# Cuantización Vectorial (VQ: Vector Quantization)

La idea de **Cuantización Vectorial** en aplicaciones de Reconocimiento de Voz es representar a los vectores característicos (por ejemplo en el dominio cepstral), que pueden tomar infinitos valores en un continuo, por un conjunto finito de vectores que constituyen lo que se denomina un **Libro de Código**. Los vectores del LC proveen una única representación espectral de cada unidad básica de voz (*e.g.*, fonemas).

Con esta representación se logra una reducción considerable de la tasa de información. Por ejemplo: señal de voz muestreada a 10 Kz, y codificada las amplitudes con 16 bits $\Rightarrow$ 160 Kbps de tasa de información. Considerando ahora vectores cepstral de dimensión Q=10 y con 100 cuadros por segundo como tasa de frames, resulta



una tasa de información de 100 x 10 x 16 = 16 Kbps, que representa una reducción de 10-a-1 en la tasa de información. Supongamos que se genera un libro de código de 1024 vectores espectrales (aproximadamente 20 variantes de cada uno de los aprox. 50 fonemas básicos). Luego sólo se necesita un número de 10 bits para indicar el índice en el LC. Asumiendo una tasa de frame de 100 frames por seg., resulta en una tasa de bit de 1000 bps, que es una reducción de 16-a-1 de la tasa requerida usando vectores continuos. Esta es una de las principales razones para la utilización de VQ.

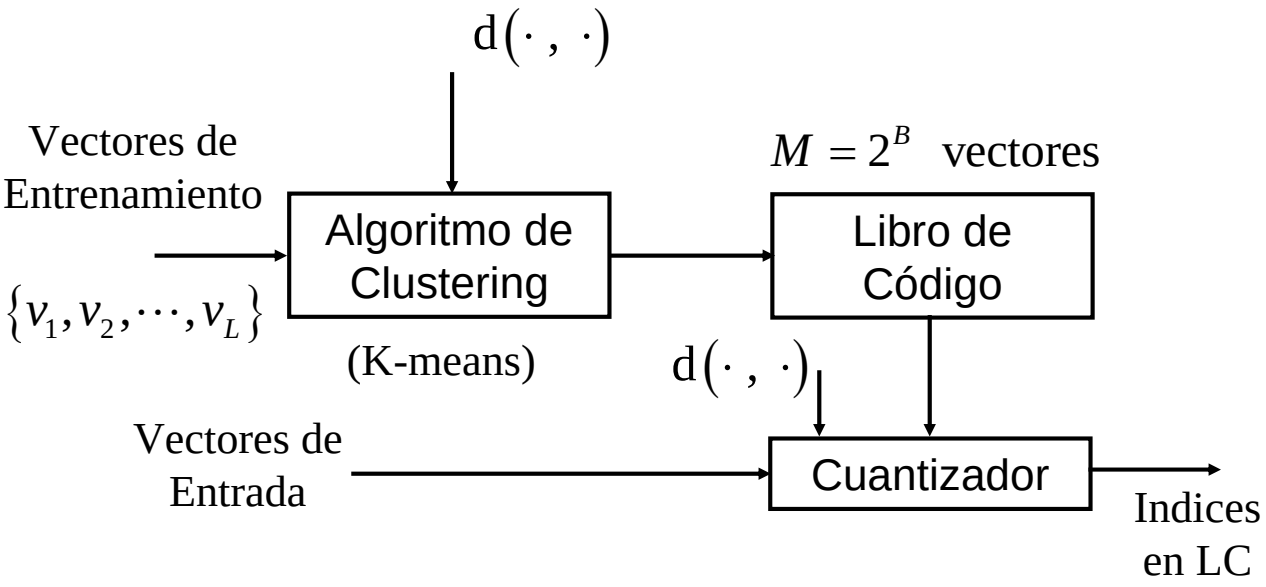


Fig. 1: Diagrama de Bloques de un cuantizador vectorial.



# Algoritmos de Clustering

Durante la etapa de **entrenamiento** en el enfoque de reconocimiento por comparación de patrones se deben generar $M$ patrones de referencia basándose en $L$ vectores de entrenamiento, donde $M < L$ . Al conjunto de patrones de referencia se lo denomina **libro de códigos** y a los patrones, **palabras de código.** Los algoritmos que permiten clasificar un conjunto de $L$ vectores de entrenamiento en $M$ clases distintas (o **clusters**) caracterizadas por un vector (el **centroide** de cada cluster) se denominan **algoritmos de clustering**.

Un algoritmo muy utilizado es el algoritmo de Lloyd generalizado o el **algoritmo de clustering de K-means** (o K-vecinos). El algoritmo puede resumirse en los siguientes pasos.

## Algoritmo de K-means

1. **Inicialización:** Se eligen arbitrariamente $M$ vectores del conjunto de $L$ vectores de entrenamiento como conjunto inicial de vectores (palabras código) del libro de códigos.
2. **Búsqueda del vecino más cercano:** para cada vector de entrenamiento se busca el vector en el libro de códigos actual que sea el más cercano (de acuerdo a la medida de distancia que se esté empleando) y se asigna el vector de entrenamiento a la celda correspondiente al vector del libro de código más cercano.
3. **Actualización de centroides:** se actualiza el vector asociado a cada celda por el centroide de los vectores de entrenamiento asignados a esa celda.
4. **Iteración:** Se repiten los pasos 2 y 3 hasta que la distancia promedio cae por debajo de un umbral pre-determinado.



# Algoritmos de Reconocimiento

La etapa de **reconocimiento** del enfoque basado en comparación de patrones consiste básicamente en una búsqueda exhaustiva en el libro de códigos para hallar la palabra código que mejor se ajusta al vector característico asociado a la palabra que se quiere reconocer.

Si denotamos con $\mathbf{y_m}$ , $1 \le m \le M$ a los vectores de un libro de código con $M$ palabras código, y con $\mathbf{v}$ al vector que se quiere reconocer, entonces el índice $m^*$ correspondiente a la **mejor** palabra en el libro de código es

$$m^* = \arg\min_{1 \le m \le M} d(\mathbf{v}, \mathbf{y_m})$$