

NOTA: Para las prácticas deberéis poner vuestro nombre en el código.

1. Una empresa quiere implementar un programa que lleve el control de las incidencias que se producen en sus ordenadores. Cada incidencia tiene un código: 1, 2, 3, 4, etc. Cuando se crea una nueva incidencia, se le asigna un código de forma automática y se pone el estado como “pendiente”. Al crear una incidencia hay que indicar también el número de puesto (un número entero). Cuando se resuelve una incidencia, hay que proporcionar información sobre cómo se ha resuelto o qué es lo que fallaba, además, el estado pasa a “resuelta”. El siguiente trozo de código que va dentro del main genera la salida que se muestra a continuación (El código
 se añade para salto de línea en depuración):

mainIncidencia.php

```
<?php
/**
 * @author Silvia Vilar
 * Ej1UD8 - mainIncidencia.php
 */
include "Incidencia.php";
$inc1 = new Incidencia(105, "No tiene acceso a internet");
$inc2 = new Incidencia(14, "No arranca");
$inc3 = new Incidencia(5, "La pantalla se ve rosa");
$inc4 = new Incidencia(237, "Hace un ruido extraño");
$inc5 = new Incidencia(111, "Se cuelga al abrir 3 ventanas");
$inc2->resuelve("El equipo no estaba enchufado");
$inc3->resuelve("Cambio del cable VGA");
print $inc1;
print $inc2;
print $inc3;
print $inc4;
print $inc5;
print "Incidencias pendientes: " . Incidencia::getPendientes();
?>
```

Salida de la terminal:

```
Incidencia 1 - Puesto: 105 - No tiene acceso a internet - Pendiente<br>
Incidencia 2 - Puesto: 14 - No arranca - Resuelta - El equipo no estaba enchufado<br>
Incidencia 3 - Puesto: 5 - La pantalla se ve rosa - Resuelta - Cambio del cable VGA<br>
Incidencia 4 - Puesto: 237 - Hace un ruido extraño - Pendiente<br>
Incidencia 5 - Puesto: 111 - Se cuelga al abrir 3 ventanas - Pendiente<br>
Incidencias pendientes: 3
```

2. Implementa la clase Movil como subclase de Terminal (con atributos número y tiempo de conversación y el método llama(\$terminal,\$segundosDeLlamada) donde se pasa el terminal al que se llama y se actualiza el tiempo de conversación para el terminal que llama y el llamado). Cada móvil lleva asociada una tarifa que puede ser “rata”, “mono” o “bisonte”. El coste por minuto es de 6, 12 y 30 céntimos respectivamente. Se tarifican los segundos exactos. Obviamente, cuando un móvil llama a otro, se le cobra al que llama, no al que recibe la llamada. A continuación, se proporciona el contenido del main y el resultado que debe aparecer por pantalla.

Programa principal

```
<?php
/**
 * @author Silvia Vilar
 * Ej2UD8 - mainMoviles.php
 */
include "Movil.php";

$m1 = new Movil("678 11 22 33", "rata");
$m2 = new Movil("644 74 44 69", "mono");
$m3 = new Movil("622 32 89 09", "bisonte");
print $m1 . "<br>\n";
print $m2 . "<br>\n";
$m1->llama($m2, 320);
$m1->llama($m3, 200);
$m2->llama($m3, 550);
print $m1 . "<br>\n";
print $m2 . "<br>\n";
print $m3 . "<br>\n";
?>
```

Salida de la terminal

```
Nº 678 11 22 33 – 0 m y 0s de conversación en total - tarificados 0 m y 0 s por un importe de 0 euros<br>
Nº 644 74 44 69 – 0 m y 0s de conversación en total - tarificados 0 m y 0 s por un importe de 0 euros<br>
Nº 678 11 22 33 – 8 m y 40s de conversación en total - tarificados 8 m y 40 s por un importe de 0.52 euros<br>
Nº 644 74 44 69 – 14 m y 30s de conversación en total - tarificados 9 m y 10 s por un importe de 1.1 euros<br>
Nº 622 32 89 09 – 12 m y 30s de conversación en total - tarificados 0 m y 0 s por un importe de 0 euros<br>
```

3. Crea la clase Vehiculo, así como las clases Bicicleta y Coche como subclases de la primera. Para la clase Vehiculo, crea los atributos de clase vehiculosCreados y kilometrosTotales, así como el atributo de instancia kilometrosRecorridos y los métodos avanzar, verKMRecorridos y verKMTotales. Para la clase Coche tenemos un método llenarDepósito, quemarRueda y redefinimos verKMRecorridos para que muestre que es el Coche. Para la clase Bicicleta tenemos los métodos hazCaballito y ponerCadena y redefinimos verKMRecorridos para que muestre que es la Bicicleta.

