

Guía de Trabajos Prácticos N° I – POO 1

1. Dado el ejemplo en Teoría de la clase **Persona** codifique en lenguaje JAVA.
Agregue una nueva clase “Principal” que tenga ejecución y demuestre el concepto de identidad de un objeto.

* Una clase JAVA tiene ejecución si posee un método main:

public static void main (String[] arg)

* Para poder ejecutar una clase java podemos hacerlo desde la consola (debe tener instalado previamente el runtime de java JRE):

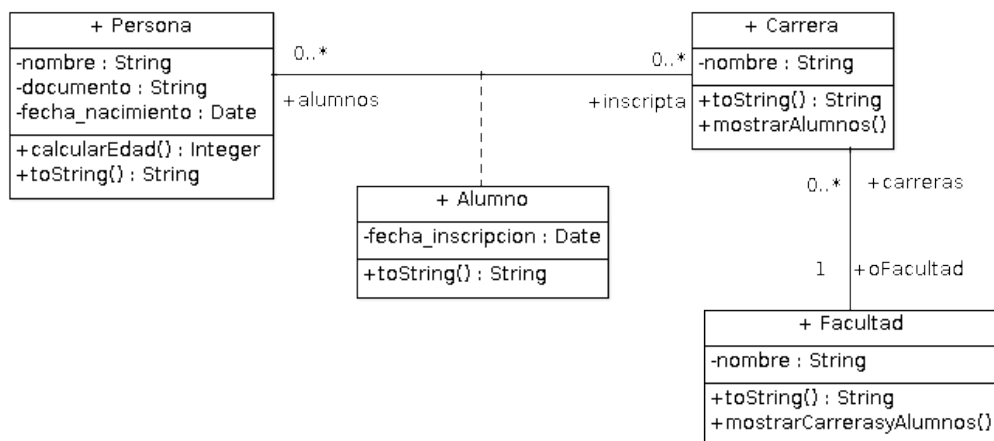
java Principal

2. A la clase Persona agréguele un atributo del tipo **java.util.Calendar** para representar la fecha de nacimiento. Modifique el constructor de la clase teniendo en cuenta el nuevo atributo y agregue un método privado que devuelva la edad de la persona y otro método mostrar() que muestre en la salida estándar o consola: apellido, nombre: edad → “Juan, Perez: 22 años.” + Día del cumpleaños en el año en curso.

* Salida en consola **System.out.println(String)**

Haga una clase ejecutable, cree una persona con sus datos personales (instancia) y ejecute el método mostrar.

3. Dado el siguiente UML, codifique las clases en java.



Use una clase que sea ejecutable y cargue los siguientes datos en la estructura de objetos:

- Facultad: FICH
- Carreras: Ingeniería en Informática – Ingeniería en Recursos Hídricos.
- Alumnos:
Alumno1, DNI 11.111.111, nacimiento 11/11/1990, inscripción 10/12/2008 en Ing. en Informática.
Alumno2, DNI 22.222.222, nacimiento 12/12/1990, inscripción 11/12/2008 en Ing. en Informática

Una vez cargada la información mandarle el mensaje **mostrarCarrerasyAlumnos** al objeto facultad instanciado. Este método tiene que mostrar por consola la siguiente salida:

```
Facultad: FICH
Carrera: Ingeniería en Informática
Alumnos:  Alumno1 – 10/12/2008
           Alumno2 – 11/12/2008
Carrera: Ingeniería en Recursos Hídricos.
```

4. Agregue a la clase “Persona”, creada en el ejercicio 2, un nuevo atributo que sea la clave personal o password, debe contener un String que se debe generar con la clave HASHEADA con SHA256, por ejemplo si la clave es “password” el atributo va a contener el valor
“5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8”

Agregue una nueva funcionalidad a la clase “Persona”. Debe validar la clave con un método que reciba como parámetro la clave a evaluar y la compara con el valor de la instancia. Este método debe devolver “Verdadero” si coincide y “Falso” en caso contrario.

Ayudin:

```
String claveSHA = StringMD.getStringMessageDigest(clave_para_encriptar, StringMD.SHA256);
```

```
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
```

```
public class StringMD {

    //algoritmos
    public static String MD2 = "MD2";
    public static String MD5 = "MD5";
    public static String SHA1 = "SHA-1";
    public static String SHA256 = "SHA-256";
    public static String SHA384 = "SHA-384";
    public static String SHA512 = "SHA-512";
```

```
/**
 * Convierte un arreglo de bytes a String usando valores hexadecimales
 * @param digest arreglo de bytes a convertir
 * @return String creado a partir de <code>digest</code>
 */
private static String toHexadecimal(byte[] digest){
    String hash = "";
    for(byte aux : digest) {
        int b = aux & 0xff;
        if (Integer.toHexString(b).length() == 1) hash += "0";
        hash += Integer.toHexString(b);
    }
    return hash;
}

/**
 * Encripta un mensaje de texto mediante algoritmo de resumen de mensaje.
 * @param message texto a encriptar
 * @param algorithm algoritmo de encriptación, puede ser: MD2,MD5,SHA-1,SHA-256,SHA-384,SHA-512
 * @return mensaje encriptado
 */
public static String getStringMessageDigest(String message, String algorithm){
    byte[] digest = null;
    byte[] buffer = message.getBytes();
    try {
        MessageDigest messageDigest = MessageDigest.getInstance(algorithm);
        messageDigest.reset();
        messageDigest.update(buffer);
        digest = messageDigest.digest();
    } catch (NoSuchAlgorithmException ex) {
        System.out.println("Error creando Digest");
    }
    return toHexadecimal(digest);
}
}
```