



Nombre y Apellidos:

**Instrucciones.** La tarea constará de diferentes puntos que crearan un modelo de clases y un programa que desarrollaremos en la unidad siguiente.

La tarea se entregará en el aula virtual en un ZIP **tareaTema19NombreApellidos.Zip** conteniendo los ejercicios contestados. Crear un proyecto en eclipse **tareaTema10NombreApellidos**. Cada ejercicio en Java se guardará en una clase diferente como **Problema1NombreApellidos.java** si es la Problema 1, etc.

**RAE 7.** Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.

#### Criterios de evaluación:

- a) Identificar los conceptos de herencia, superclase y subclase. Indicadores 76
- d) Crear clases heredadas que sobrescriban la implementación de métodos de la superclase. Indicadores 71
- e) Diseñar y aplicar jerarquías de clases. Indicadores 72
- g) Realizar programas que implementen y utilicen jerarquías de clases. Indicadores 74 y 77

## Detalles de la tarea de esta unidad.

### Enunciado.

En esta tarea vamos a aplicar lo aprendido en el tema 10, desarrollando dos patrones de diseño, Decorator y Adapter sobre nuevas clases y patrones.

### Parte I. Añadimos cuentas a nuestros empleados. (6 puntos)

Indicadores 71,72,74, 76, 77

1. *Se trata de añadir a nuestro programa de la Tarea 6 parte 2, cuentas bancarias, creando una clase nueva cuentas bancarias dentro del paquete `datos.cuentasbancarias`, para poder realizar el pago a los trabajadores. Debemos añadir una nueva clase `CuentaBancaria.java` y un nuevo atributo a trabajador cuenta bancaria. (1 punto)*



Nombre y Apellidos:

### **Atributos**

- **Titular de la cuenta (id del trabajador).**
- **Número de cuenta (IBAN). (String)**
- **Comision de mantenimiento**
- **Tipo de interés, sólo para las cuentas de ahorro.**
- **Función (optativo)**

**El constructor de Cuenta Bancaria con comisión y tipo de interés a 0. No hacemos setters para estos dos atributos, ni para función en caso de que lo añadais.**

```
public class CuentaBancaria {  
  
    private int titular;  
    private String Iban;  
    private Double comisionMantemimimiento=0;  
    private Double tipoInteres=0;  
  
    public CuentaBancaria() {  
  
    }  
  
    public CuentaBancaria(int titular, String iban) {  
        super();  
        this.titular = titular;  
        Iban = iban;  
  
    }  
}
```

### **Métodos para el Decorator relativos a**

- **Comision de mantenimiento**



Nombre y Apellidos:

- *Tipo de interés, sólo para las cuentas de ahorro.*
- *Función*

**Clonar el resto de atributos de CuentaBancaria**

**Añadir este método toString para probar las cuentas. Podéis añadir aquí función si queréis (Optativo)**

```
@Override  
  
    public String toString() {  
  
        return "CuentaBancaria [titular=" + titular + ", Iban=" + Iban +  
        ", comisionMantenimiento=" + getComisionMantenimiento() + ", tipoInteres=" +  
        getTipoInteres() + "];"  
  
    }
```

**2. Las cuentas bancarias pueden ser de tres tipos: cuentas de ahorro o bien cuentas corrientes o cuentas de préstamo, de manera que una cuenta puede ser de los tres tipos (2 puntos)**

- *Las cuentas de ahorro tienen como función el ahorro al 2% de interes y un coste de mantenimiento de 1 euro anuales. Función AHORRO.*
- *Las cuentas corrientes tienen como función realizar pagos y un coste de mantenimiento de 50 euros anuales. Función PAGOS.*
- *Las cuentas de préstamo añaden un sobre coste de 40 euros anuales por el seguro de préstamo. Función PRESTAMO.*

**Aplicar un patron Decorator DecoratorCuenta cuenta que permita combinar estas cuentas, tener por ejemplo una cuenta de ahorro que además es de préstamo o una cuenta corriente que además es de prestamo. Es vuestra elección si el patrón es tradicional o funcional.**

