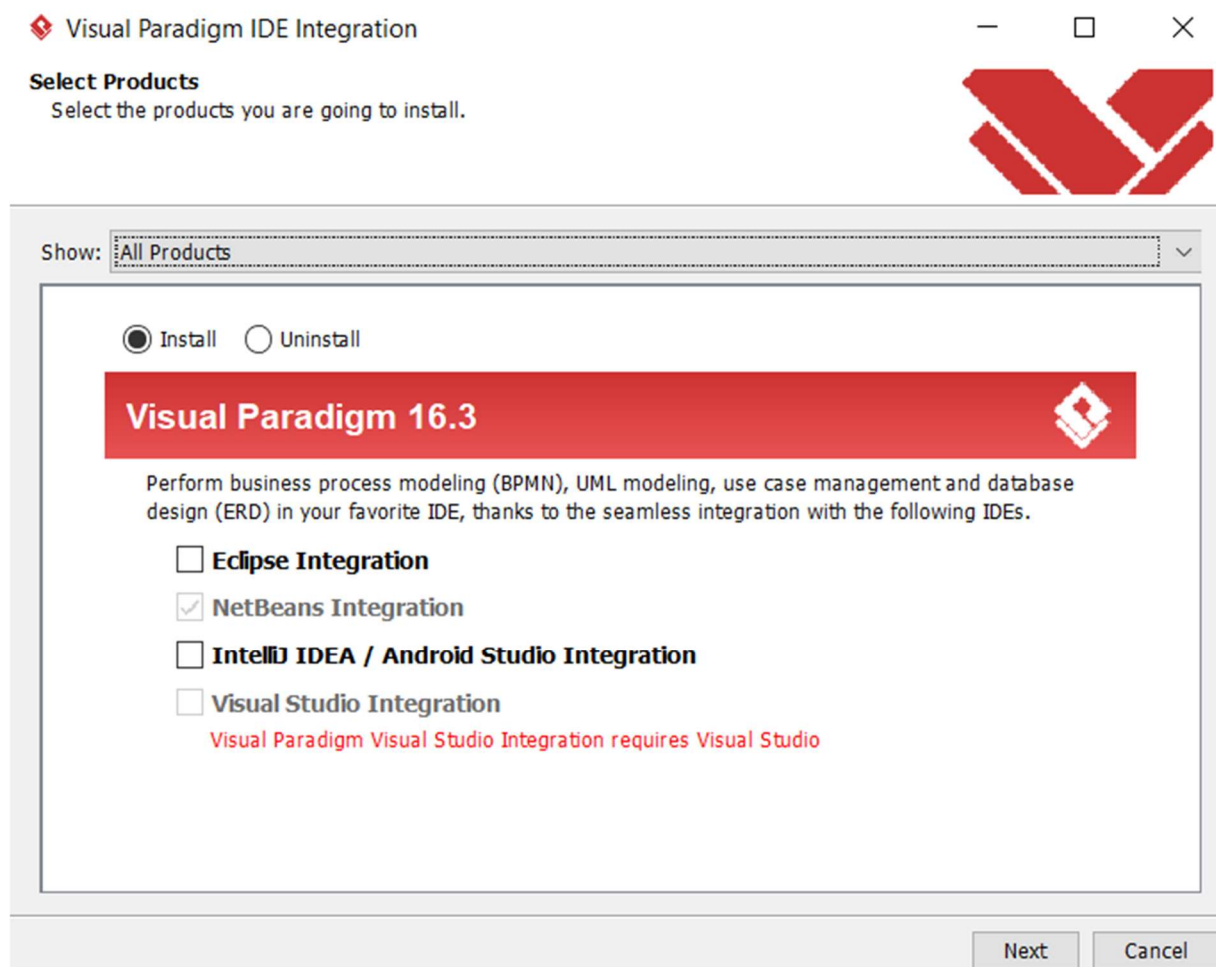


1.5 [Incluir el Proyecto VP-UML en un proyecto de NetBeans](#)

