



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.

Se creará un proyecto *sanchez_manas_begona_PROG08_TareaParteI para la Parte I* y otro proyecto dentro del zip *sanchez_manas_begona_PROG08_TareaParteII para la Parte II*

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. Indicador 68,78
- d) Crear clases heredadas que sobrescriban la implementación de métodos de la superclase. Indicador 71
- g) Realizar programas que implementen y utilicen jerarquías de clases. Indicador 76, 78, 79

Detalles de la tarea de esta unidad.

Enunciado.

Tarea Unidad 8

Introducción

En esta tarea **vamos a evaluar todos los conceptos introducidos en el tema 5** a lo largo de los ejercicios. Durante el tema hemos visto los **conceptos básicos de estructuras de control en programación en Java. estructuras de control a nivel de clases**, con **clases abstractas e interfaces abstractas**. Igualmente hemos visto **los principios de diseño y los patrones**. Es una **continuación de la tarea del Tema 3** en el que vimos **orientación a objetos**.

La tarea consta de dos partes:



Nombre y Apellidos:

Parte I. Manejo avanzados de clases e interfaces java. Principios de diseño (5 puntos)

Parte II. Moviendonos hacia modelos más sencillos y con menos clases. Patrones de diseño (5 puntos)

Colocamos estos ejercicios en tres proyectos, **indicados a principio de cada parte.**

Parte I. Manejo avanzado de clases e interfaces java. (5 puntos)

Indicadores 68, 71, 76

Volvemos a cambiar la practica de la tarea 4, para añadir varias cosas:

- Para la **categoría y la antigüedad vamos a añadir dos tipos enumerados.**
- Un **interfaz funcional BiFunction para calcular los sueldos.**
- Dos nuevas categorías FreeLance y Anonimo, pero sólo la primera será una subclase.

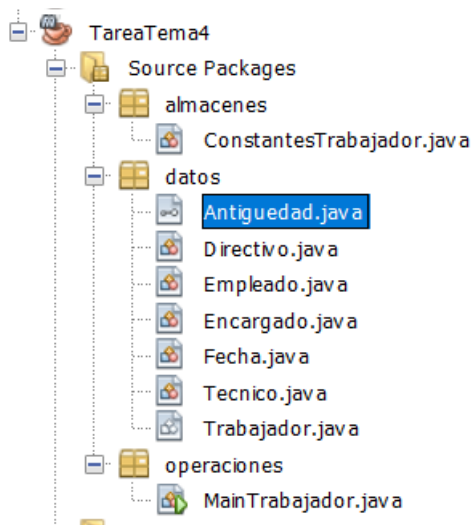
Creareis un proyecto para la parte I llamado:

apellido1_apellido2_nombre_PROG08_TareaParteI

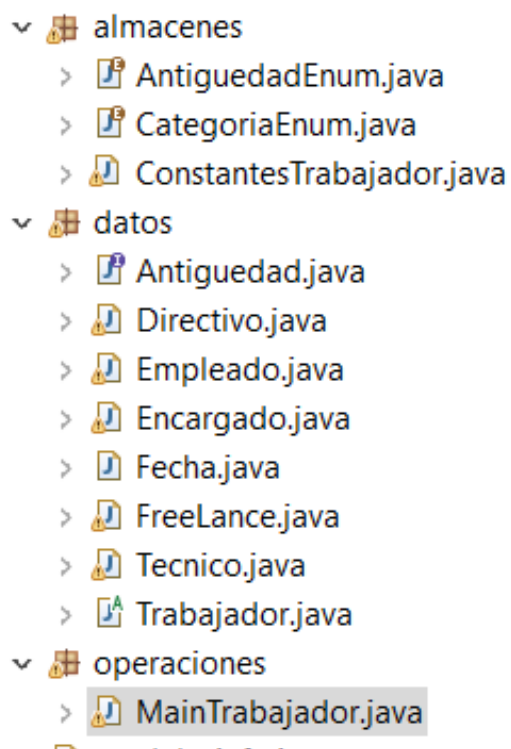
Estructura de la practica 4.2



Nombre y Apellidos:



La estructura de nuestro proyecto se dispondrá como sigue:





Nombre y Apellidos:

Apartado 1 (1 punto)

Creemos un tipo enumerado para Categoría y Antigüedad, llamados **CategoríaEnum**, **AntigüedadEnum**, moviendo la información del del fichero de constantes a los nuevos enumerados

Añadimos dos nuevas Categorías a CategoríaEnum.

