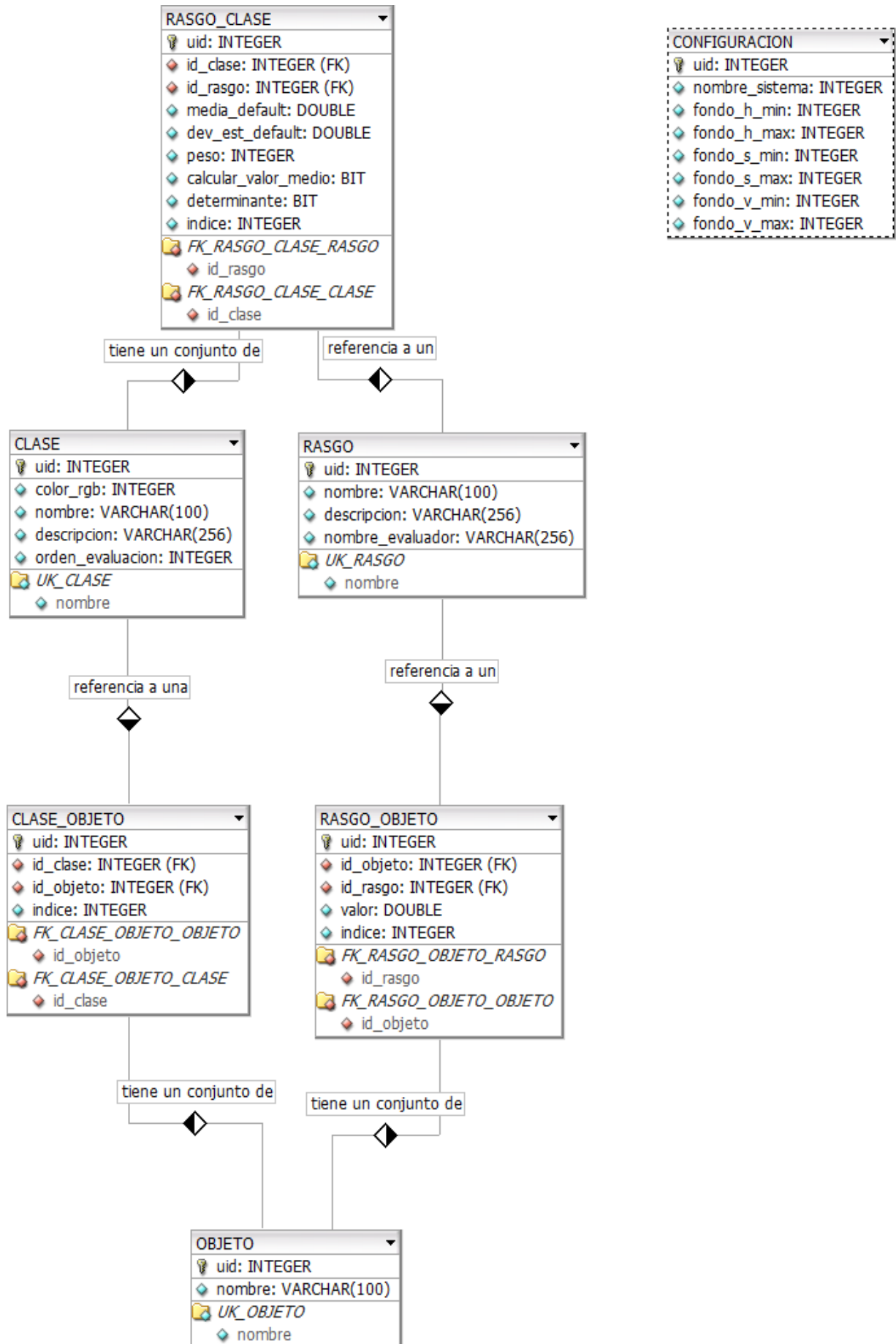


Diagrama de la base de datos



Tablas del sistema

CLASE: Tabla que guarda las clases que se van a identificar al realizar la clasificación. Por ejemplo: Maíz, Grano Quebrado, Grano Dañado, Materia extraña.