Prueba las clases creadas mediante un programa con un menú como el que se muestra a continuación:

Programa Principal:

```
<?php
/**
 * author Silvia Vilar
 * Ej3UD8 - mainVehiculos.php
```

```
*/
include "Bicicleta.php";
include "Coche.php";
$bicicleta = new Bicicleta();
$coche = new Coche();
do{
    echo "MENU PRINCIPAL<br>\n";
    echo "=====<br>\n";
    echo "1. Avanza con la bicicleta<br>\n";
    echo "2. Haz el caballito con la bicicleta<br>\n";
    echo "3. Poner cadena de la bicicleta<br>\n";
    echo "4. Avanza con el coche<br>\n";
    echo "5. Quema rueda con el coche<br>\n";
    echo "6. Llenar depósito del coche<br>\n";
    echo "7. Ver kilometraje de la bicicleta<br>\n";
    echo "8. Ver kilometraje del coche<br>\n";
    echo "9. Ver kilometraje total<br>\n";
    echo "10. Ver número de vehículos<br>\n";
    echo "11. Añade una bicicleta<br>\n";
    echo "12. Añade un coche<br>\n";
    echo "X. Salir<br>\n";
    echo "Elige una opción:<br>\n";
    $opcion= readline();
    switch (strtoupper($opcion)) {
        case '1': //avanza con la bicicleta
            echo "Cuantos km quieres recorrer:<br>\n";
            $km=intval(readline());
            $bicicleta->avanza($km);
            break;
        case '2': //haz el caballito con la bicicleta
            $bicicleta->hacerCaballito();
            break;

        case '3': //poner cadena de la bicicleta
            $bicicleta->ponerCadena();
            break;
        case '4': //avanza con el coche
            echo "Cuantos km quieres recorrer:<br>\n";
            $km=intval(readline());
            $coche->avanza($km);
            break;
        case '5': //quema rueda con el coche
            $coche->quemaRueda();
            break;
        case '6': //llenar depósito del coche
            echo $coche->llenarDeposito();
            break;
        case '7': //ver kilometraje de la bicicleta
```

```

        echo $bicicleta->verKMRecorridos();
        break;
    case '8': //ver kilometraje del coche
        echo $coche->verKMRecorridos();
        break;
    case '9': //ver kilometraje total
        echo Vehiculo::verKMTotales();
        break;
    case '10': //ver número de vehículos
        echo Vehiculo::verVehiculosCreados();
        break;
    case '11': //añade una bicicleta
        $nombrebici = readline("Introduce el nombre de la bicicleta: ");
        $$nombrebici = new Bicicleta();
        break;
    case '12': //añade un coche
        $nombrecoche = readline("Introduce el nombre del coche: ");
        $$nombrecoche = new COche();
        break;
    case 'X':
        echo "Programa finalizado";
        break;

    default:
        echo "Opción no válida";
        break;
}
} while ($opcion!='X');
?>

```

4. Crea una clase llamada Persona con los atributos nombre, edad, DNI, sexo (H hombre, M mujer), peso y altura. Define las siguientes constantes de clase: INFRAPESO (valor -1), PESO_IDEAL (valor 0) y SOBREPESO (valor 1);

Se implementarán varios constructores:

- En el constructor por defecto, todos los atributos excepto el DNI serán valores por defecto según su tipo (0 números, cadena vacía para String, etc.). Sexo tendrá el valor hombre "H" por defecto, se deberá comprobar con la función correspondiente su valor "H" o "M".
- Un "constructor" con el nombre, edad y sexo, el resto por defecto.
- Un "constructor" con todos los atributos como parámetro.

Los métodos que se implementarán son:

- getters y setters necesarios
- comprobarSexo(\$sexo): comprueba que el sexo introducido por el usuario es "H" o "M". Si no es correcto, asignará "H" por defecto. Sólo debe ser visible en la clase.
- strIMC(): llamará a calcularIMC y devolverá un string con el nombre de la persona y si está por debajo de su peso, si está en su peso ideal o si tiene sobrepeso en función de las constantes de la clase.

