

### **Trabajo Práctico N° 3: Persistencia**

1. Escribir la clase ArchivoDeEnteros que:

- almacene una secuencia de números enteros en un archivo de acceso aleatorio
- recupere y muestre los datos almacenados
- busque un determinado entero
- calcule el promedio de los valores almacenados
- ordene de menor a mayor el archivo
- agregue más elementos en el archivo.
- Almacene los pares en el archivo de acceso aleatorio "pares.dat" y los impares en "impares.dat"

NOTA: Una clase diferente contiene el método main con el menú respectivo para ejecutar las operaciones requeridas del objeto ArchivoDeEnteros.

```
public static void main(String[] args){
    // declaracion de variables del menu, bucle, etc.
    int opcion;
    ArchivoDeEnteros archent;
    -
    -
    archent = new ArchivoDeEnteros("enteros.dat");
    -
    do{
        System.out.println("1: Agregar enteros");
        -
        -
        opcion = Console.readInt("OPCION: ");
        switch (opcion)
        {
            case 1: {
                comienzo bucle
                leer entero por teclado o generar random
                archent.guardar(entero)
                -
                -
                fin bucle
                break
            }
            -
            -
        }
    }
}
```

2. Escribir la clase ArchivoPersonas que:

- Genere el archivo de acceso aleatorio "datos.dat" que almacene los datos de los objetos Persona (nombre, dni y altura).
- Recupere los datos almacenados en "datos.dat" y los muestre por pantalla.

NOTA: Utilizar el mismo esquema que en el ejercicio 1 para la implementación, añadiendo la clase Persona.

3. Al ejercicio anterior agregar las siguientes funciones:

- Buscar una determinada persona por dni
- Calcular la altura promedio de las personas almacenadas en el archivo
- Generar el archivo "auxiliar.dat" con aquellas personas cuya altura supera 1.60 mts.

4. La clase PersonManager gestiona objetos Persona que se almacenan en un Vector (clase java.util.Vector) y cuenta con los métodos:

- addPerson(Person p) // agrega p al vector
- removePerson (String n) // elimina del vector la persona con nombre n
- searchPerson (String n) // busca en el vector la persona con nombre n

Además el gestor proporciona funcionalidades para hacer persistentes las personas del Vector.

**Programación Orientada a Objetos**  
**Analista de Sistemas – Licenciatura en Sistemas**

```
- savePersons(String file) // almacena en file cada una de las personas del vector,  
si este no esta vacío.  
- retrievePersons(String file) // los objetos Persona almacenados en file se  
agregan al Vector
```

Implemente la clase PersonManager y realice varias pruebas. (Usar serialización)

5. Al ejercicio anterior se le agregan los métodos para hacer persistente al contenedor (Vector) con todas las personas que contenga. Complete la implementación propuesta. (Usar serialización)

```
void saveContainer(String file)  
    { // almacena el contenedor en file }  
void retrieveContainer(String file)  
    { // obtiene de file el contenedor del gestor }  
}
```