Nombre Columna	Descripcion
uid	Identificador numerico de una clase
nombre	Nombre unico de una clase
descripcion	Descripcion de la clase
color_rgb	Color para identificar la clase
orden_evaluacion	Determina el orden en que se va a evaluar la pertenencia de un objeto a una clase. Se evaluan las clases en orden creciente de orden de evaluacion. Primero se evalua si un objeto pertenece a la clase de orden 0, despues a la de orden 1, y asi sucesivamente.

RASGO: Tabla que guarda todos los rasgos que se pueden evaluar de un objeto. Por ejemplo volumen, diametro, circularidad, etc.

Nombre Columna	Descripcion
uid	Identificador numerico del rasgo
nombre	Nombre que identifica al rasgo
descripcion	Descripcion del rasgo
nombre_evaluador	Nombre de la clase java encargada de evaluar el valor del rasgo en un objeto. Debe el nombre completo de una clase java que herede de la clase EvaluadorRasgo

RASGO_CLASE: Tabla que guarda los rasgos que se deben evaluar para a una clase dada. Asociacion entre un rasgo y una clase.

Nombre Columna	Descripcion
uid	Identificador numerico del rasgo de una clase
id_clase	Identifiicador de la clase
id_rasgo	Identificador del rasgo
media_default	Valor medio por defecto cuando no hay datos para poder calcularlo desde otros objetos pertenecientes a la clase
dev_est_default	Desvio estandar por defecto cuando no hay datos para poder calcularlo desde otros objetos pertenecientes a la clase
peso	Determina que rasgo tiene mas peso que otro dentro de una clase
calcular_valor_medio	Valor booleano. Indica si se debe tener en cuenta el valor medio y el desvio estandar para determinar si un objeto entra dentro del rango del rasgo para la clase. Si es false se usa el maximo y minimo valor encontrado para los objetos de la clase
determinante	Indica si el rasgo es determinante para la pertenencia de un objeto a la clase. Si es false se calcula el valor del rasgo para el objeto, pero solo para mostrarlo, no para determinar si pertenece o no a la clase.
indice	Indice usado para propositos de Hibernate para mapear el objeto a una lista. Determina la posicion dentro de la lista

OJBETO: Tabla que guarda los objetos que se han clasificado.

Nombre Columna	Descripcion
uid	Identificador numerico de un objeto. Es autoincremental
nombre	Nombre unico de un objeto

RASGO_OBJETO: Guarda el valor de un rasgo para un objeto clasificado. Por ejemplo el volumen, diametro, etc. de un objeto.

Nombre Columna	Descripcion
uid	Identificador numerico de un rasgo de un objeto
id_objeto	Identificador del objeto
id_rasgo	Identificador del rasgo
valor	Valor del rasgo para el objeto
indice	Indice usado para propositos de Hibernate para mapear el objeto a una lista. Determina la posicion dentro de la lista

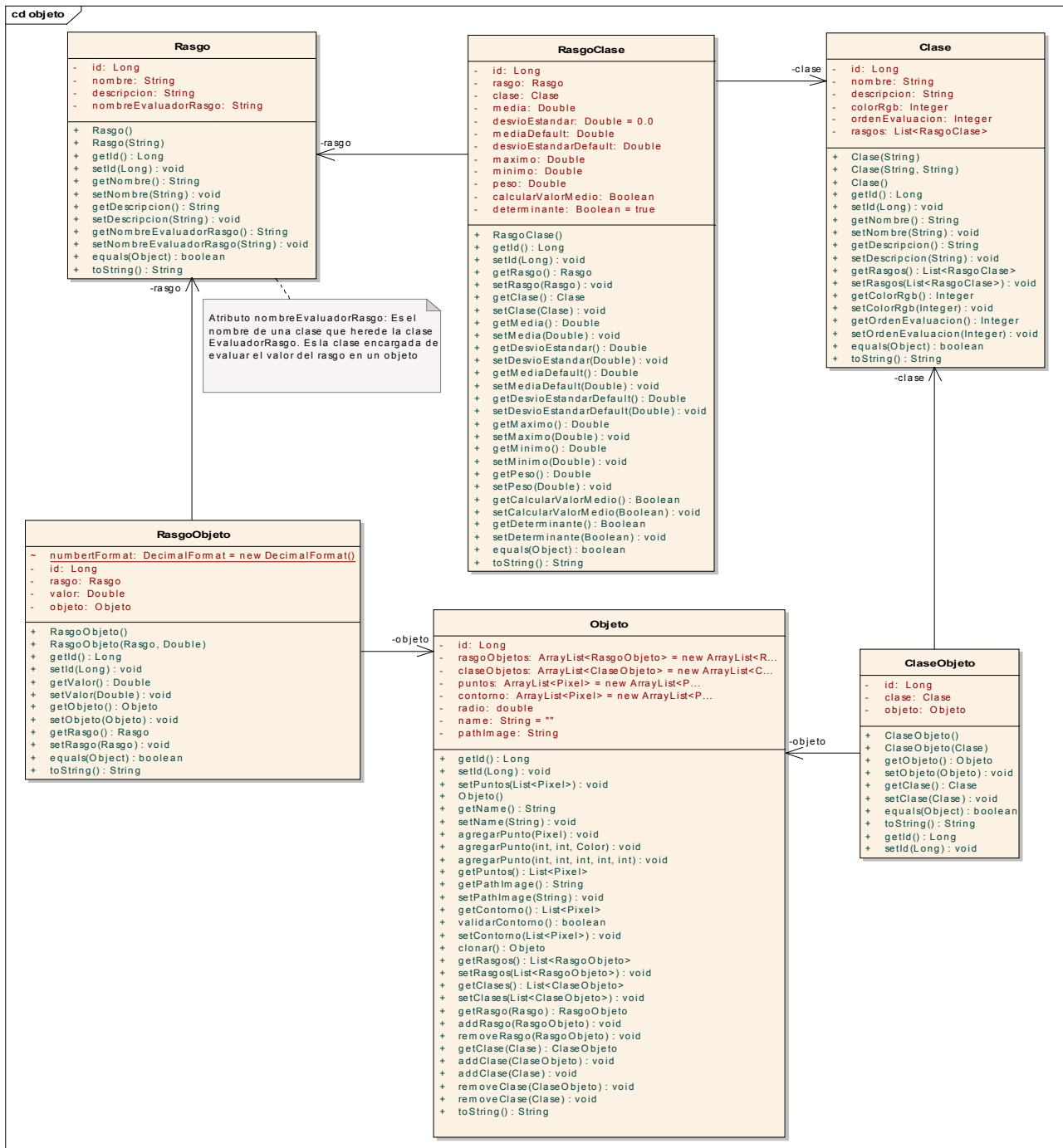
CLASE_OBJETO: Tabla que guarda la asociacion de una clase a un objeto. Indica de que clase es el objeto.

Nombre Columna	Descripcion
uid	Identificador numerico de la asociacion entre una clase y un objeto
id_objeto	Identificador del objeto
id_clase	Identificador de la clase
indice	Indice usado para propositos de Hibernate para mapear el objeto a una lista. Determina la posicion dentro de la lista

CONFIGURACION: Tabla que guarda valores de configuracion generales para la clasificacion como por ejemplo el color del fondo de la imagen, etc.