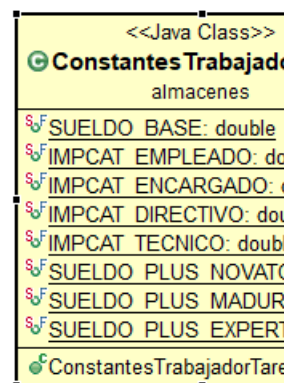
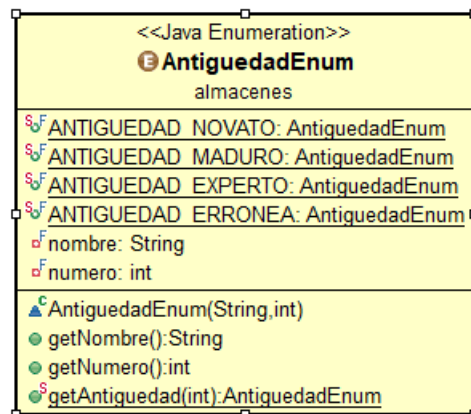
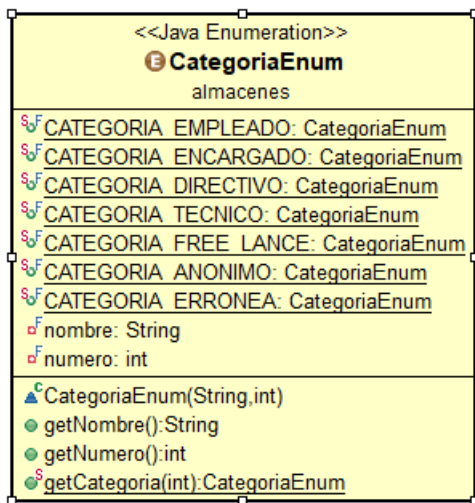
```
CATEGORIA_FREE_LANCE("FREELANCE", 4),
```

```
CATEGORIA_ANONIMO("ANONIMO",5),
```

Podéis comprobar los métodos y valores necesarios para CategoríaEnum y AntigüedadEnum en el modelo.



Nombre y Apellidos:



Apartado 2. (1 punto)



Nombre y Apellidos:

Los métodos **getNombreCategoria** y **getNombreAntiguedad** dejan de ser abstractos y **se implementan en la clase trabajador** usando los tipos enumerados. Ya no son abstractos. Lo veremos en el modelo posterior.

Como abstracto sólo el calculo de sueldo en Trabajador

```
public abstract double sueldoTrabajador();
```

Apartado 3. (1 punto)

El calculo de sueldos lo realizaremos con un interfaz funcional BiFunction. Declararlo, así como propiedad protected en la clase Trabajador. Debéis implementar este interfaz en las subclases que sea necesario igualmente.

```
protected BiFunction<Double,Double,Double> sueldoFunction= A definir por  
vosotros;
```

Para FreeLance el cálculo de sueldo será:

Sueldo = numHoras* sueldoHora.

Para Anonimo el calculo del sueldo será:

Sueldo=Suelto base

Para el resto cada uno con su Porcentaje por categoría y su plus de antigüedad



Nombre y Apellidos:

Sueldo del Trabajador = SueldoBase + %sobre el Sueldobase + importe por antigüedad.

El calculo de sueldos para Freelance cambia. **FreeLance.java** es una nueva clase java que añadimos en el modelo. Calcularemos su sueldo con la siguiente formula. Necesitareis esos nuevos atributos.

*Sueldo del Trabajador = numHoras * sueldoHoras*

Apartado 4. (1 punto)

Aplicando la creación de clases anónimas crearemos un empleado con categoría anónima en la función principal en tiempo de ejecución. Sobreescribiremos el sueldo para que sea fijo y que hemos recogido por pantalla. Lo encontrareis en la versión de MainTrabajador.java que se os proporciona en la práctica.

```
else if (CategoriaEnum.CATEGORIA_ANONIMO.getNumero() == categoria) {
```

RECOGER DATOS NECESARIOS

CREAR AQUÍ LA CLASE COMO ANÓNIMA Y SOBREESCRIBIR LO NECESARIO

```
}
```



Nombre y Apellidos:

Apartado 5. Se os proporciona la clase Main nueva. (1 punto)