**3. A la clase trabajador le añadimos un atributo de composición de tipo CuentaBancaria. Debemos modificar el Builder para añadir también un numero de cuenta. Las cuentas deben ser españolas. (1 punto)**



Nombre y Apellidos:

*Generaremos los números de cuenta de manera aleatoria, introduciendo un método nuevo en `GeneraciónDeNumerosAleatorios.java`.*

```
public static String getIBAN()
```

*No se añadirán al usuario en el Builder. Se añadirán posteriormente como cuenta bancaria, con un método `setCuentaBancaria` en el trabajador.*

4. *Introducimos una nueva clase `GenerarCuentas.java`. Nos generará las cuentas de manera aleatoria, siendo al menos de uno de los tres tipos indicados, con un método estático `generaCuenta(int idTrabajador)`. (1,5 punto)*

5. *Añadimos al Builder la cuenta bancaria cuando creamos al Trabajador. (0,5 puntos)*

*El IBAN funciona como sigue*

*Se forma con el identificador nacional (es el C.C.C. en España) al que se le antepone el código de país (Código ISO de 2 letras) y una clave numérica 2 cifras para control. Para escribirlo es preciso hacerlo en grupos de 4 números dejando espacio cada cuatro y precederlo con el nombre de IBAN. Este es un ejemplo de la estructura de un código numérico IBAN.*

#### ESTRUCTURA Y FORMATO DEL IBAN EN ESPAÑA

C.C.C.: 0019 0020 96 1234567890

IBAN : IBAN ES66 0019 0020 9612 3456 7890

Etiqueta

Identificador nacional (C.C.C)

Clave País y  
Código de Control

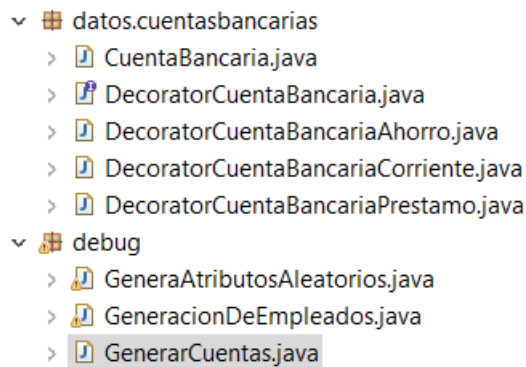


Nombre y Apellidos:

***ES21 es el código para las cuentas españolas.***

- ***Añadimos un nuevo método a trabajador llamado pagoMensual. Este método contendrá un printf indicando que el sueldo ha sido enviado. Comprobaremos que el día es 28 y si lo es se envía el pago.***

***Modelo para esta parte que podéis usar:***



***Podéis usar esta clase para probar la generación de cuentas***

```
import java.util.stream.IntStream;

public class ProbarCuenta {

    public static void main(String[] args) {

        IntStream
            .range(0,100)
            .mapToObj(i-> GenerarCuentas.generaCuenta(i)).forEach((c) ->
                System.out.println(c + " FUNCION:" + c.getFuncion()));
    }
}
```



Nombre y Apellidos:

}

}

CuentaBancaria [titular=87, Iban=IBANES212839670921931574,  
comisionMantemimiento=51.0, tipoInteres=0.02] FUNCION: AHORRO CORRIENTE

CuentaBancaria [titular=88, Iban=IBANES218746241010509992,  
comisionMantemimiento=41.0, tipoInteres=0.02] FUNCION: AHORRO PAGO

CuentaBancaria [titular=89, Iban=IBANES211570244984851698,  
comisionMantemimiento=1.0, tipoInteres=0.02] FUNCION: AHORRO

CuentaBancaria [titular=90, Iban=IBANES215296958751153312,  
comisionMantemimiento=91.0, tipoInteres=0.02] FUNCION: AHORRO CORRIENTE PAGO

