

**U.B.A. FACULTAD DE INGENIERÍA**

**Departamento de Computación**

**Organización de Datos 75.06  
Informática**

**TRABAJO PRÁCTICO N°1**

*PAGINAS DORADAS*

**Curso: Servetto**

**2013 - 2do Cuatrimestre**

<b>GRUPO - DRINK TEAM</b>	
<b>APELLIDO, Nombres</b>	<b>N° PADRÓN</b>
<b>Baracat, Juan Manuel</b>	<b>92775</b>
<b>Duarte, Matias Ariel</b>	<b>92186</b>
<b>Funes Clapier, Quimey</b>	<b>92324</b>
<b>Rodriguez, Mariano</b>	<b>94409</b>
<b>Calificación :</b>	
<b>Firma de Aprobación :</b>	

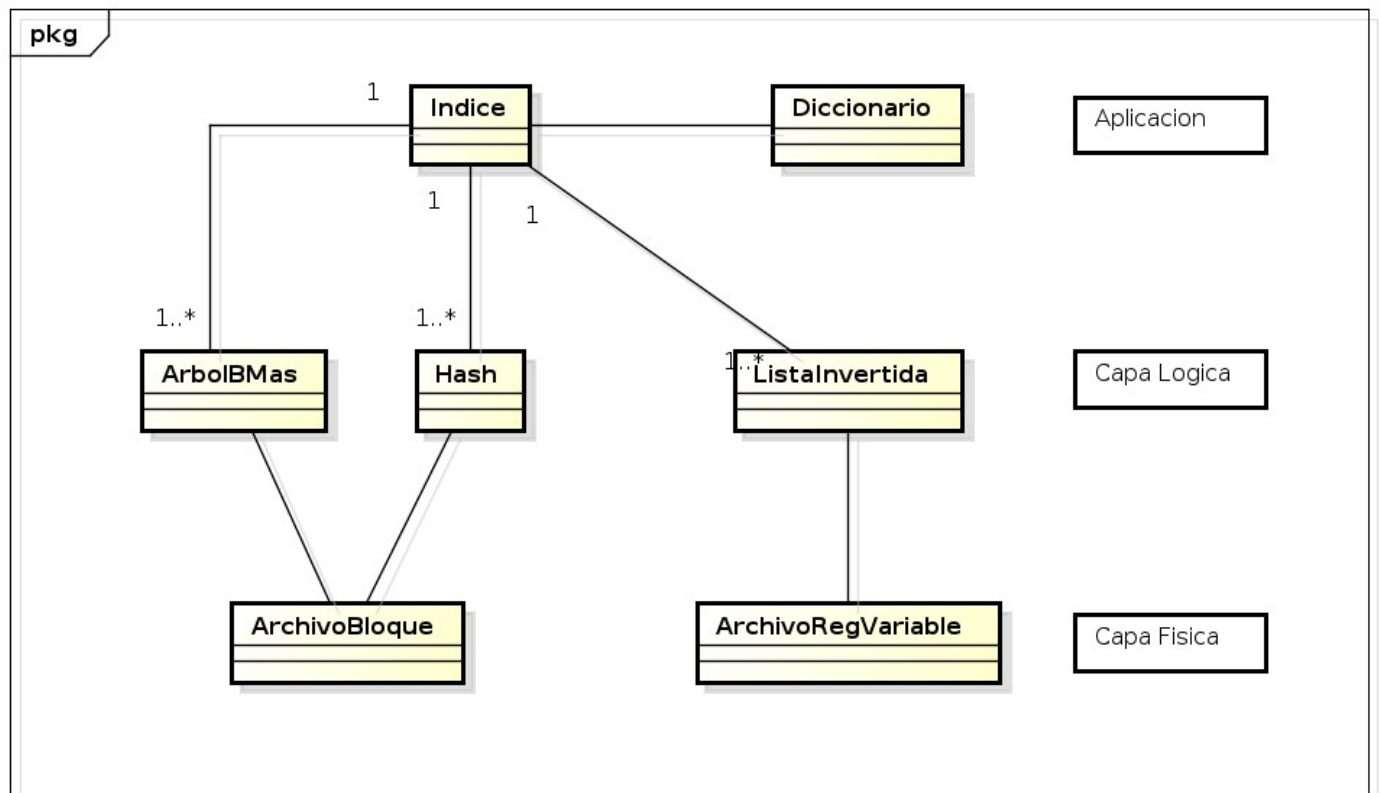
**Observaciones:**

## **Objetivo**

El objetivo del trabajo práctico fue realizar un sistema de compra/venta donde los usuarios sean capaces de publicar servicios, así como también consultar y responder preguntas sobre los mismos. Para ello se utilizaron estructuras complejas de organización de datos (Hash, Árboles) las cuales son muy eficientes al momento de almacenar datos y sobre todo para la búsqueda de los mismos.

## Detalles de implementación

A continuación se muestra un diagrama general que resume como se relacionan las estructuras principales utilizadas en el trabajo y en que capa se encuentran las mismas. Como se puede observar, tenemos la capa física en donde están los archivos en bloque y los archivos de registro variables donde se persiste toda la información que es ingresada. Luego tenemos la capa lógica, la cual esta compuesta por las estructuras mas complejas del sistema, el hash, el árbol y las listas invertidas, las mismas se explican en detalle mas abajo. Por ultimo se encuentra la aplicación, en donde se encuentra el índice, el cual se encarga de manejar a todas las estructuras de la capa lógica. Otras clases que también se encuentran en la aplicación son las requeridas en el trabajo, usuarios, productos, servicios, etc. No fue necesario incluirlas en el diagrama ya que en el trabajo se da uno bien detallado.