Debéis hacerla funcionar modificándola sólo en la creación de la clase anónima para el empleado anónimo. Si no funciona, la segunda parte tiene un cero. **El programa debe funcionar con esta clase MainTrabajador. La puntuación será cero para toda esta parte II de la tarea sino lo hace.**

MainTrabajador.java

```
import almacen.es.AntiguedadEnum;
import almacen.es.CategoriaEnum;
import almacen.es.ConstantesTrabajador;
import datos.*;

import java.util.Scanner;

/**
 *
 * @author carlo
```




Nombre y Apellidos:

*/

```
public class MainTrabajador {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        String nombre;

        int edad;

        int categoria;

        int antiguedad;

        //LECTURA DE DATOS DE EMPLEADO:

        System.out.printf("Introduce el nombre: ");

        nombre=sc.nextLine();

        System.out.printf("Introduce la edad: ");

        edad = sc.nextInt();

        sc.nextLine();

        System.out.printf("Introduce la categoría (0 a 5): ");

        categoria = sc.nextInt();

        sc.nextLine();

        System.out.printf("Introduce la antigüedad (0 a 2): ");

        antigüedad = sc.nextInt();

        sc.nextLine();

        AntigüedadEnum antiEnum = AntigüedadEnum.getAntigüedad(antigüedad);

        //LECTURA DE LA FECHA
```



Nombre y Apellidos:

```
System.out.println("FECHA DE ALTA: ");
System.out.printf("\tIntroduce el día: ");
int dia = sc.nextInt();
sc.nextLine();

System.out.printf("\tIntroduce el mes: ");
int mes = sc.nextInt();
sc.nextLine();
System.out.printf("\tIntroduce el año: ");
int anio = sc.nextInt();
sc.nextLine();

Fecha fec = new Fecha(dia, mes, anio);
Trabajador t;

if(CategoriaEnum.CATEGORIA_EMPLEADO.getNumero()==categoria)
    t =new Empleado(nombre, edad, antiEnum, fec);
else if (CategoriaEnum.CATEGORIA_DIRECTIVO.getNumero() == categoria)
    t =new Directivo(nombre, edad, antiEnum, fec);

else if ( CategoriaEnum.CATEGORIA_ENCARGADO.getNumero() == categoria)
    t =new Encargado(nombre, edad, antiEnum, fec);

else if ( CategoriaEnum.CATEGORIA_TECNICO.getNumero() == categoria)
    t =new Tecnico(nombre, edad, antiEnum, fec);

else if ( CategoriaEnum.CATEGORIA_FREE_LANCE.getNumero() ==
categoria) {
```



Nombre y Apellidos:

```
t =new Freelance(nombre, edad, antiEnum,
fec,numHoras,sueldoHoras);

} else if (CategoriaEnum.CATEGORIA_ANONIMO.getNumero() == categoria)
{

    System.out.printf("\tIntroduce el sueldo neto ");
    double sueldoNeto = sc.nextInt();
    t =new Trabajador

AQUÍ TENDREIS QUE CREAR LA CLASE ANÓNIMA CONFORME A LOS APUNTES

}

else

    t =new Empleado(nombre, edad, antiEnum, fec);

System.out.println("=====");
System.out.println("CONTROL DE DATOS DE ENTRADA: ");
System.out.println("\t¿Es correcto el mes:? " + fec.mesCorrecto());
System.out.println("\tEl nombre del mes es: " + fec.nombreMes());

System.out.println("\tNombre de la categoría: " +
t.nombreCategoria());

System.out.println("\t¿Es correcta la antigüedad:? " +
t.antigüedadCorrecta());

System.out.println("\tNombre de la antigüedad: " +
t.nombreAntigüedad(antigüedad));
```



Nombre y Apellidos:

```
System.out.println("DATOS DEL TRABAJADOR: " + t.toString());
```

```
System.out.println("=====");
```

```
}//main
```

```
}//FIN
```

Ejemplos de ejecuciones.

```
=====
==== CONTROL DE DATOS DE ENTRADA:
    ¿Es correcto el mes:? true
        El nombre del mes es: Mayo
    ¿Es correcta la categoría:? true
        Nombre de la categoría: DIRECTIVO
    ¿Es correcta la antigüedad:?
true      Nombre de la antigüedad:
EXPERTO DATOS DEL TRABAJADOR:
    Alicia, edad: 40
    Categoría: DIRECTIVO
    Antigüedad: EXPERTO
    Fecha de Alta: 10 del mes de Mayo de 1998
    Sueldo: 1571.2
=====
=====
```

EJEMPLO DE EJECUCIÓN CUANDO LA CATEGORÍA NO ES CORRECTA:

```
Introduce el nombre: Alicia
Introduce la edad: 40
Introduce la categoría (0 a 3): 5
Introduce la antigüedad (0 a 2): 2
FECHA DE ALTA:
```



Nombre y Apellidos:

```
Introduce el día: 10
Introduce el mes: 5
Introduce el año: 1998
=====
==== CONTROL DE DATOS DE ENTRADA:
¿Es correcto el mes:? true
El nombre del mes es: Mayo
¿Es correcta la categoría:? false
Nombre de la categoría: CATEGORÍA ERRÓNEA
¿Es correcta la antigüedad:?
true Nombre de la antigüedad:
EXPERTO DATOS DEL TRABAJADOR:
Alicia, edad: 40
Categoría: CATEGORÍA ERRÓNEA
Antigüedad: EXPERTO
Fecha de Alta: 10 del mes de Mayo de 1998
Sueldo: 1207.0
=====
=====
```

EJEMPLO DE EJECUCIÓN CUANDO LA ANTIGÜEDAD NO ES CORRECTA:

```
Introduce el nombre: Alicia
Introduce la edad: 40
Introduce la categoría (0 a 3):
2 Introduce la antigüedad (0 a
2): 4 FECHA DE ALTA:
Introduce el día: 10
Introduce el mes: 5
Introduce el año: 1998
=====
==== CONTROL DE DATOS DE ENTRADA:
¿Es correcto el mes:? true
El nombre del mes es: Mayo
¿Es correcta la categoría:? true
Nombre de la categoría: DIRECTIVO
¿Es correcta la antigüedad:? false
Nombre de la antigüedad:
ANTIGÜEDAD ERRÓNEA DATOS DEL
TRABAJADOR:
Alicia, edad: 40
Categoría: DIRECTIVO
Antigüedad: ANTIGÜEDAD ERRÓNEA
Fecha de Alta: 10 del mes de Mayo de 1998
Sueldo: 971.2
```



Nombre y Apellidos:

=====

=====

EJEMPLO DE EJECUCIÓN CUANDO LA ANTIGÜEDAD Y LA CATEGORÍA NO SON CORRECTAS:

Introduce el nombre: Ana
 Introduce la edad: 40
 Introduce la categoría (0 a 3):
 5
 Introduce la antigüedad (0 a
 2): 4 FECHA DE ALTA:
 Introduce el día: 10
 Introduce el mes: 5
 Introduce el año: 1998

=====

===== CONTROL DE DATOS DE ENTRADA:

¿Es correcto el mes:? true
 El nombre del mes es: Mayo
 ¿Es correcta la categoría:? false
 Nombre de la categoría: CATEGORÍA ERRÓNEA
 ¿Es correcta la antigüedad:? false
 Nombre de la antigüedad:
 ANTIGÜEDAD ERRÓNEA DATOS DEL
 TRABAJADOR:
 Ana, edad: 40
 Categoría: CATEGORÍA ERRÓNEA
 Antigüedad: ANTIGÜEDAD ERRÓNEA
 Fecha de Alta: 10 del mes de Mayo de 1998
 Sueldo: 607.0

=====

===== EJEMPLO DE EJECUCIÓN CUANDO EL MES NO ES CORRECTO:

Introduce el nombre: Alicia
 Introduce la edad: 40
 Introduce la categoría (0 a 3):
 2 Introduce la antigüedad (0 a
 2): 2
 FECHA DE ALTA:
 Introduce el día: 10
 Introduce el mes: 55
 Introduce el año: 1998

=====

===== CONTROL DE DATOS DE ENTRADA:

¿Es correcto el mes:? false



Nombre y Apellidos:

```
El nombre del mes es: ERROR
¿Es correcta la categoría:? true
Nombre de la categoría: DIRECTIVO
¿Es correcta la antigüedad:?
true      Nombre de la antigüedad:
EXPERTO DATOS DEL TRABAJADOR:
    Alicia, edad: 40
    Categoría: DIRECTIVO
    Antigüedad: EXPERTO
    Fecha de Alta: 10 del mes de ERROR de 1998
    Sueldo: 1571.2
=====
=====
```

```
Introduce el nombre: Lulo Da Silva
Introduce la edad: 67
Introduce la categoría (0 a 5): 4
Introduce la antigüedad (0 a 2): 2
FECHA DE ALTA:
Introduce el día: 1
Introduce el mes: 1
Introduce el año: 2020
Introduce numero de horas: 500
Introduce sueldo por hora: 500
=====
CONTROL DE DATOS DE ENTRADA:
¿Es correcto el mes:? true
El nombre del mes es: Enero
Nombre de la categoría: FREELANCE
¿Es correcta la antigüedad:? true
Nombre de la antigüedad: EXPERTO
DATOS DEL TRABAJADOR: Trabajador { nombre=, Lulo Da Silva,
edad=67, Antigüedad=EXPERTO, Fecha de Alta=1 del mes de Enero
de 2020, Sueldo=250000}
=====
```



Nombre y Apellidos:












Parte III. Moviendonos hacia modelos más sencillos y con menos clases. (5 puntos)

Indicadores 68, 71, 76, 78 y 79

Hablamos **del patrón Builder en los apuntes**. Vamos a usarlo en esta parte final de la práctica. La **ventaja del patrón Builder** como ya vimos en los apuntes es que **nos permite crear los objetos a demanda añadiéndole lo que necesitan**. Es lo que vamos a hacer en esta **parte final de la práctica**. Vamos a realizar **un modelo más sencillo y vamos a eliminar las clases hijas de la clase abstracta trabajador**. Identificaremos el **tipo de trabajador por su categoría**, de esta manera obtenemos un modelo más sencillo

apellido1_apellido2_nombre_PROG05_TareaParteIII

La nueva estructura del proyecto

- ▼  almacenes
 - >  AntiguedadEnum.java
 - >  CategoriaEnum.java
 - >  ConstantesTrabajador.java
- ▼  datos
 - >  Antiguedad.java
 - >  Fecha.java
 - >  Trabajador.java
 - >  TrabajadorBuilder.java
- ▼  operaciones
 - >  MainTrabajador.java



Nombre y Apellidos:

Apartado 1. Eliminamos las clases que heredan de Trabajador y la clase Constantes y construimos un Builder

(2 puntos)

Siguiendo el patrón Builder creamos una clase **TrabajadorBuilder**. Hemos eliminado de constantes el sueldo base, el porcentaje de sueldo por categoría y el plus por antigüedad, introducirlo donde sea necesario para construir el Trabajador según su categoría. Ahora dos trabajadores con la misma categoría y antigüedad, podrán tener porcentaje diferente, plus diferente por antigüedad y sueldo base diferente. Se decide todo en tiempo de ejecución en la creación del trabajador en particular.

Apartado 2. Calcular el sueldo para cada tipo de categoría de trabajador usando un interfaz funcional.

(1 punto)

El interfaz funcional se pasará en la construcción del objeto a necesidad, según lo definido en el apartado 1. Se construye el interfaz funcional cuando se construye el objeto Trabajador con el TrabajadorBuilder.

Para calcular el sueldo de Empleado, Encargado, Directivo y Tecnico usaremos al fórmular:

Sueldo = SueldoBasePersonal + SueldoBasePersonal*PorcentajeCategoriaPersonal + plusAntigüedadPersonal;

Para FreeLance el cálculo de sueldo será:

Sueldo = numHoras* sueldoHora.

Para Anonimo el calculo del sueldo será:

Sueldo=Suelo base recogido por pantalla



Nombre y Apellidos:

Debemos introducir todos estos parámetros como atributos de clase, ya no tenemos constantes, sólo los tipos enumerados

Apartado 3. Esta vez modificareis la función MainTrabajador (2 puntos)

Para crear **los tipos de trabajador necesarios** **modificareis la función main** añadiendo a **cada tipo de trabajador sólo lo necesario** para poder realizar sus propios cálculos. Es **necesario que funcione** para puntuar el resto de los apartados de esta parte. **Sino funciona la puntuación será cero para la parte III.**

MainTrabajador.java

```
public class MainTrabajador {  
  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        String nombre;  
        int edad;  
        int categoria;  
        int antigüedad;  
  
        //LECTURA DE DATOS DE EMPLEADO:  
        System.out.printf("Introduce el nombre: ");
```



Nombre y Apellidos:

```
nombre=sc.nextLine();

System.out.printf("Introduce la edad: ");
edad = sc.nextInt();
sc.nextLine();

System.out.printf("Introduce la categoría (0 a 5): ");
categoria = sc.nextInt();
sc.nextLine();
CategoriaEnum catEnum= CategoriaEnum.getCategoria(categoria);


System.out.printf("Introduce la antigüedad (0 a 2): ");
antigüedad = sc.nextInt();
sc.nextLine();

AntigüedadEnum antiEnum = AntigüedadEnum.getAntigüedad(antigüedad);

//LECTURA DE LA FECHA

System.out.println("FECHA DE ALTA: ");
System.out.printf("\tIntroduce el día: ");
int dia = sc.nextInt();
sc.nextLine();

System.out.printf("\tIntroduce el mes: ");
int mes = sc.nextInt();
sc.nextLine();
System.out.printf("\tIntroduce el año: ");
int anio = sc.nextInt();
sc.nextLine();
```



Nombre y Apellidos:

```
Fecha fec = new Fecha(dia, mes, anio);
```

```
TrabajadorBuilder tBuilder = new TrabajadorBuilder();
```

```
tBuilder.nombre(nombre)
        .edad(edad)
        .categoria(catEnum)
        .antiguedad(antiEnum)
        .fecha_alta(fec);
```

```
if ( CategoriaEnum.CATEGORIA_FREE_LANCE.getNumero() == categoria) {
```

RECOGER LOS DATOS NECESARIOS PARA FREELANCE Y CREARLO CON EL
OBJETO BUILDER

```
    } else if (CategoriaEnum.CATEGORIA_ANONIMO.getNumero() == categoria)
{
```

RECOGER LOS DATOS NECESARIOS PARA ANONIMO Y CREARLO CON EL
OBJETO BUILDER

```
}
```

```
else {
```

RECOGER LOS DATOS NECESARIOS PARA EL RESTO DE CATEGORIAS Y CREARLO CON EL
OBJETO BUILDER

```
}
```

```
System.out.println("=====");
```



Nombre y Apellidos:

```
System.out.println("CONTROL DE  DATOS DE ENTRADA: ");

System.out.println("\t¿Es correcto el mes:? " + fec.mesCorrecto());

System.out.println("\tEl nombre del mes es: " + fec.nombreMes());


    System.out.println("\tNombre de la categoría: " +
t.nombreCategoria());

    System.out.println("\t¿Es correcta la antigüedad:? " +
t.antigüedadCorrecta());

    System.out.println("\tNombre de la antigüedad: " +
t.nombreAntigüedad(antigüedad));


    System.out.println("DATOS DEL TRABAJADOR: " + t.toString());


System.out.println("=====");

    }//main

};//FIN
```



Nombre y Apellidos:

CRITERIOS DE PUNTUACIÓN. TOTAL 10 PUNTOS.

- En cada ejercicio se expone la puntuación.

RECURSOS NECESARIOS PARA REALIZAR LA TAREA.

- Ordenador personal.
- Sistema operativo Windows o Linux. JDK y JRE de Java, versión 1.8 o superior , NetBeans IDE 8 o superior. Eclipse IDE.

INDICACIONES DE ENTREGA.

Una vez realizada la tarea comprimirás el proyecto Java (**Tarea8_ApellidosNombre**) **con algún compresor de archivos (zip)** y lo enviarás. El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:

apellido1_apellido2_nombre_PROGxx_Tarea

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños, ni espacios en blanco. Así por ejemplo la alumna **Begoña Sánchez Mañas para la tercera unidad del MP de PROG**, debería nombrar esta tarea como...

sanchez_manas_begona_PROG08_Tarea

Se entregará un proyecto ***sanchez_manas_begona_PROG08_TareaParteI*** y ***otro proyecto dentro del zip sanchez_manas_begona_PROG08_TareaParteII***

Se entregará en un pdf igualmente el código, una captura de ejecución.

sanchez_manas_begona_PROG08_Tarea.pdf



Nombre y Apellidos:

Muy importante: dentro de un zip *sanchez_manas_begona_PROG05_Tarea.zip* poneis el **pdf y el programa con la tarea**. La no entrega en este formato supone suspender la tarea. Recordar que tenéis dos oportunidades para cada una.