CuentaBancaria [titular=91, Iban=IBANES215696636064806966,  
comisionMantemimiento=91.0, tipoInteres=0.02] FUNCION: AHORRO CORRIENTE PAGO

CuentaBancaria [titular=92, Iban=IBANES212058566113381564,  
comisionMantemimiento=1.0, tipoInteres=0.02] FUNCION: AHORRO

CuentaBancaria [titular=93, Iban=IBANES217454320119013964,  
comisionMantemimiento=51.0, tipoInteres=0.02] FUNCION: AHORRO CORRIENTE

CuentaBancaria [titular=94, Iban=IBANES217816369636034736,  
comisionMantemimiento=91.0, tipoInteres=0.02] FUNCION: AHORRO CORRIENTE PAGO

## Parte II. Nueva clase EstudianteEnPracticas (4 puntos)

Indicadores 71,72,74, 76, 77

- 1. Introducimos la figura del EstudianteEnPracticas. Este estudiante trabajará para la empresa, pero su comportamiento es diferente. No tiene sueldo, tiene categoría ESTUDIANTE. Su comportamiento es diferente al resto de empleados de la empresa. Generaremos un estudiante con todos sus campos rellenos a partir de un método estático of, el constructor será privado.**

**(1 punto)**

```
public class EstudianteEnPracticas {
```

```
    private String nombre;  
    private int edad;  
    private Fecha fecha_alta;  
    private String CentroDeEstudios;
```



Nombre y Apellidos:

```
private String direccion="";
```

*Necesitamos getters y setters para esta clase, el método toString.*

2. *Construiremos un EstudianteAdapter, que nos permita que el Estudiante se adapte a nuestro modelo de trabajo con trabajadores. Su sueldo será 0, su Categoría ESTUDIANTE, nueva categoría, y su Antigüedad NOVATO, siempre, y su IBAN será todo ceros, salvo la etiqueta inicial. Aquí deberéis cambiar los modificadores para los constructores sin que se pueda hacer new de EstudianteEnPracticas, EstudianteAdapter y Trabajador en el programa principal, pero si en nuestro paquete modelo en caso de necesidad.*

*(2 punto)*

```
public class EstudianteAdapter extends Trabajador {  
  
    EstudianteAdapter(TrabajadorBuilder trabajadorBuilder,  
        EstudianteEnPracticas estudiante)
```

*Aseguraos que trabajador tiene todos sus setters y getters pues extendemos de el en EstudianteAdapter.*

3. *Debe ser una opción mas a generar por el Builder pero en este caso partir del EstudianteEnPracticas lleno y los datos anteriormente proporcionados. Para ello añadimos un nuevo método buildAdapter al Builder y estudianteEnPracticas.(1 punto)*

*Construiremos el EstudianteAdapter con el Builder de la siguiente manera*

```
new TrabajadorBuilder()  
    .trabajador_id(i)  
    .sueldoFunction((tr)->0.0)
```



Nombre y Apellidos:

```
.categoria(cat)

.estudianteEnPracticas( generaEstudiante())

.build();

}
```

Añadir un nuevo método a generacionDeEmpleados llamado generaEstudiante() para hacerlo más cómodo.

```
public static final String centros[] = {"IES Brianda de Mendoza", "IES Buero
Vallejo", "IES Ribera del Tajo", "IES Castilla",
"IES Jose Luis Sampedro", "IES Leonardo Da Vinci", "IES Virrey
Morcillo", "IES Valdehierro", "IES Maestre de Calatrava", "CIFP Virgen de
Gracia", "IES Maestre de Calatrava", "IES Azarquiel", "IES Pedro
Mercedes"};
```

El nuevo Builder debe ser tener la siguiente forma:

```
public Trabajador build() {

    if (this.categoria==CategoriaEnum.CATEGORIA_ESTUDIANTE)
        return new EstudianteAdapter(this,estudiante);

    else

        return new Trabajador(this);








}
```





Nombre y Apellidos:

El nuevo modelo de datos.

- ▼  datos
  - >  Antiguedad.java
  - >  EstudianteAdapter.java
  - >  EstudianteEnPracticas.java
  - >  Fecha.java
  - >  Trabajador.java
  - >  TrabajadorBuilder.java

Ejemplo de ejecución

Trabajador [trabajador\_id=1, nombre=DOLORES Moreno Muñoz, edad=52, categoria=CATEGORIA\_TECNICO, antiguedad=ANTIGUEDAD\_EXPERTO, fecha\_alta=1 del mes de Noviembre de 2013, sueldoHora=0.0, horas=0.0, sueldoBase=2954.0, plusAntiguedad=0.0, direccion=Alfonso López de Haro, porcentajeCategoria=0.0, cuenta=CuentaBancaria [titular=1, Iban=IBANES217265236713672559, comisionMantemimiento=41.0, tipoInteres=0.02]]

Trabajador [trabajador\_id=2, nombre=JOSE ANTONIO Romero Álvarez, edad=42, categoria=CATEGORIA\_EMPLEADO, antiguedad=ANTIGUEDAD\_NOVATO, fecha\_alta=9 del mes de Octubre de 2001, sueldoHora=0.0, horas=0.0, sueldoBase=2035.0, plusAntiguedad=0.0, direccion=Calle de Alvarfáñez de Minaya, porcentajeCategoria=0.0, cuenta=CuentaBancaria [titular=2, Iban=IBANES212790512612037191, comisionMantemimiento=90.0, tipoInteres=0.0]]

Trabajador [trabajador\_id=3, nombre=JAVIER Gutiérrez Romero, edad=35, categoria=CATEGORIA\_ESTUDIANTE, antiguedad=ANTIGUEDAD\_NOVATO, fecha\_alta=24 del mes de Junio de 2019, sueldoHora=0.0, horas=0.0, sueldoBase=0.0, plusAntiguedad=0.0, direccion=Calle Hermanos Galiano, porcentajeCategoria=0.0, cuenta=null]

Trabajador [trabajador\_id=4, nombre=MANUEL Domínguez Ruiz, edad=39, categoria=CATEGORIA\_ENCARGADO, antiguedad=ANTIGUEDAD\_NOVATO, fecha\_alta=8 del mes de Noviembre de 2010, sueldoHora=0.0, horas=0.0, sueldoBase=2779.0, plusAntiguedad=0.0, direccion=Calle de Arrabal del Agua, porcentajeCategoria=0.0, cuenta=CuentaBancaria [titular=4, Iban=IBANES213692661755979053, comisionMantemimiento=90.0, tipoInteres=0.0]]

Trabajador [trabajador\_id=5, nombre=LUCIA Navarro Vázquez, edad=64, categoria=CATEGORIA\_DIRECTIVO, antiguedad=ANTIGUEDAD\_NOVATO, fecha\_alta=27 del mes de Abril de 2018, sueldoHora=0.0, horas=0.0, sueldoBase=1841.0, plusAntiguedad=0.0, direccion=Calle Minaya, porcentajeCategoria=0.0, cuenta=CuentaBancaria [titular=5, Iban=IBANES212780916828418884, comisionMantemimiento=51.0, tipoInteres=0.02]]

Trabajador [trabajador\_id=6, nombre=FRANCISCO González Martin, edad=52, categoria=CATEGORIA\_DIRECTIVO, antiguedad=ANTIGUEDAD\_NOVATO, fecha\_alta=19 del mes de Enero de 2004, sueldoHora=0.0, horas=0.0, sueldoBase=1670.0,



Nombre y Apellidos:

plusAntiguedad=0.0, direccion=Calle de Arcipreste de Hita,  
porcentajeCategoria=0.0, cuenta=CuentaBancaria [titular=6,  
Iban=IBANES212362606284516603, comisionMantemimiento=50.0, tipoInteres=0.0]]

Trabajador [trabajador\_id=7, nombre=MANUEL Pérez Alonso, edad=49,  
categoria=CATEGORIA\_FREE\_LANCE, antiguedad=ANTIGUEDAD\_NOVATO, fecha\_alta=2  
del mes de Enero de 2012, sueldoHora=0.0, horas=0.0, sueldoBase=2977.0,  
plusAntiguedad=0.0, direccion=Alfonso López de Haro, porcentajeCategoria=0.0,  
cuenta=CuentaBancaria [titular=7, Iban=IBANES214252102232553038,  
comisionMantemimiento=40.0, tipoInteres=0.0]]

Trabajador [trabajador\_id=8, nombre=DANIEL González Martin, edad=62,  
categoria=CATEGORIA\_ESTUDIANTE, antiguedad=ANTIGUEDAD\_NOVATO, fecha\_alta=18  
del mes de Febrero de 2008, sueldoHora=0.0, horas=0.0, sueldoBase=0.0,  
plusAntiguedad=0.0, direccion=Plaza de Beladiez, porcentajeCategoria=0.0,  
cuenta=null]

Trabajador [trabajador\_id=9, nombre=JUAN López González, edad=66,  
categoria=CATEGORIA\_ESTUDIANTE, antiguedad=ANTIGUEDAD\_NOVATO, fecha\_alta=19  
del mes de Enero de 2005, sueldoHora=0.0, horas=0.0, sueldoBase=0.0,  
plusAntiguedad=0.0, direccion=Calle de Alvarfáñez de Minaya,  
porcentajeCategoria=0.0, cuenta=null]

Trabajador [trabajador\_id=10, nombre=MANUEL Jiménez Martínez, edad=49,  
categoria=CATEGORIA\_TECNICO, antiguedad=ANTIGUEDAD\_EXPERTO, fecha\_alta=15 del  
mes de Noviembre de 2016, sueldoHora=0.0, horas=0.0, sueldoBase=2573.0,  
plusAntiguedad=0.0, direccion=Calle de Alvarfáñez de Minaya,  
porcentajeCategoria=0.0, cuenta=CuentaBancaria [titular=10,  
Iban=IBANES219012298582318447, comisionMantemimiento=40.0, tipoInteres=0.0]]

Trabajador [trabajador\_id=11, nombre=JOSE ANTONIO Alonso González, edad=59,  
categoria=CATEGORIA\_ANONIMO, antiguedad=ANTIGUEDAD\_EXPERTO, fecha\_alta=28 del  
mes de Marzo de 2005, sueldoHora=0.0, horas=0.0, sueldoBase=1100.0,  
plusAntiguedad=0.0, direccion=Plaza de Caídos en la Guerra civil,  
porcentajeCategoria=0.0, cuenta=CuentaBancaria [titular=11,  
Iban=IBANES212952817814161548, comisionMantemimiento=50.0, tipoInteres=0.0]]

Trabajador [trabajador\_id=12, nombre=JESUS Hernández López, edad=22,  
categoria=CATEGORIA\_ANONIMO, antiguedad=ANTIGUEDAD\_NOVATO, fecha\_alta=16 del  
mes de Junio de 2002, sueldoHora=0.0, horas=0.0, sueldoBase=2227.0,  
plusAntiguedad=0.0, direccion=Calle Minaya, porcentajeCategoria=0.0,  
cuenta=CuentaBancaria [titular=12, Iban=IBANES216543891950816778,  
comisionMantemimiento=50.0, tipoInteres=0.0]]

Trabajador [trabajador\_id=13, nombre=ELENA Álvarez Álvarez, edad=50,  
categoria=CATEGORIA\_ANONIMO, antiguedad=ANTIGUEDAD\_EXPERTO, fecha\_alta=19 del  
mes de Julio de 2013, sueldoHora=0.0, horas=0.0, sueldoBase=2790.0,  
plusAntiguedad=0.0, direccion=Plaza de Beladiez, porcentajeCategoria=0.0,  
cuenta=CuentaBancaria [titular=13, Iban=IBANES216137733672694721,  
comisionMantemimiento=90.0, tipoInteres=0.0]]