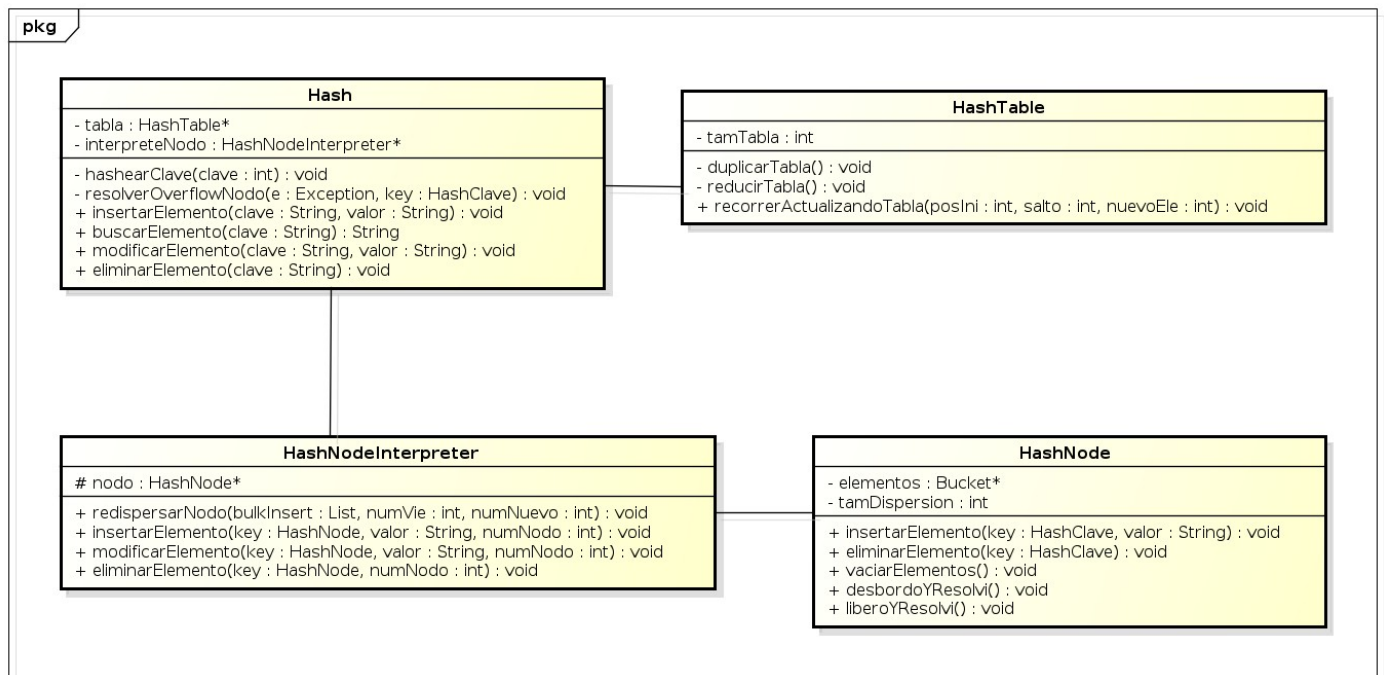
# ESTRUCTURA HASH

El Hash extensible está implementado con dispersión modular: se toma el modulo de la suma de los valores ASCII de los caracteres de la clave de cada registro para obtener el nodo al que corresponde. El hash posee dos estructuras principales: el interprete de nodo y la tabla de hash. El interprete de nodo es el encargado de realizar las operaciones en los nodos del hash: inserción, eliminación, etc. Es el encargado de redispersar el nodo, al cambiar el tamaño de la tabla del hash. Cuando ocurre un overflow, es el mismo hash quien en ultima instancia, se encarga de su resolución.

En cada nodo hay un Bucket, en el que se insertan los registros a guardar.

Los registros y la tabla se almacenan en dos archivos de bloques. (Ej: nodos.dat y tabla.dat).

A continuación se puede apreciar el diagrama de clases de la estructura Hash.



## Estructura Árbol B+

El árbol esta compuesto por dos tipos de Nodos, los Nodos Hoja y los Nodos Internos, ambos utilizan archivos en bloque para guardar la información. Como es sabido, en los arboles b+, toda la información se encuentra en las hojas y los nodos internos solo contienen referencias para poder llegar a ellas, es por eso que se creo la clase RegistroArbol, constituida por una Clave y un valor principal, necesarios para la implementación de los índices secundarios, estos registros solo se almacenan en los Nodos Hoja. La clase clave, esta compuesta por un identificador y es almacenada solo en los Nodos Internos. La clase ArbolBMas es la clase principal, se encarga de manejar los overflow y los underflow, equilibrando o haciendo merge sobre los nodos cuando sea necesario. También se encargar de insertar, modificar o eliminar los registros en los correspondientes nodos, chequeando siempre que todo quede de manera correcta. A continuación se muestra un diagrama de clases del árbol para ver todas las interacciones recién explicadas.