- `calcularIMC()`: calculará si la persona está en su peso ideal (kg/m^2). Si el resultado es un valor menor que 20, la función devuelve `INFRAPESO`, si el resultado es un número entre 20 y 25 (incluidos), significa que está por debajo de su peso ideal y la función devuelve `PESO_IDEAL` y si el resultado es mayor que 25 la función devuelve `SOBREPESO`.
- `esMayorDeEdad()`: indica si es mayor de edad, devuelve un booleano.
- `MostrarIMC()`: devuelve el String generado por la función `strIMC`
- `__toString()`: devuelve toda la información del objeto en un String.

Se creará un trait para usar en la clase `Persona` con dos funciones:

- `generarDNI()`: función pública que genera un número aleatorio de 8 cifras, y calcula el resto de su división por 23. Luego llamará a una función `generaLetraDNI($idLetra)` del mismo trait para obtener la letra. Debe devolver el número DNI obtenido junto con su letra correspondiente. Este método sera invocado en el constructor de `Persona`.
- `generaLetraDNI($idLetra)`: devuelve la letra en la posición indicada ('T', 'R', 'W', 'A', 'G', 'M', 'Y', 'F', 'P', 'D', 'X', 'B', 'N', 'J', 'Z', 'S', 'Q', 'V', 'H', 'L', 'C', 'K', 'E'). No debe ser visible fuera de la clase

Programa Principal

```
<?php
/**
 * @author Silvia Vilar
 * Ej4UD8 - mainPersona.php
 */
include "Persona.php";
//Obtenemos los datos de la persona
print "Introduce el nombre de la persona: ";
$nombre= readline();
print "Introduce la edad de la persona: ";
$edad=intval(readline());
print "Introduce el sexo de la persona (H o M): ";
$sexo= readline();
print "Introduce el peso de la persona: ";
$peso=floatval(readline());
print "Introduce la altura de la persona: ";
$altura=floatval(readline());
//Creamos 3 personas usando cada uno de los constructores
$p1 = Persona::consFull($nombre, $edad, $sexo, $peso, $altura);
$p2 = Persona::consNomEdSex("Maria", 18, 'M');
$p2->setPeso(50);
$p2->setAltura(1.65);
$p3 = new Persona();
$p3->setNombre("Juan");
$p3->setEdad(16);
$p3->setSexo('Z');
$p3->setPeso(70);
$p3->setAltura(1.70);
//Calculamos el IMC
print $p1->mostrarIMC();
```

```
print $p2->mostrarIMC();
print $p3->mostrarIMC();
//Indicamos si es mayor de edad
$p1->esMayorDeEdad();
$p2->esMayorDeEdad();
$p3->esMayorDeEdad();
//Imprimimos los datos de las personas
print " PERSONA 1 ";
print $p1;
print " PERSONA 2 ";
print $p2;
print " PERSONA 3 ";
print $p3;
print " FIN DEL PROGRAMA ";
?>
```

Salida de la terminal:

```
Introduce el nombre de la persona: Paco
Introduce la edad de la persona: 23
Introduce el sexo de la persona (H o M): j
Introduce el peso de la persona: 80
Introduce la altura de la persona: 1.7
Paco tiene sobrepeso<br>
Maria está por debajo de su peso ideal<br>
Juan está en su peso ideal<br>
Paco con DNI 90944993A es mayor de edad<br>
Maria con DNI 23063125J es mayor de edad<br>
Juan con DNI 37204149Q es menor de edad<br>
PERSONA 1 Informacion de la persona:<br>
DNI: 90944993A
    Nombre: Paco
    Sexo: Hombre
    Edad: 23
    Peso: 80 Kg
    Altura: 1.7 metros
    Resultado IMC: Paco tiene sobrepeso<br>
PERSONA 2 Informacion de la persona:<br>
DNI: 23063125J
    Nombre: Maria
    Sexo: Mujer
    Edad: 18
    Peso: 50 Kg
    Altura: 1.65 metros
    Resultado IMC: Maria está por debajo de su peso ideal<br>
PERSONA 3 Informacion de la persona:<br>
DNI: 37204149Q
    Nombre: Juan
    Sexo: Hombre
```

```
Edad: 16
Peso: 70 Kg
Altura: 1.7 metros
Resultado IMC: Juan está en su peso ideal<br>
FIN DEL PROGRAMA
```

