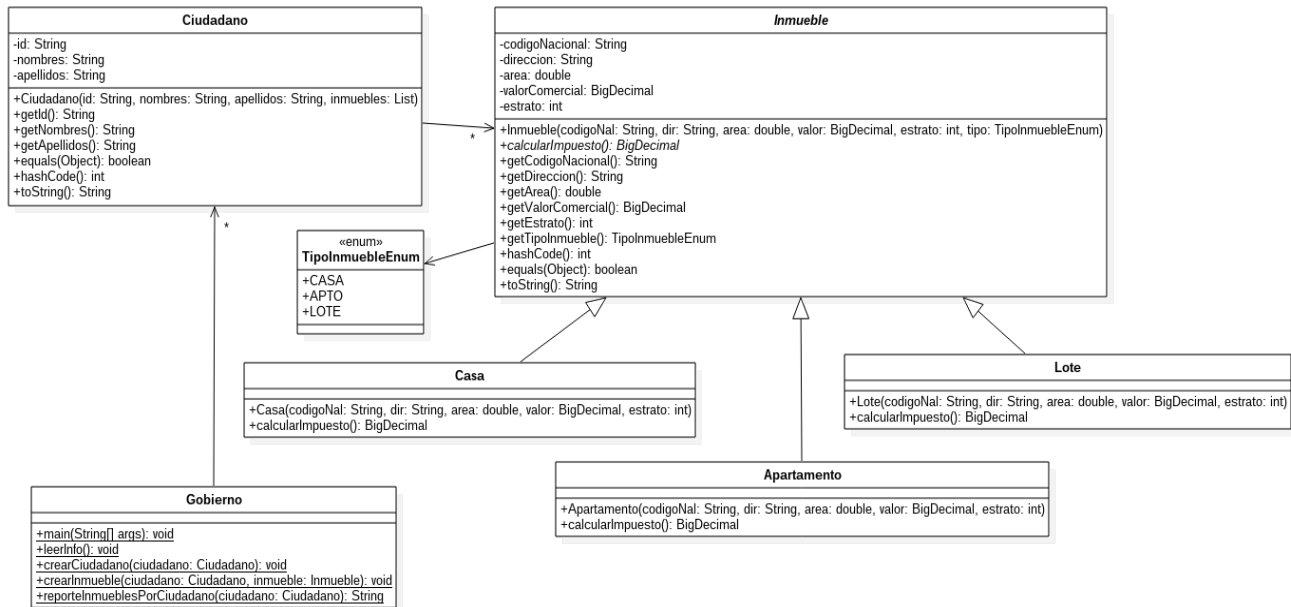


## TRABAJO FINAL JAVA BÁSICO GRUPO 1

El gobierno municipal requiere de un sistema que le permita llevar el control de los ciudadanos y sus inmuebles. A continuación se muestra el diagrama de clases requerido por el cliente para cumplir con sus necesidades:



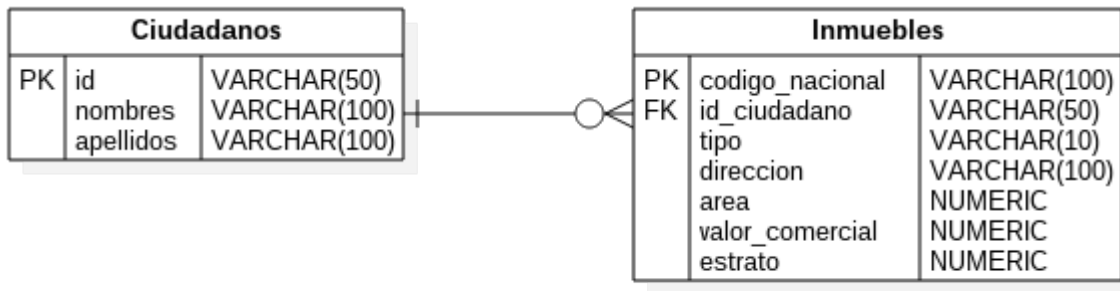
La clase inmueble es abstracta y su método `+calcularImpuesto():BigDecimal` es abstracto, dado que el cálculo depende del tipo de inmueble y sus clases hijas deben sobre-escribirlo así:

- Casa:  $\text{Impuesto} = \text{area} * \text{valorComercial} * 0.9$
- Apartamento:  $\text{Impuesto} = \text{area} * \text{valorComercial} * 0.8$
- Lote:  $\text{Impuesto} = \text{area} * \text{valorComercial}$

La clase Gobierno contiene el método main y se espera que desde allí se invoquen los demás métodos mostrados en dicha clase:

- `+leerInfo():void` → Debe cargar la información de la base de datos
- `+crearCiudadano(ciudadano:Ciudadano):void` → Debe crear el ciudadano recibido como param en la base de datos.
- `+crearInmueble(ciudadano:Ciudadano, inmueble:Inmueble):void` → Crear un nuevo inmueble y asociarlo al ciudadano en la base de datos
- `reporteInmueblesPorCiudadano(ciudadano:Ciudadano):String` → Del ciudadano recibido como param, debe retornar una cadena de texto separada por comas, con el siguiente formato para todos sus inmuebles, ordenados por estrato:
- `"idCiudadano,codigoNacional,area, estrato,valorComercial"`

El MER para la base de datos a implementar se muestra a continuación:



### Condiciones

- Fecha máxima de entrega Viernes 08 de Septiembre de 2017 23:00
- El trabajo se debe realizar en parejas
- Se calificará de 1 a 5, siendo 1 la menor nota y 5 la mayor nota.
- Este trabajo tiene un valor del 20% de su nota final
- Cada una de las opciones tiene un puntaje máximo de 1 punto. Para lograr el puntaje máximo, no basta solo con cumplir lo que se solicita en el enunciado del problema, sino que se revisará que se hayan utilizado los conceptos vistos durante el curso, tales como: Igualdad entre objetos, Expresiones Lambda, API Stream, Date and Time API, Try con recursos propios, etc.
- El trabajo debe ser subido a Github por uno de los miembros del equipo, de donde se tomará para ser calificado teniendo en cuenta la fecha de corte.
- La copia no está permitida, si se detecta código copiado de otro trabajo, se anularán ambos trabajos y se notificará a su jefe inmediato.