

tipo_moneda	
PK	<u>moneda Id Int NOT NULL</u>
	descripcion varchar(50) NOT NULL
	abreviatura varchar(5) NOT NULL

tipo_moneda	
PK	<u>moneda Id Int NOT NULL</u>
	descripcion varchar(50) NOT NULL
	abreviatura varchar(5) NOT NULL

Diagrama de bases de datos en formato de tabla que muestra las relaciones entre las tablas **barrio**, **localidad** y **provincia**.

Tabla barrio:

barrio	
PK	<u>barrio_id int NOT NULL</u>
	nombre varchar(50) NOT NULL
FK	localidad_id int NOT NULL

Tabla localidad:

localidad	
PK	<u>localidad_id int NOT NULL</u>
	nombre varchar(50) NOT NULL
FK	provincia_id int NOT NULL

Tabla provincia:

provincia	
PK	<u>localidad_id int NOT NULL</u>
	nombre varchar(50) NOT NULL

Tabla direccion:

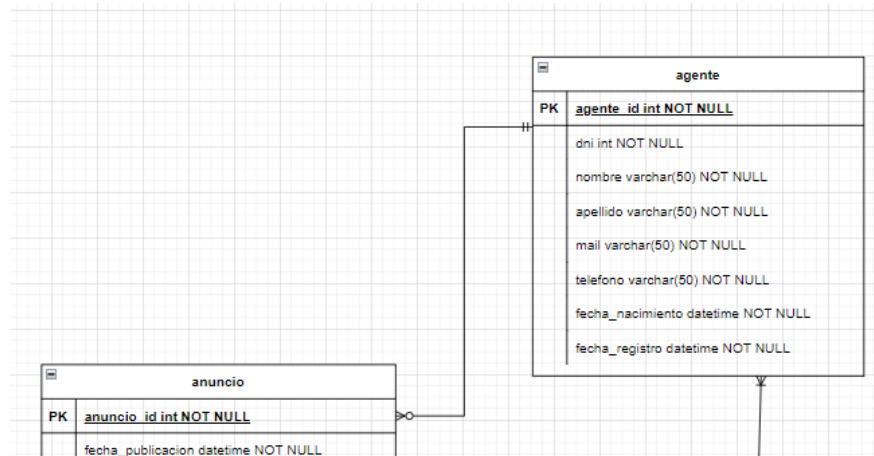
direccion	
PK	<u>direccion_id int NOT NULL</u>
	calle varchar(50) NOT NULL
	altura int NOT NULL
FK	barrio_id NOT NULL

Relaciones:

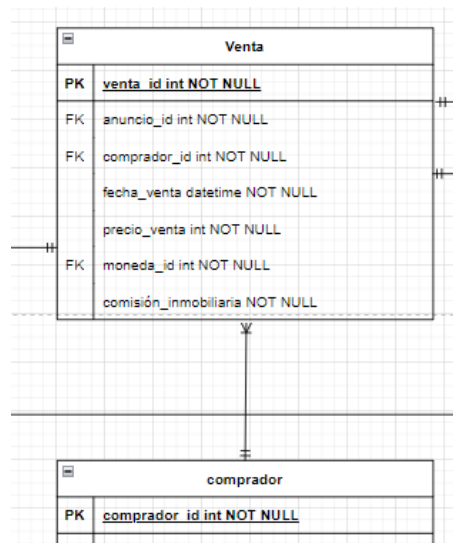
- Una línea vertical con una barra perpendicular en el extremo superior (provincia) y una flecha en el extremo inferior (localidad) indica una relación uno a muchos entre **provincia** y **localidad**.
- Una línea vertical con una barra perpendicular en el extremo superior (localidad) y una flecha en el extremo inferior (barrio) indica una relación uno a muchos entre **localidad** y **barrio**.
- Una línea vertical con una barra perpendicular en el extremo superior (barrio) y una flecha en el extremo inferior (direccion) indica una relación uno a muchos entre **barrio** y **direccion**.

pago_alquiler	
PK	<u>pago_id</u> int NOT NULL
FK	alquiler_id int NOT NULL
	fecha_pago date NOT NULL
	nro_periodo_pago int NOT NULL
	descripcion_periodo text
	fecha_inicio date NOT NULL
	fecha_fin date NOT NULL
	fecha_vencimiento date NOT NULL
	importe numeric NOT NULL
FK	medio_pago_id int NOT NULL

Decidimos realizar una relación one to many de agente a anuncio ya que un agente podría tener varios anuncios a su nombre, y determinamos que no sea obligatoria esta relación, es decir que un agente podría llegar a existir sin tener ningún anuncio a su nombre. Esta decisión la tomamos ya que pensamos que un agente podría estar inscripto en el sistema hace poco tiempo (en el caso en el que sea nuevo por ejemplo), y por lo tanto no habría concretado todavía ningún acuerdo ya sea de alquiler o venta.



Decidimos que sea obligatoria la relación de comprador a venta, pensando que no tendría sentido que exista un comprador cargado en la base de datos el cual no haya comprado nada



Hicimos entidades aparte para comprador, agente y propietario ya que, si bien comparten todos los atributos de las tablas, cada una de ellas representa una idea independiente, por lo que no sería conveniente juntarlas todas para que una persona pueda ser comprador, agente y propietario a la vez.

Tomamos la decisión de hacer una relación many to many de inmueble a característica, ya que las características de los inmuebles tienen la misma estructura pero solo cambia su descripción.

Consideramos que no tiene sentido modelarlo como columnas en inmuebles cuando se puede simplificar el diseño, legibilidad y rendimiento haciendo una relación many to many. Ya que creemos que las columnas pueden tender a confusión, poco orden en inmueble y creímos que si teníamos muchas características inmueble quedaría una entidad excesivamente grande. En este momento eso no pasa pero es posible que se agreguen características y de la forma anterior implicaría agregar muchas columnas por cada inmueble existente.

Además elegimos una relación many to many debido a que un inmueble puede tener muchas características y una característica puede estar en muchos inmuebles.