Nombre Columna	Descripcion
uid	Identificador numerico de la configuracion de un sistema
nombre_sistema	Nombre unico que identifica a un sistema
fondo_h_min	Valor minimo del canal H del color de fondo de la imagen
fondo_h_max	Valor maximo del canal H del color de fondo de la imagen
fondo_s_min	Valor minimo del canal S del color de fondo de la imagen
fondo_s_max	Valor maximo del canal S del color de fondo de la imagen
fondo_v_min	Valor minimo del canal V del color de fondo de la imagen
fondo_v_max	Valor maximo del canal V del color de fondo de la imagen

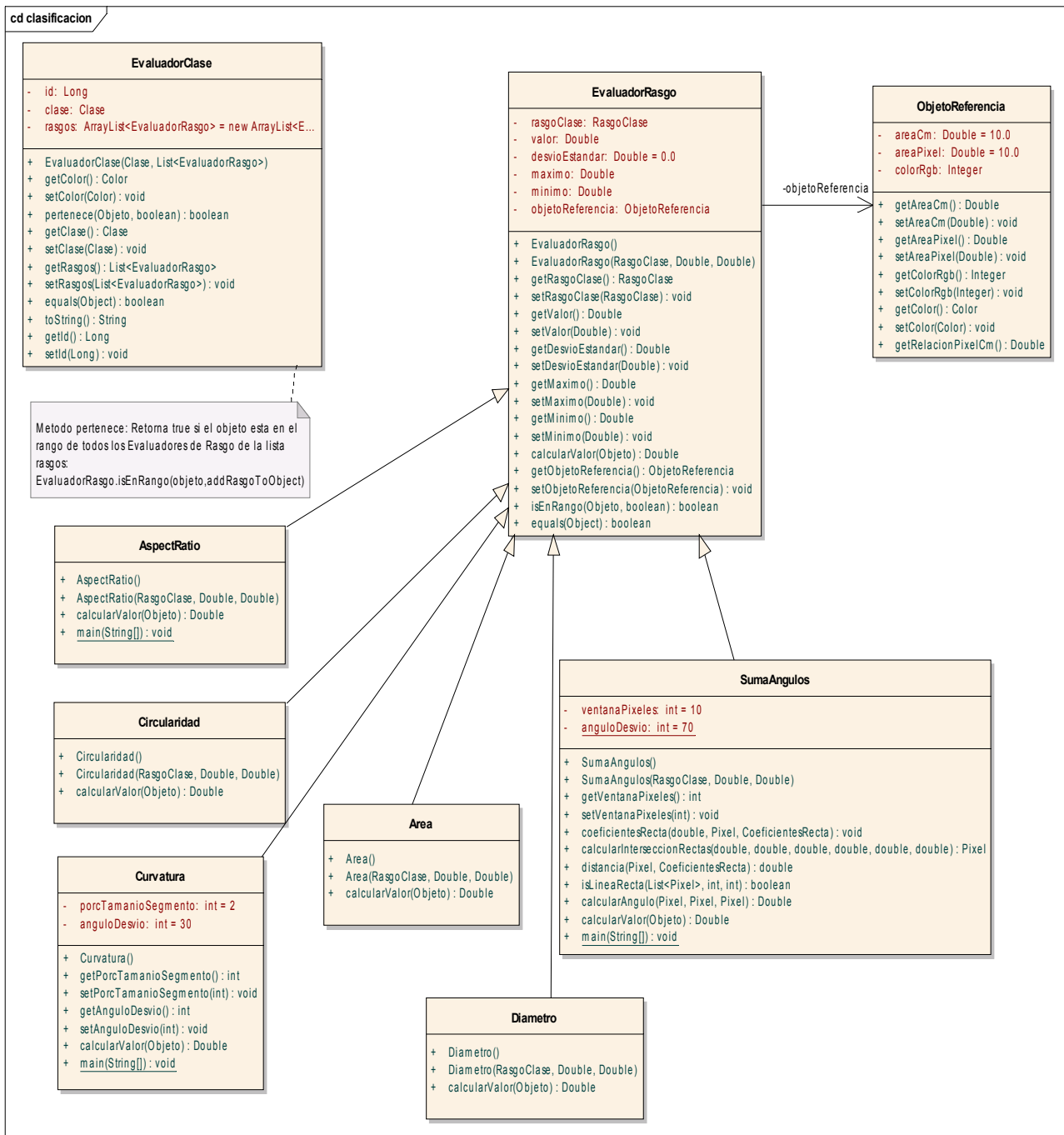
Diagrama de clases



Existe un mapeo entre estas clases y las tablas de la base de datos:

Nombre Clase	Nombre Tabla
Clase	CLASE
Rasgo	RASGO
RasgoClase	RASGO_CLASE
Objeto	OBJETO
RasgoObjeto	RASGO_OBJETO
ClaseObjeto	CLASE_OBJETO

Clase EvaluadorClase y EvaluadorRasgo



Se construye un objeto EvaluadorClase por cada instancia de la clase Clase existente. Un EvaluadorClase posee una lista de EvaluadorRasgo. Se crea un objeto EvaluadorRasgo por cada objeto RasgoClase de una instancia de Clase. Un objeto pertenece a una clase si el metodo pertenece(..) de la instancia de EvaluadorClase correspondiente retorna true para ese objeto. Si un objeto no pertenece a ninguna clase entonces se lo asigna a la clase especial INDETERMINADO.

La clase Clasificador



La clase Clasificador se encarga de recuperar todas las clases junto con sus rasgos almacenados en la base de datos y de inicializar una nueva clasificación. Para ello guarda una variable de instancia *clasificacion* de tipo HashMap que es una asociación entre las clases existentes y los objetos clasificados pertenecientes a esa clase. El método *inicializarClasificacion* se encarga de inicializar esta tabla de hash. Este metodo hace lo siguiente:

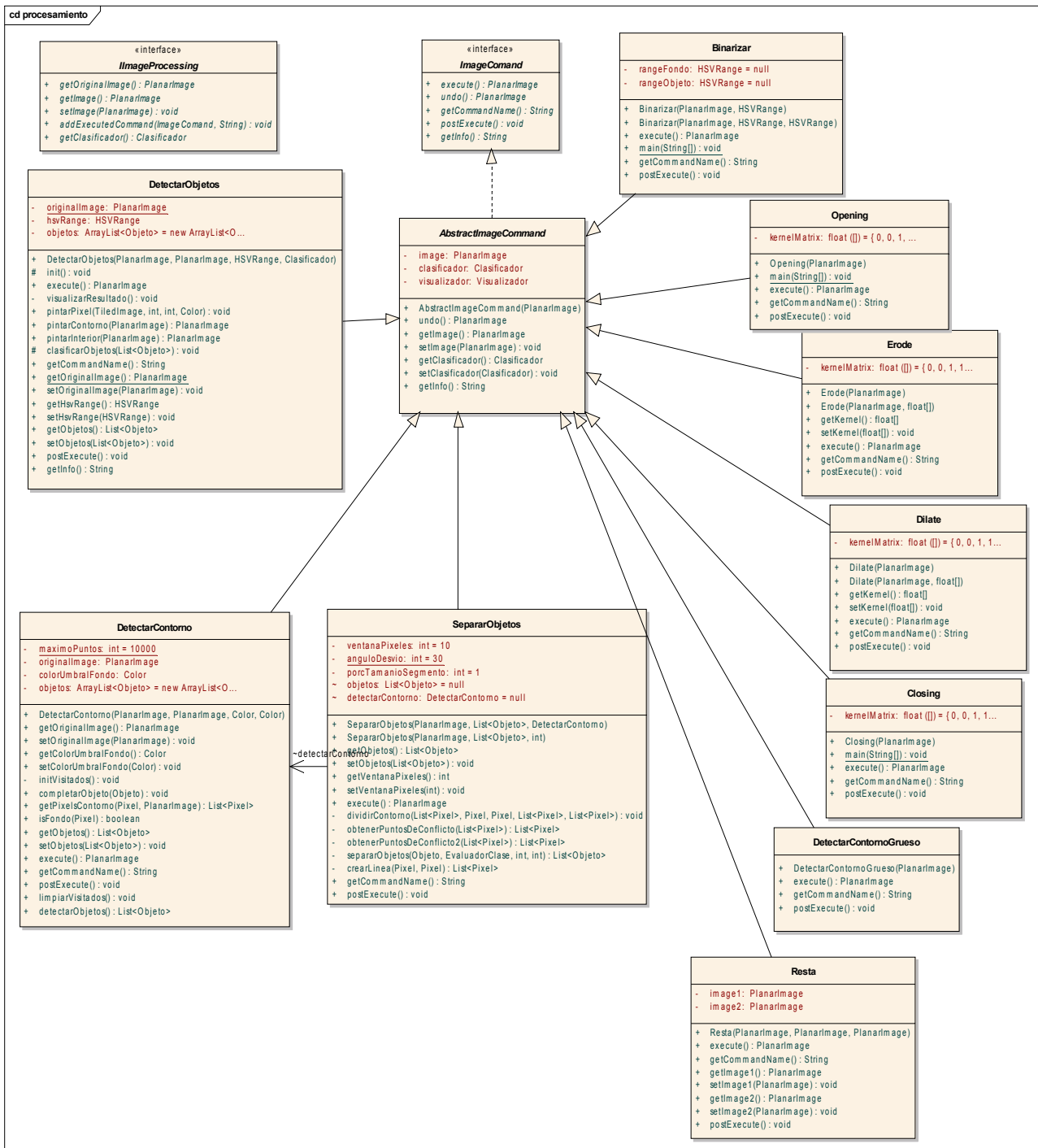
1. Recupera todas las clases de la base de datos, excepto la clase INDETERMINADO.
2. Por casa clase construye el objeto EvaluadorClase correspondiente:
 1. Por cada Rasgo de la Clase construye el objeto EvaluadorRasgo correspondiente (`Rasgo.getNombreEvaluadorRasgo()`) y lo agrega a la colección evaluadores de rasgo del objeto EvaluadorClase.