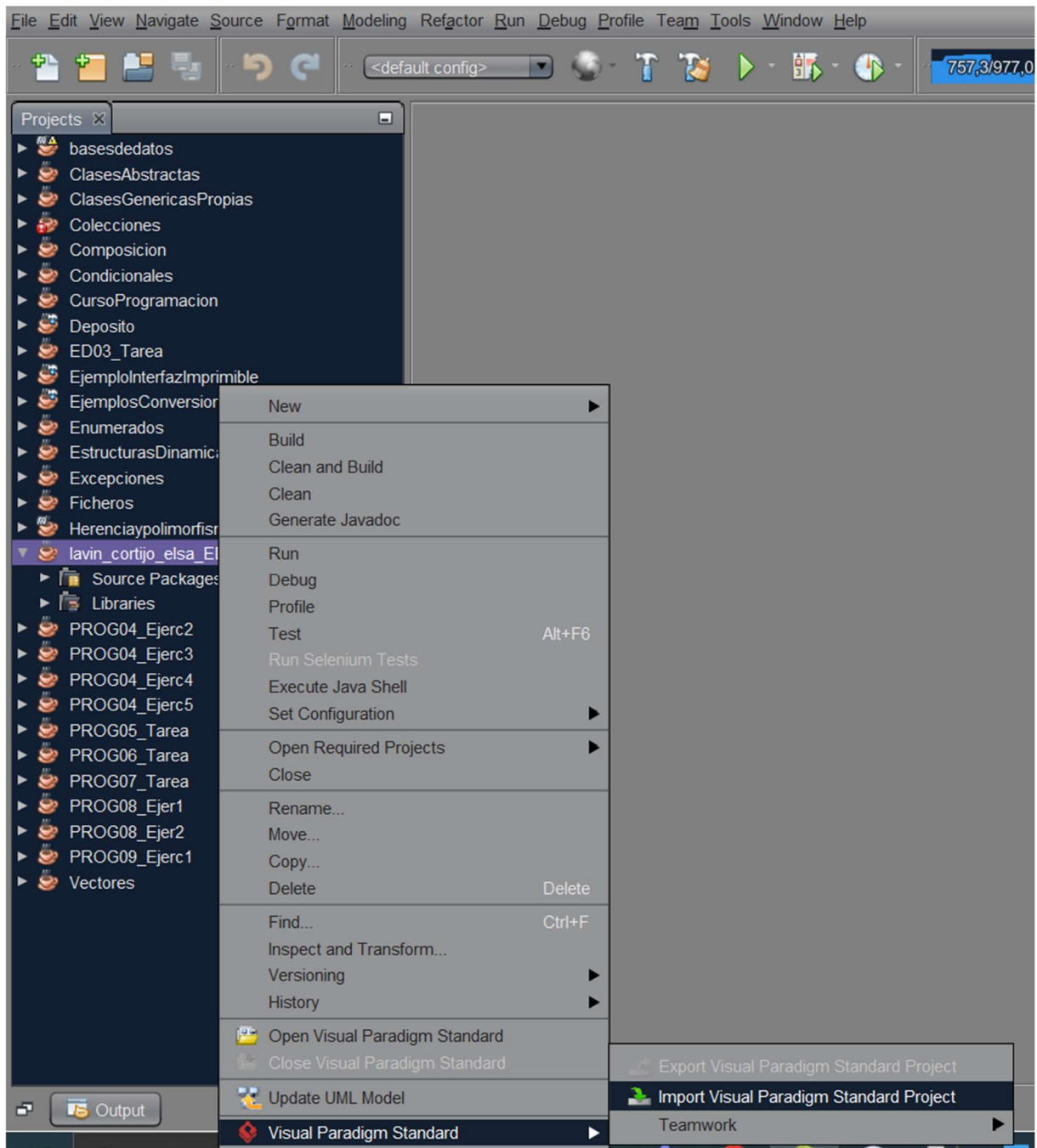
Para Integrar VisualParadigm y NetBeans, en Visual Paradigm, seleccionamos IDE Integration, en Window.



Seleccionamos NetBeans como entorno a integrar.



En el menú contextual del Proyecto de NetBeans seleccionamos importar el Proyecto de Visual Paradigm



Buscamos el proyecto en la ruta y lo importamos