5. En una biblioteca tenemos dos tipos de Publicaciones: Libros y Revistas. Toda publicación tiene su ISBN, título y año (por defecto será 2024). Las publicaciones no se pueden instanciar. Las Revistas tienen además un número de Publicación y los Libros son Prestables. El que un Libro sea Prestable, implica que tenga los métodos estaPrestado, prestar y devolver. Los objetos deben implementar el método __toString para poder ser imprimidos con toda su información. Implementa las clases e interfaces necesarias para poder obtener la siguiente salida:

Programa Principal

```
<?php
/**
 * @author Silvia Vilar
 * Ej5UD8 - mainBiblioteca.php
 */
include_once 'Libro.php';
include_once 'Revista.php';
$libro1 = new Libro("123456", "La Ruta Prohibida", 2007);
$libro2 = new Libro("112233", "Los Otros", 2016);
$libro3 = new Libro("456789", "La rosa del mundo", 1995);
$revista1 = new Revista("444555", "Año Cero", 2019, 344);
$revista2 = new Revista("002244", "National Geographic", 2003, 255);
print($libro1);
print($libro2);
print($libro3);
print($revista1);
print($revista2);
$libro2->presta();
$libro2->mostrarPrestado();
$libro2->presta();
$libro2->devuelve();
$libro2->mostrarPrestado();
$libro3->presta();
print($libro2);
print($libro3);
?>
```

Salida de la terminal

```
ISBN: 123456, título: La Ruta Prohibida, año de publicación: 2007 (no prestado)<br>
ISBN: 112233, título: Los Otros, año de publicación: 2016 (no prestado)<br>
ISBN: 456789, título: La rosa del mundo, año de publicación: 1995 (no prestado)<br>
ISBN: 444555, título: Año Cero, año de publicación: 2019, número: 344<br>
ISBN: 002244, título: National Geographic, año de publicación: 2003, número: 255<br>
Se ha prestado el libro 'Los Otros'.<br>
El libro 'Los Otros' está prestado<br>
```

```
No se ha podido prestar, el libro 'Los Otros' ya está prestado.<br>
Se ha devuelto el libro 'Los Otros'.<br>
Se ha prestado el libro 'La rosa del mundo'.<br>
ISBN: 112233, título: Los Otros, año de publicación: 2016 (no prestado)<br>
ISBN: 456789, título: La rosa del mundo, año de publicación: 1995 (prestado)<br>
```

6. Crea una clase `Animal` que no se pueda instanciar y contenga el atributo `sexo` (“H” para hembra o “M” para macho). Se debe llevar la cuenta de los animales que se van creando. Debe poder permitid crear un animal sin indicar atributos (por defecto el sexo será “M” o crear animales indicando el sexo. Esta clase tendrá los métodos `dormirse()`, `alimentarse($comida)` y `morirse()` que imprimirán un texto descriptivo y modificar el número de animales si es necesario. Al imprimir los datos de un animal, se debe indicar que es un animal.

Crea una clase `Ave` subclase de `Animal` no instanciable que lleve la cuenta del número de aves que se crean en sus constructores. Añade un método `ponerHuevo()` y actualiza datos en el caso de muerte del animal. Al imprimir el ave, debe añadir que es un ave. Crea otra clase `Mamifero` subclase de `Animal`, no instanciable, que lleve la cuenta del número de mamíferos que se crean en sus constructores. Añade un método `amamantar()` y actualiza datos en el caso de muerte del animal. Al imprimir el mamífero, debe añadir que es un mamífero.

Crea las clases `Canario` (`Ave`), `Gato` (`Mamifero`), `Lagarto` (`Animal`), `Perro` (`Mamifero`) y `Pingüino` (`Ave`), todas ellas con atributo `nombre` y en el caso de `Perro` y `Gato` con el atributo `raza`. Añade un constructor o dos para poner los nuevos atributos `nombre` y `raza` según sea necesario. Deberán redefinir los métodos heredados para añadir qué animal son y cuál es su nombre. Al imprimirlos, deberán describirse completamente como animales, indicar el sexo con la palabra `HEMBRA` o `MACHO` en mayúsculas, en lugar de la primera letra y su nombre si lo tienen (pueden no tener nombre excepto el `Perro` y el `Gato`). A continuación se indican los métodos propios:

`Canario`: método `pia()` que muestra “Canario Nombre: Pio pio pio”. El canario come alpiste

`Gato`: método `maulla()` que muestra “Gato Nombre: Miauuuu”. El gato come pescado

`Lagarto`: método `tomarSol()` que muestra “Lagarto Nombre: Estoy tomando el sol”. El lagarto come insectos

`Perro`: método `ladra()` que muestra “Perro Nombre: Guau guau”. El perro come carne

`Pingüino`: método `pia()` que muestra “Pingüino Nombre: Soy un pingüino programando en PHP”. El pingüino come peces.

Programa Principal

```
<?php
/**
 * @author Silvia Vilar
 * Ej6UD8 - PruebaAnimales.php
 */
include_once "Canario.php";
include_once "Pinguino.php";
include_once "Perro.php";
include_once "Lagarto.php";
include_once "Gato.php";
print Animal::getTotalAnimales();
```



```
print Ave::getTotalAves();
print Mamifero::getTotalMamiferos();
//creamos varios animales de Lagarto
$godzilla = new Lagarto();
$godzilla->setNombre("Godzilla");
$diana = Lagarto::consSexo("H");
$diana->setNombre("Diana");
$juancho = Lagarto::consFull("M", "Juancho");
print Animal::getTotalAnimales();
print($godzilla);
$godzilla->tomarSol();
$godzilla->dormirse();
$godzilla->morirse();
print($diana);
$diana->tomarSol();
print($juancho);
$juancho->alimentarse();
print Animal::getTotalAnimales();
//creamos varios animales de Canario
$amarillo=new Canario();
$amarillo->setNombre("Amarillo");
$canaria=Canario::consSexo("H");
$canaria->setNombre("Canaria");
$piolin=Canario::consFull("M","Piolín");
print Animal::getTotalAnimales();
print Ave::getTotalAves();
print($piolin);
$piolin->alimentarse();
$piolin->ponerHuevo();
print($amarillo);
$amarillo->pia();
$amarillo->morirse();
print($canaria);
$canaria->alimentarse();
$canaria->ponerHuevo();
print Animal::getTotalAnimales();
print Ave::getTotalAves();
//creamos varios animales de Pinguino
$p1=new Pinguino();
$pingui=Pinguino::consSexo("H");
$pingui->setNombre("Pingu");
$tux=Pinguino::consFull("H","TUX");
print Animal::getTotalAnimales();
print Ave::getTotalAves();
print($tux);
$tux->alimentarse();
$tux->programar();
$tux->dormirse();
```

```
print($pingui);
$pingui->alimentarse();
$pingui->ponerHuevo();
print($p1);
$p1->programar();
$p1->morirse();
print Animal::getTotalAnimales();
print Ave::getTotalAves();
//creamos varios animales de Perro
$perro=new Perro();
$laika = Perro::consSexoNombre("H", "Laika");
$toby = Perro::consFull("M", "Toby", "Cocker");
print Animal::getTotalAnimales();
print Ave::getTotalAves();
print Mamifero::getTotalMamiferos();
print($perro);
$perro->alimentarse();
$perro->morirse();
print($laika);
$laika->dormirse();
$laika->alimentarse();
$laika->amamantar();
print($toby);
$toby->ladra();
$toby->dormirse();
$toby->amamantar();
print Animal::getTotalAnimales();
print Ave::getTotalAves();
print Mamifero::getTotalMamiferos();
//creamos varios animales de Gato
$cat=new Gato();
$sofia=Gato::consSexoNombre("H","Sofía");
$isidoro=Gato::consFull("M","Isidoro","Persa");
print Animal::getTotalAnimales();
print Mamifero::getTotalMamiferos();
print($cat) ;
$cat->alimentarse();
$cat->morirse();
print($sofia);
$sofia->maulla();
$sofia->amamantar();
$sofia->dormirse();
print($isidoro);
$isidoro->alimentarse();
$isidoro->maulla();
$isidoro->dormirse();
//recuento final
print Animal::getTotalAnimales();
```

```
print Ave::getTotalAves();  
print Mamifero::getTotalMamiferos();  
print("Fin del Programa");  
?>
```