Una instancia de la clase Clasificacion tambien tiene una referencia a la instancia del objeto Configuracion del sistema.

La clase clasificación tambien posee un método encargado de persistir la clasificación en la base de datos: *guardarClasificacion()*.

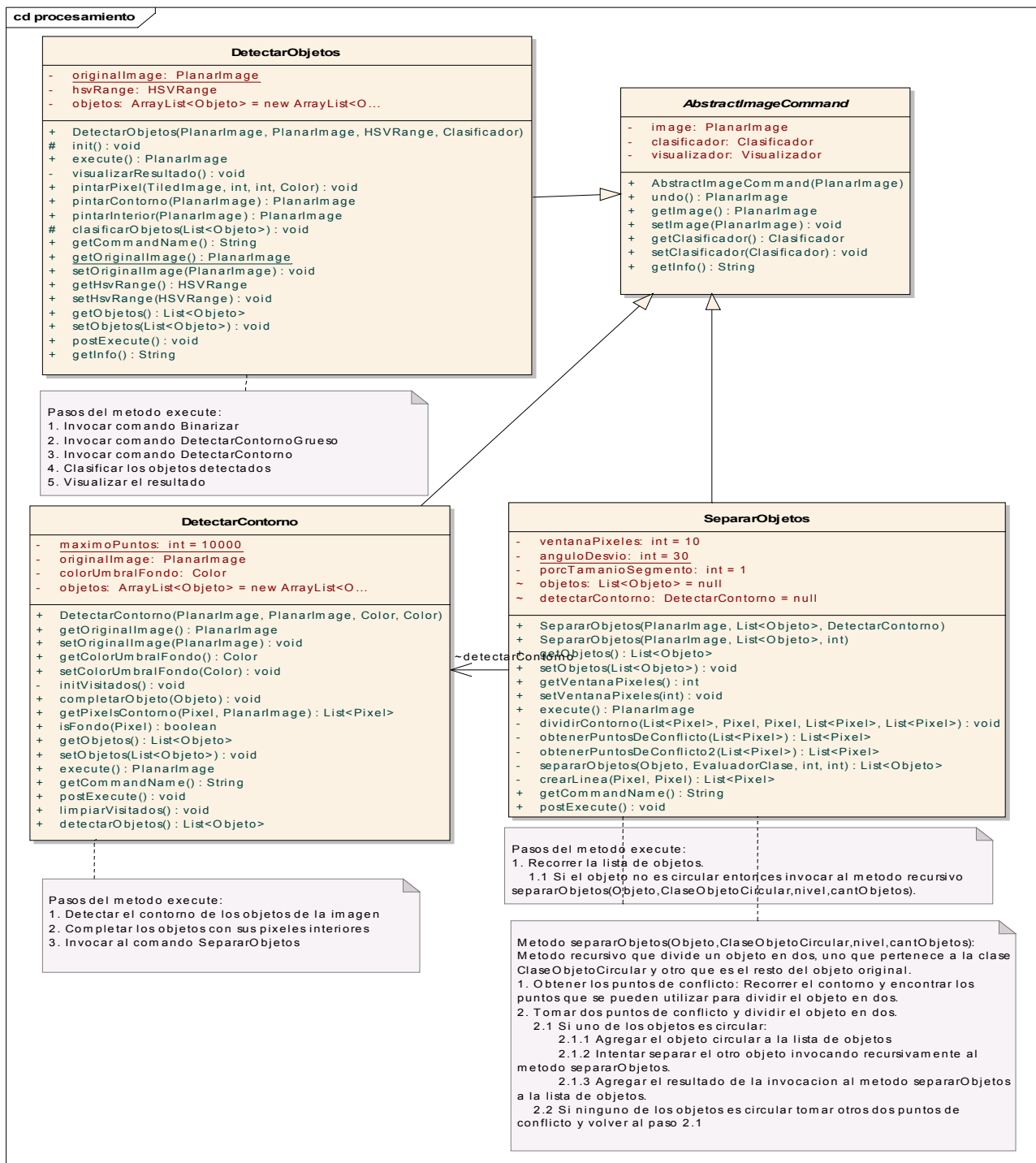
Procesamiento de imágenes

Las clases para realizar el procesamiento de las imágenes se estructuran bajo el patrón Command, para permitir desde la interfaz de usuario ejecutar un comando, y también deshacerlo.



La interfaz *ImageCommand* define los métodos para ejecutar (*execute()*) y deshacer (*undo()*) un comando además de otros. La clase abstracta *AbstractImageCommand* implementa la interfaz *ImageCommand* y guarda las referencias a la imagen y el objeto Clasificador. Para cada operación que se pueden realizar sobre una imagen existe una clase que extiende de *AbstractImageCommand*. Estas operaciones pueden ser simple como Erode, Dilate, Opening, Closing, etc., o mas complejas como DetectarObjetos o DetectarContorno.

La clase DetectarObjetos



La clase principal para realizar la clasificación de los objetos en una imagen es DetectarObjetos. En su método *execute()* define el algoritmo para procesar la imagen:

1. Inicializar el objeto Clasificacion: llama al metodo *inicializarClasificacion()* del objeto Clasificador.
2. Invocar al comando Binarizar: Binariza la imagen dejando en color negro el fondo y en color blanco los objetos.
3. Invocar al comando DetectarContornoGrueso: Basandose en la imagen binarizada este comando genera una imagen con el contorno de los objetos en color blanco, pero este contorno puede ser de mas de un pixel de espesor.

4. Invocar al comando DetectarContorno: Basandose en la imagen con el contorno de los objetos, este comando se encarga de recorrer el contorno de cada objeto y crear la instancia de la clase Objeto correspondiente. Por cada objeto detectado se evalua su circularidad y en caso de no serlo se intenta separarlo en uno o más objetos. Esto se hace invocando al comando SepararObjetos.
5. Clasificar los objetos detectados: Por cada objeto detectado se evalua si pertenece a alguno de los objetos EvaluadorClase creados. Si pertenece a alguno se asigna el objeto a esa clase.
6. Visualizar el resultado: Se invoca a la pantalla que muestra el resultado de la clasificación mostrando los objetos asignados a cada clase.

Diagrama de secuencia:

