# The Speaker (3era Entrega)

Juan Manuel Barrenche, *Padrón Nro. 86.152*
snipperme@gmail.com
Martín Fernández, *Padrón Nro. 88.171*
tinchof@gmail.com
Marcos J. Medrano, *Padrón Nro. 86.729*
marcosmedrano0@gmail.com
Federico Valido, *Padrón Nro. 82.490*
fvalido@gmail.com

Grupo Nro. 11 (YES)
Ayudante: Renzo Navas
1er. Cuatrimestre de 2009
75.06 Organización de Datos - Titular: Arturo Servetto
Facultad de Ingeniería, Universidad de Buenos Aires

Domingo 24 de Mayo de 2009

**Resumen**

Breve descripción de la arquitectura a utilizar para la 3era entrega del trabajo práctico del curso de *75.06 Organización de Datos* de la cátedra Servetto.
Se detallan las clases e interfaces principales y su interacción, pero no se hace referencia a la arquitectura utilizada en las entregas anteriores ya que el punto de contacto con la nueva funcionalidad es mínimo. Este documento ha sido desarrollado en LaTeX.

# 1. The Big Picture

La idea principal es implementar los compresores como **Serializadores**. En nuestra arquitectura actual, los serializadores son utilizados en todos los manejadores de archivos (VLFM, StraightFM,...) para hidratar y deshidratar objetos.
Implementar los compresores de esta manera tiene la ventaja de no tener que modificar nuestra arquitectura actual, que actualmente está funcionando correctamente. Simplemente se reemplazan serializadores actuales por los nuevos serializadores que serán capaces de comprimir, en nuestro caso, los Documentos.

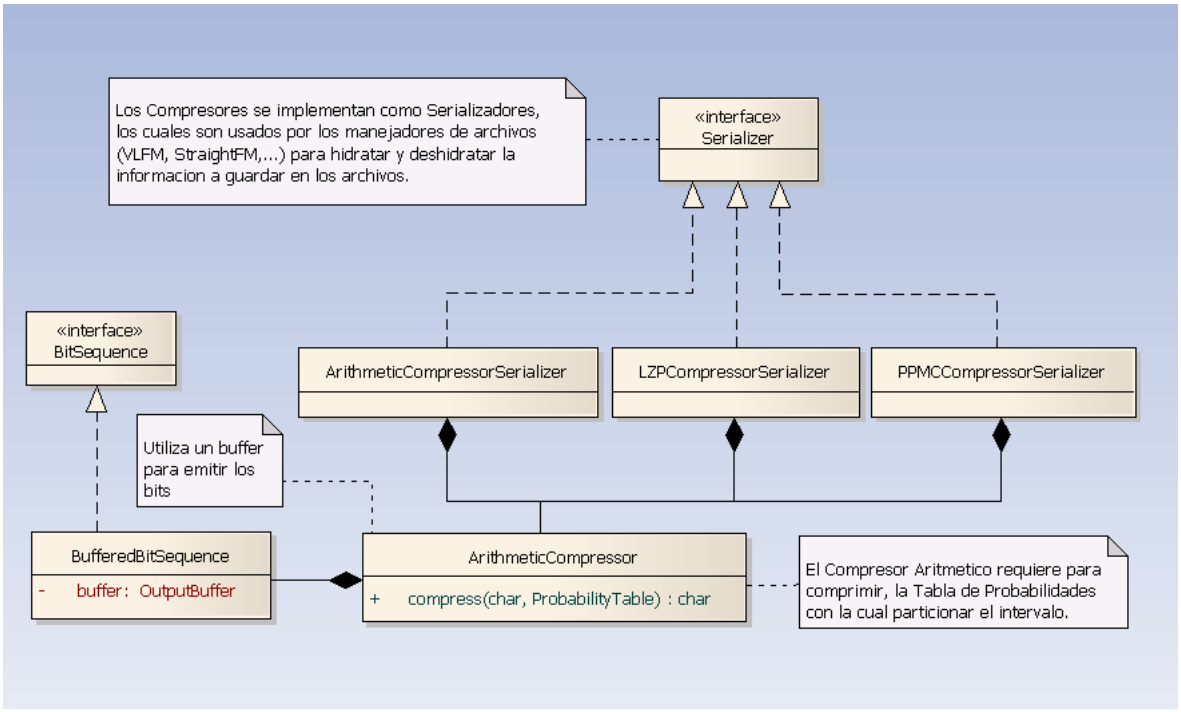


Figura 1: Diagrama de clases general de la arquitectura propuesta

# 2. Compresor Aritmético

La clase *ArithmeticCompressor* es el compresor aritmético *per se*. Encargado de determinar que bits deben ser emitidos de acuerdo a la tabla de probabilidades que se le entrega y al caracter que se le indica que corresponde ser emitido. Dado que nuestros buffers trabajan de a bytes el compresor delega en un wrapper de dicho buffer que permite ingresar la información de a un bit por vez (y que luego emite de a bytes compltos). Luego, el compresor va iterando sobre los caracteres de la tabla (Que, si bien se muestran como char, esta clase es un wrapper de la clase Character de Java, ver mas adelante) y calcula los pisos y techos para cada caracter hasta que encuentre que el caracter a emitir *matchea* con el caracter iterado. Luego de verificar si hay Overflow o Underflow, y emitir lo que corresponda, actualiza su piso y techo para estar listo para un nuevo caracter a emitir.

El proceso de descompresión es el proceso inverso, en base a la tabla de probabilidades va armando los pisos y techos del caracter actual hasta que encuentra que la información de su buffer se encuentra dentro de dicho rango. Actualiza Overflow y Underflow, si corresponde, y devuelve el caracter correspondiente al nuevo rango para que el que el usuario del compresor pueda actualizar contextos y demás tareas.

### 2.1. La clase Char

La clase *Char* permite tener valores fuera de los reales en un texto, es decir, el caracter **EOF**, y el caracter de **Escape**, que utiliza PPMC. Este último caracter tiene la particularidad que matchea con cualquier caracter, de manera que, en PPMC, estando en un contexto en el que no existe el caracter buscado, cuando el *ArithmeticCompressor* verifica si el caracter matchea siempre va a encontrar que el de Escape si matchea. Es por esto que el compresor debe devolver el caracter de la tabla que encontró matcheo con el recibido para que el PPMC pueda actuar según corresponda.

## 3. PPMC

En cuanto a la arquitectura para el PPMC no hay mucho que decir, simplemente la clase PPMC se encarga de mantener las frecuencia de ocurrencia de los caracteres de cada contexto, las cuales se encuentran encadenadas de acuerdo a subniveles para facilitar la búsqueda, y de recordar cual es el contexto actual para pedirle al compresor aritmético que emita con dicha tabla de probabilidades. Cuando llama al aritmético arma la tabla de probabilidades excluyendo, si corresponde, los caracteres que otro contexto ya verificó en su tabla de frecuencias.

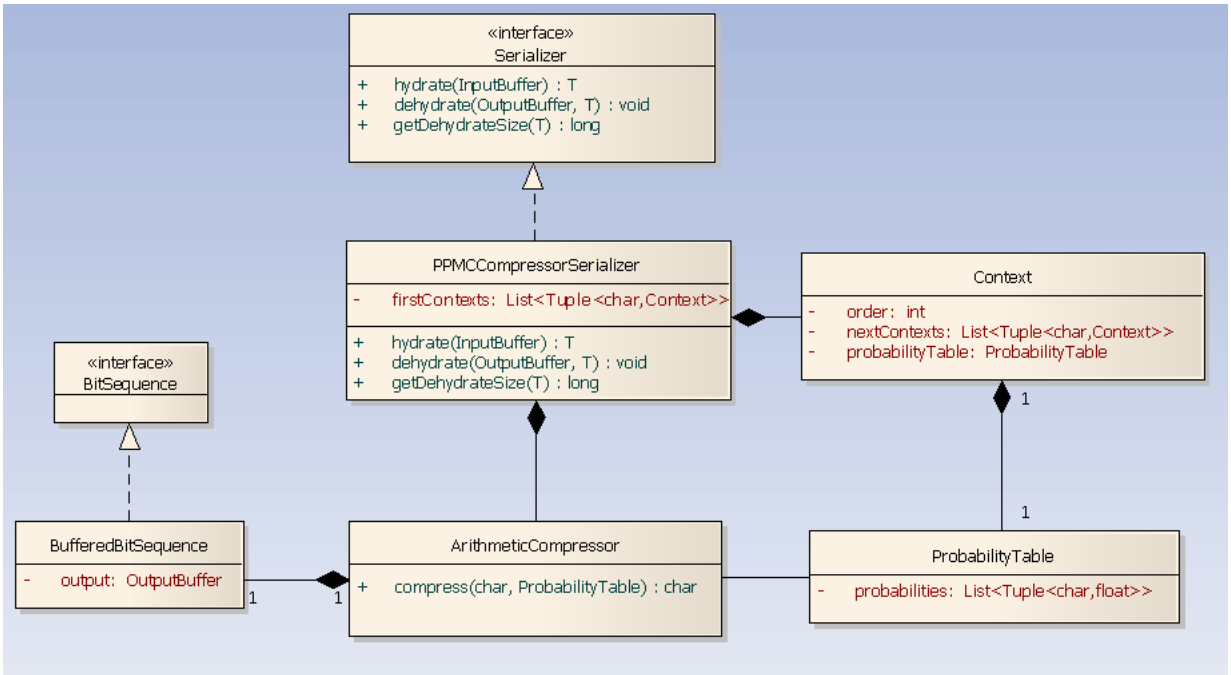


Figura 2: Diagrama de clases de la arquitectura para PPMC