

## 7.G. Anexos de ejercicios resueltos.

### 3. Anexo III.- Métodos para los atributos de las clases Alumno y Profesor.

#### ENUNCIADO

Dadas las clases **Persona**, **Alumno** y **Profesor** que has utilizado anteriormente, implementa métodos **get** y **set** en la clase **Persona** para trabajar con sus tres atributos y en las clases **Alumno** y **Profesor** para trabajar con sus cinco atributos (tres heredados más dos específicos), teniendo en cuenta que los métodos que ya hayas definido para **Persona** van a ser heredados en **Alumno** y en **Profesor**.

#### POSIBLE SOLUCIÓN

##### 1. Clase **Persona**.

```
public class Persona {  
  
    protected String nombre;  
  
    protected String apellidos;  
  
    protected GregorianCalendar fechaNacim;  
  
  
    // Método getNombre  
  
    public String getNombre (){  
  
        return nombre;  
  
    }  
  
  
    // Método getApellidos  
  
    public String getApellidos (){  
  
        return apellidos;  
  
    }  
  
  
    // Método getFechaNacim  
  
    public GregorianCalendar getFechaNacim (){  
  
        return this.fechaNacim;  
  
    }  
  
  
    // Método setNombre  
  
    public void setNombre (String nombre){  
  
        this.nombre= nombre;  
  
    }  
  
  
    // Método setApellidos  
  
    public void setApellidos (String apellidos){  
  
        this.apellidos= apellidos;  
  
    }  
  
}
```

```
// Método setFechaNacim

public void setFechaNacim (GregorianCalendar fechaNacim){

    this.fechaNacim= fechaNacim;

}
```

## 2. Clase Alumno.

Al heredar de la clase **Persona** tan solo es necesario escribir métodos para los nuevos atributos (**métodos especializados** de acceso a los **atributos especializados**), pues los **métodos genéricos** (de acceso a los **atributos genéricos**) ya forman parte de la clase al haberlos heredado.

```
public class Alumno extends Persona {

    protected String grupo;

    protected double notaMedia;

    // Método getGrupo

    public String getGrupo (){

        return grupo;

    }

    // Método getNotaMedia

    public double getNotaMedia (){

        return notaMedia;

    }

    // Método setGrupo

    public void setGrupo (String grupo){

        this.grupo= grupo;

    }

    // Método setNotaMedia

    public void setNotaMedia (double notaMedia){

        this.notaMedia= notaMedia;

    }

}
```

Aquí tienes una demostración práctica de cómo la herencia permite una reutilización eficiente del código, evitando tener que repetir atributos y métodos. Sólo has tenido que escribir cuatro métodos en lugar de diez.

## 3. Clase Profesor.

Seguimos exactamente el mismo procedimiento que con la clase **Alumno**.

```
public class Profesor extends Profesor {
```

```
    String especialidad;
```

```
    double salario;
```

```
    // Método getEspecialidad
```

```
    public String getEspecialidad (){
```

```
        return especialidad;
```

```
    }
```

```
    // Método getSalario
```

```
    public double getSalario (){
```

```
        return salario;
```

```
    }
```

```
    // Método setSalario
```

```
    public void setSalario (double salario){
```

```
        this.salario= salario;
```

```
    }
```

```
    // Método setEspecialidad
```

```
    public void setEspecialidad (String especialidad){
```

```
        this.especialidad= especialidad;
```

```
    }
```

```
}
```