Salida de la terminal

```
Hay un total de 0 animales<br>  
Hay un total de 0 aves<br>  
Hay un total de 0 mamíferos<br>  
Hay un total de 3 animales<br>  
Soy un Animal, en concreto un Lagarto, con sexo MACHO, llamado Godzilla<br>  
Lagarto Godzilla: Estoy tomando el Sol<br>  
Lagarto Godzilla: Zzzzzzz<br>  
Lagarto Godzilla: Adiós!<br>  
Soy un Animal, en concreto un Lagarto, con sexo HEMBRA, llamado Diana<br>  
Lagarto Diana: Estoy tomando el Sol<br>  
Soy un Animal, en concreto un Lagarto, con sexo MACHO, llamado Juancho<br>  
Lagarto Juancho: Estoy comiendo insectos<br>  
Hay un total de 2 animales<br>  
Hay un total de 5 animales<br>  
Hay un total de 3 aves<br>  
Soy un Animal, un Ave, en concreto un Canario, con sexo MACHO, llamado Piolín<br>  
Canario Piolín: Estoy comiendo alpiste<br>  
Canario Piolín: Soy macho, no puedo poner huevos<br>  
Soy un Animal, un Ave, en concreto un Canario, con sexo MACHO, llamado Amarillo<br>  
Canario Amarillo: Pio pio pio<br>  
Canario Amarillo: Adiós!<br>  
Soy un Animal, un Ave, en concreto un Canario, con sexo HEMBRA, llamado Canaria<br>  
Canario Canaria: Estoy comiendo alpiste<br>  
Canario Canaria: He puesto un huevo!<br>  
Hay un total de 4 animales<br>  
Hay un total de 2 aves<br>  
Hay un total de 7 animales<br>  
Hay un total de 5 aves<br>  
Soy un Animal, un Ave, en concreto un Pingüino, con sexo HEMBRA, llamado TUX<br>  
Pingüino TUX: Estoy comiendo peces<br>  
Pingüino TUX: Soy un pingüino programador en PHPPingüino TUX: Zzzzzzz<br>  
Soy un Animal, un Ave, en concreto un Pingüino, con sexo HEMBRA, llamado Pingui<br>  
Pingüino Pingui: Estoy comiendo peces<br>  
Pingüino Pingui: He puesto un huevo!<br>  
Soy un Animal, un Ave, en concreto un Pingüino, con sexo MACHO<br>  
Pingüino : Soy un pingüino programador en PHPPingüino : Adiós!<br>  
Hay un total de 6 animales<br>  
Hay un total de 4 aves<br>  
Hay un total de 9 animales<br>  
Hay un total de 4 aves<br>  
Hay un total de 3 mamíferos<br>
```

Soy un Animal, un Mamífero, en concreto un Perro, con sexo MACHO, raza teckel y no tengo nombre

Perro : Estoy comiendo carne

Perro : Adiós!

Soy un Animal, un Mamífero, en concreto un Perro, con sexo HEMBRA, raza teckel y mi nombre es Laika

Perro Laika: Zzzzzzz

Perro Laika: Estoy comiendo carne

Perro Laika: Amamantando a mis crías

Soy un Animal, un Mamífero, en concreto un Perro, con sexo MACHO, raza Cocker y mi nombre es Toby

Perro Toby: Guau guau

Perro Toby: Zzzzzzz

Perro Toby: Soy macho, no puedo amamantar

Hay un total de 8 animales

Hay un total de 4 aves

Hay un total de 2 mamíferos

Hay un total de 11 animales

Hay un total de 5 mamíferos

Soy un Animal, un Mamífero, en concreto un Gato, con sexo MACHO, raza y no tengo nombre

Gato : Estoy comiendo pescado

Gato : Adiós!

Soy un Animal, un Mamífero, en concreto un Gato, con sexo HEMBRA, raza y mi nombre es Sofía

Gato Sofía: Miauuuu

Gato Sofía: Amamantando a mis crías

Gato Sofía: Zzzzzzz

Soy un Animal, un Mamífero, en concreto un Gato, con sexo MACHO, raza y mi nombre es Isidoro

Gato Isidoro: Estoy comiendo pescado

Gato Isidoro: Miauuuu

Gato Isidoro: Zzzzzzz

Hay un total de 10 animales

Hay un total de 4 aves

Hay un total de 4 mamíferos

Fin del Programa