

## **U.B.A. FACULTAD DE INGENIERÍA**

### **Departamento de Computación**

#### **Organización de Datos 75.06 Informática**

### **TRABAJO PRÁCTICO N°1**

*Gestock y Ventas*

**Curso 2014 - 2do Cuatrimestre**

**Curso: Servetto**

<b>GRUPO KIKA</b>	
<b>APELLIDO, Nombres</b>	<b>N° PADRÓN</b>
<b>Segui, Joaquín</b>	<b>91451</b>
<b>Menniti, Sebastián</b>	<b>93445</b>
<b>Nery, Francisco</b>	<b>94506</b>
<b>Hidalgo, Juan Manuel</b>	<b>93383</b>
<b>Fecha de Entrega : 11/10/2014</b>	
<b>Calificación :</b>	
<b>Firma de Aprobación :</b>	

**Observaciones:**

## **Objetivo**

El objetivo de este trabajo práctico es diseñar un sistema de gestión de stock y ventas para locales de indumentaria, en este caso, corpiños. Se utilizarán herramientas vistas en las clases para implementar organizaciones y herramientas para poder crear Entidades, agregar Instancias, listarlas, entre otras funciones. Utilizamos los conocimientos aprendidos en clase sobre archivos y forma de organizarlos, en concreto estructuras como Arbol B+ y Hash, para guardar los datos necesarios del sistema.

El sistema estará compuesto de 5 módulos:

- 1- Administración de Entidades
- 2- Registro de Ingresos
- 3- Registro de Ventas
- 4- Consultas
- 5- Libro de Quejas en línea

En esta entrega del trabajo se desarrolló lo necesario para el 1er modulo.

## **Detalles de implementación**

Se adjunta a continuación el diagrama de clases que muestra cómo se relacionan las clases principales utilizadas y a que capa corresponde cada una.

Como se puede observar, se encuentra la capa física en donde están los archivos de registros fijos, de registros variables y de bloques, los cuales se utilizan para guardar los datos de las instancias e entidades. Hay una clase abstracta llamada Archivo, de la cual heredan los tres tipos de archivos mencionados anteriormente.

Luego tenemos la Aplicación, que contiene a las clases Intefaz, AdministradorEntidades, entre otras, las cuales se encargan de proveer una interfaz para que el usuario realice las operaciones que desee.

Para la próxima entrega, se llevara a cabo una capa lógica, que contendrá las estructuras vistas en clase de Árbol B+ y de Hash, para poder realizar búsquedas y armar índices en los archivos que contentan instancias y entidades.

Para cada archivo que contiene las instancias, se encuentra otro archivo con información del espacio libre y/o ocupado.

## **Implementación de los distintos tipos de archivos:**

### **RegistroFijo:**

Al principio del archivo hay 2 números q ocupan 4 bytes cada uno, cantidad de registros y el tamaño de cada registro. Todos los registros tienen el mismo tamaño y este tamaño es igual a la máxima cantidad de bytes que puede ocupar la entidad.

El mapa de bits tiene un 1 si el bloque está ocupado y 0 si libre.

Ej: Entidad tiene como atributos 1 entero y 1 string de 20 bytes, los registros pesan 24 bytes cada uno. El primer registro esta desde el byte 8 al 32. Luego de 32 a 56,etc.

### **RegistroVariable:**

Al principio del archivo hay 1 número q ocupa 4 bytes, la cantidad de instancias. Cada string tiene un caracter separador al final para saber cuando termina.

El mapa de bits tiene 2 números por instancia: donde empieza y donde termina la instancia. Y esta ordenado, es decir, el fin de una instancia es menor al comienzo de la próxima. El mapa de bits tiene 8 bytes por instancia del archivo.

En Ej: Tiene un string de 9 bytes (máx. 20). Ocupa  $4 + 9 + 1 = 14$  bytes. Ocupa desde byte 4 a 18. Y en mapa de bits se guarda el 4 y el 18.

### **Bloques:**

Hay 2 números al principio: la cantidad de bloques y el tamaño de cada bloque. Los string también tienen carácter separador. Además, antes de cada instancia se guarda el tamaño de la instancia. El tamaño de la instancia incluye su propio peso (4 bytes).

El mapa de bits guarda el espacio libre y la cantidad de instancias de cada Bloque.

Como en un bloque pueden haber varias instancias, se declaró una constante que indica la cantidad mínima de instancias que puede haber por bloque.

En Ej: String de 9 bytes. TamañoInstancia:  $4$  (por tamaño) +  $4 + 9 + 1 = 18$  bytes. Desde byte 8 a 26 (suponiendo que es el primer registro).

El maximo que puede pesar una instancia seria:  $4 + 4 + 20 + 1 = 29$  bytes. El tamaño de bloque es, entonces, de:  $29 * \text{constante (constante} = 4) : 116$  bytes.

## Diagrama de Clases:

