

Pandemia

España 2030. Un virus desconocido está haciendo estragos en un lejano país de oriente. Según el número de contagios observado y la velocidad a la que se propaga se teme que en pocas semanas se extienda por todo el mundo y llegue hasta nosotros.

Las autoridades locales han solicitado nuestra ayuda. Necesitan una simulación de cómo afectará el virus a nuestra población y a nuestras mascotas ya que se ha demostrado que los humanos pueden contagiar también a sus mascotas pero solamente a perros y gatos.

Para realizar la simulación de cómo afectaría la pandemia a una población determinada, el programa pedirá la cantidad de personas y mascotas que forman el total de la población. Nos interesa saber cómo afecta el virus particularmente a los sanitarios y a los cuerpos de seguridad por ser personal esencial, por lo tanto tendremos que conocer la cantidad total de sanitarios, la cantidad total de personal que forma parte de los cuerpos de seguridad y el resto de población que llamaremos población civil.

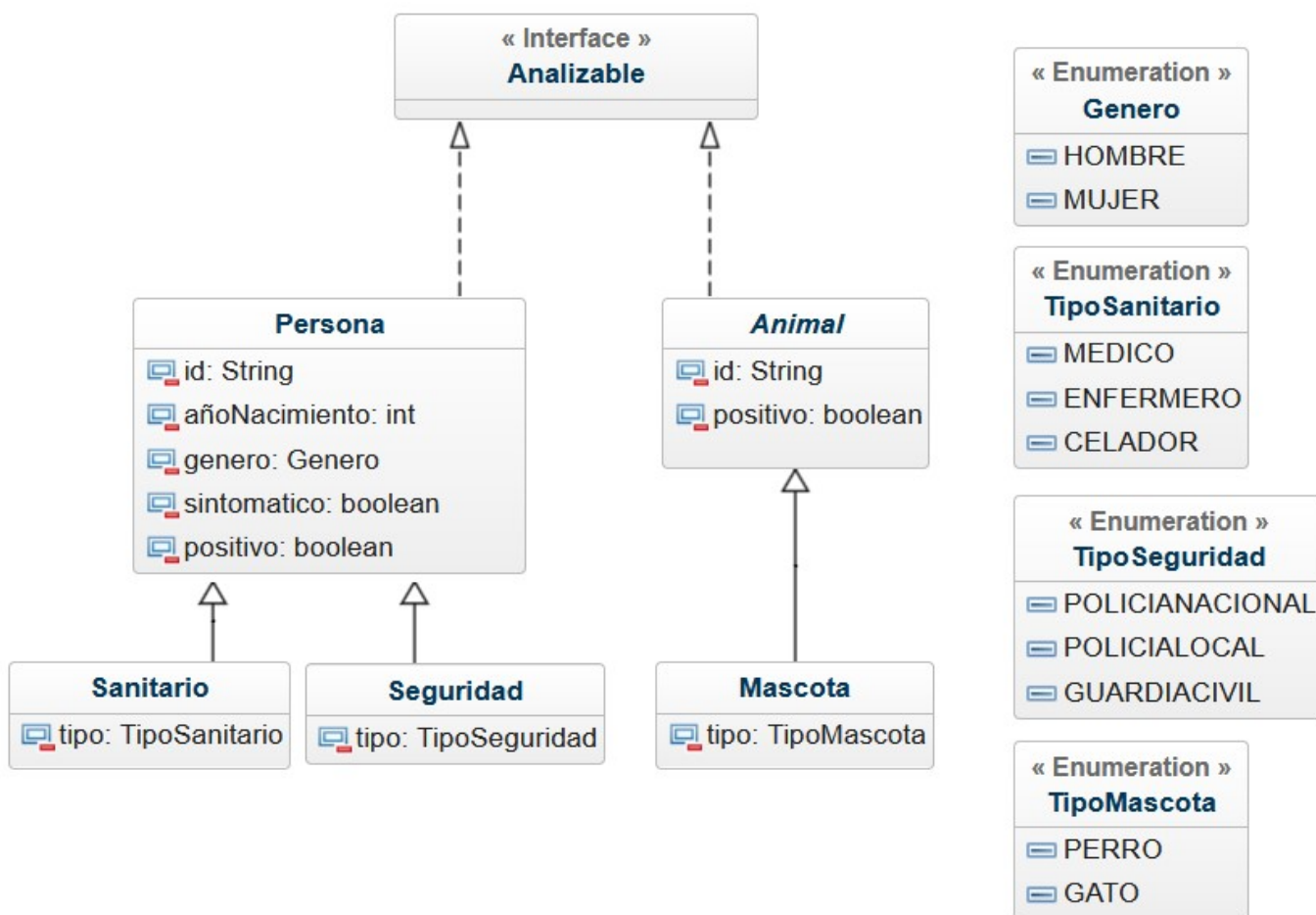
Crearemos tantos objetos de cada clase como nos hayan indicado.

Como estructura de datos para almacenar los objetos se utilizará un **ArrayList**.

El programa realizará un contagio aleatorio de los integrantes de la población.

Finalmente se mostrarán una serie de estadísticas sobre la incidencia del virus en la población.

El diagrama de clases de la aplicación es el siguiente:



Clase Persona

Contiene los atributos:

Id de tipo String. Es el identificador asignado a cada persona. Está formado por una letra mayúscula seguida de cinco dígitos.

añoMacimiento de tipo int.

genero es un **enum** con 2 valores posibles: HOMBRE, MUJER.

sintomático de tipo boolean, su valor es true si la persona presenta algún síntoma.

positivo de tipo boolean, su valor es true si la persona está contagiada.

Los objetos de esta clase representan a la población civil. Serán todas las personas de la población excluidos los sanitarios y los cuerpos de seguridad.

Clase Sanitario

Contiene el atributo *tipo*. Es un enum con 3 valores posibles: MEDICO, ENFERMERO, CELADOR.

Los objetos de esta clase representan al personal sanitario de la población.

Clase Seguridad

Contiene el atributo *tipo*. Es un enum con 3 valores posibles: POLICIANACIONAL, POLICIALLOCAL, GUARDIACIVIL.

Los objetos de esta clase representan a los cuerpos de seguridad presentes en la población.

Clase Animal

Clase Abstracta.

Contiene los atributos:

Id de tipo String. Es el identificador asignado a cada animal. Está formado por dos letras mayúsculas seguidas de dos dígitos.

positivo de tipo boolean, su valor es true si el animal está contagiado.

Clase Mascota

Contiene el atributo *tipo*. Es un enum con 2 valores posibles: PERRO, GATO.

Los objetos de esta clase representan a las mascotas que hay en la población.

Interface Analizable

Contiene los métodos:

- void contagiar();

Este método asigna el valor **true** al atributo *positivo* en personas y animales.

- String obtenerTipo();

Este método devuelve un String que indica el tipo de objeto según el siguiente formato:

Para las Mascotas devuelve "PERRO" o "GATO" según su tipo.

Para las Personas (población civil) devuelve el String "CIV".

Para el personal sanitario devuelve "MED", "ENF" o "CEL" según sean médicos, enfermeros o celadores.

Para los cuerpos de seguridad devuelve "PN", "PL" o "GC" según sean policía nacional, policía local o guardia civil.

- String obtenerId();

Este método devuelve el id de las personas y los animales.

No se puede añadir más métodos a la interface.

Clase Principal

El método main de la clase principal contendrá las siguientes llamadas a métodos:

```
public static void main(String[] args) {  
    crearPoblacion();  
    contagiarPoblacion();  
    mostrarPoblacion();  
    mostrarEstadisticas();  
}
```

Método crearPoblacion()

En este método se crearán todos los objetos (personas y mascotas) que forman parte de la población y que intervendrán en la simulación de la pandemia.

Primero se pide la cantidad de personas, separando entre población civil, sanitarios y cuerpos de seguridad, y a continuación el número de mascotas.

Para que el estudio sea fiable, el tamaño de la población civil debe ser mayor o igual a 50 personas.

El número de sanitarios, cuerpos de seguridad y mascotas puede ser cero.

Un ejemplo de lectura de datos puede ser este:

```
Introduce tamaño de la población civil (>=50): 1260  
Introduce número de sanitarios (>=0): 20  
Introduce número de personal de seguridad (>=0): 12  
Introduce número de mascotas (>=0): 47
```

A continuación hay que crear todos estos objetos y añadirlos al ArrayList.

Para crear los objetos no habrá que introducir datos por teclado, todos los valores se obtendrán de forma aleatoria de la siguiente forma:

Para las **personas** (población civil):

Id: a cada persona se le asigna un identificador formado por una letra mayúscula seguida de cinco dígitos. Para formar este String se obtendrá una letra mayúscula al azar y las cinco cifras siguientes también al azar. El Id de la persona será la unión de la letra inicial más las 5 cifras obtenidas. No puede haber personas con id iguales. Si se obtiene un id que ya existe se vuelve a calcular otro.

añoNacimiento: Se obtiene un valor entero al azar con las siguientes limitaciones:

El valor del año de nacimiento de las personas debe ser menor o igual que el año actual.

Consideramos que en la población no hay personas mayores de 110 años.

Puede haber personas nacidas este mismo año.

genero: se obtiene un número al azar y se le asigna un valor del enum Genero (HOMBRE o MUJER) según el valor aleatorio obtenido.

sintomatico: se obtiene un número al azar y se le asigna true o false según el valor obtenido. Un valor true indica que la persona presenta algún síntoma (tos, catarro, dolor de cabeza, fiebre, etc.) aunque no quiere decir que esté contagiado por el virus.

Para los **sanitarios**: Además de los valores para las personas hay que obtener el valor del atributo *tipo*.

tipo: se obtiene un número al azar y se le asigna un valor de enum TipoSanitario (MEDICO, ENFERMERO, CELADOR) según el valor aleatorio obtenido.

El personal sanitario debe tener entre 18 y 65 años. Cuando se obtenga su año de nacimiento habrá que comprobarlo. Si es menor de 18 o mayor de 65 se obtendrá un nuevo año de nacimiento.

Para el personal de **seguridad**: Además de los valores para las personas hay que obtener el valor del atributo *tipo*.

tipo: se obtiene un número al azar y se le asigna un valor de enum TipoSeguridad (POLICIANACIONAL, POLICIALLOCAL, GUARDIACIVIL) según el valor aleatorio obtenido.

El personal de seguridad debe tener entre 18 y 65 años. Cuando se obtenga su año de nacimiento habrá que comprobarlo. Si es menor de 18 o mayor de 65 se obtendrá un nuevo año de nacimiento.

Para las **mascotas**:

Id: a cada mascota se le asigna un identificador formado por dos letras mayúsculas seguidas de dos dígitos. Para formar este String se obtendrá dos letras mayúsculas al azar y las dos cifras siguientes también al azar. El Id de la mascota será la unión de las dos letras más las 2 cifras obtenidas. No puede haber mascotas con id iguales. Si se obtiene un id que ya existe se vuelve a calcular otro.

tipo: se obtiene un número al azar y se le asigna un valor de enum TipoMascota (PERRO, GATO) según el valor aleatorio obtenido.

Como se puede observar, se asignan valores aleatorios a todos los atributos de las clases excepto al atributo *positivo*. Cuando se crean, el valor de este atributo será *false* (que es el valor por defecto de los boolean). Este valor solamente se modifica en el método *contagiar()*.

Método *contagiarPoblacion()*

Una vez creados los objetos que forman el total de integrantes de la población, el siguiente paso es simular el proceso de contagio.

Para el contagio se realizará lo siguiente:

Primero se obtiene un número N al azar que corresponderá al número de contagios que se producirán. El valor de este número estará comprendido entre 0 y el tamaño total de la población (personas + mascotas).

A continuación se obtienen N posiciones al azar del ArrayList y mediante el método *contagiar()* se produce su contagio.

Por ejemplo si la población total entre personas y mascotas son 10589 se obtendrá un número N al azar entre 0 y 10589. Supongamos que hemos obtenido N = 2043. Se trata de repetir 2043 veces lo siguiente: obtener una posición al azar y realizar el contagio.

Es posible que una misma posición del ArrayList se obtenga varias veces. Es decir, que intentemos contagiar una posición que ya está contagiada. En este caso no hay que hacer nada especial ni volver a obtener otra posición. Esa persona o mascota seguirá teniendo el atributo *positivo* con valor *true* independientemente de que se le haya contagiado 1 o 20 veces.

Método *mostrarPoblacion()*

Este método muestra el estado de la población después del contagio.

El listado se muestra ordenado alfabéticamente por id.

Primero se muestran las personas y a continuación las mascotas.

El listado se mostrará en el formato que aparece en el ejemplo.

Por ejemplo, si hemos introducido estos datos de población:

```
Introduce tamaño de la población civil (>=50): 15589
Introduce número de sanitarios (>=0): 220
Introduce número de personal de seguridad (>=0): 60
Introduce número de mascotas (>=0): 55
```

Se mostrará el siguiente listado. Como el listado de personas es bastante largo se muestra solo el principio, una parte del centro y el final

Listado de personas

```

A00035 CIV H Edad: 95 P S | A00185 CIV M Edad: 103 P A | A00384 CIV H Edad: 38 N S | A00478 ENF M Edad: 28 P A
A00508 ENF M Edad: 20 N A | A00603 CIV M Edad: 49 N S | A00650 CIV H Edad: 20 N S | A00760 CIV M Edad: 74 N S
A00814 CIV M Edad: 1 P A | A00915 CIV M Edad: 25 N A | A01356 CEL H Edad: 63 N A | A01491 CIV M Edad: 80 P A
A01556 CIV H Edad: 13 P S | A01699 CIV H Edad: 62 N A | A01758 CIV M Edad: 68 N S | A02009 CIV H Edad: 39 N S
A02226 CIV M Edad: 53 P A | A02331 CIV H Edad: 96 N S | A02423 CIV H Edad: 26 N A | A02427 CIV H Edad: 65 P S
A02976 CIV M Edad: 79 P A | A03281 CIV H Edad: 80 P A | A03407 CIV H Edad: 102 P S | A03686 CIV H Edad: 54 P A
A03727 CIV H Edad: 90 P A | A03798 CIV H Edad: 54 N S | A03822 CIV M Edad: 21 N S | A03913 CIV H Edad: 21 N S
A03962 CIV H Edad: 27 P A | A04121 PN H Edad: 23 N A | A04144 CIV H Edad: 35 N A | A04155 CIV H Edad: 43 N A

.....

M99066 CIV H Edad: 10 P A | M99449 CIV H Edad: 105 P A | M99543 CIV M Edad: 14 N S | M99563 PL M Edad: 59 P S
M99727 CIV M Edad: 47 P S | M99763 CIV M Edad: 73 P A | M99917 CIV H Edad: 90 N S | N00165 CIV M Edad: 53 P A
N00353 CIV H Edad: 13 N A | N00358 MED H Edad: 60 N A | N00637 CIV M Edad: 89 P S | N00739 CIV H Edad: 69 N A
N00808 CIV H Edad: 54 P S | N00819 CIV M Edad: 23 P S | N00987 CIV M Edad: 62 P S | N01022 CIV H Edad: 13 P A
N01045 CIV H Edad: 59 P S | N01342 CIV M Edad: 16 N A | N01500 MED M Edad: 28 P A | N01512 CIV H Edad: 89 N S

.....

Z96475 GC M Edad: 32 P S | Z96502 CIV M Edad: 36 N A | Z96711 CIV H Edad: 87 P A | Z96775 CIV H Edad: 98 N A
Z97374 CIV M Edad: 52 N A | Z97376 CIV M Edad: 97 P S | Z97509 CIV M Edad: 38 N S | Z97590 CIV H Edad: 37 N S
Z97639 CIV H Edad: 21 N S | Z97743 CIV M Edad: 3 N A | Z97769 CIV H Edad: 33 P A | Z97798 CIV H Edad: 61 N S
Z98297 CIV H Edad: 58 N A | Z99030 CIV H Edad: 92 N S | Z99252 CIV M Edad: 99 N S | Z99457 CIV M Edad: 53 N S
Z99584 CIV H Edad: 82 P S |

```

Listado de mascotas

```

AF15 PERRO P | AN51 GATO N | BK38 GATO N | BV97 GATO P | CF04 PERRO P | EB95 GATO N
EI24 GATO P | EQ92 PERRO N | ER86 PERRO P | ES93 GATO N | FM54 PERRO N | FV00 PERRO P
GF51 PERRO N | HB56 GATO P | HN75 PERRO P | IA27 PERRO N | IC26 GATO N | JI47 GATO N
JW74 PERRO N | KD18 PERRO P | KS95 PERRO P | KT93 GATO P | LE52 PERRO P | LQ29 PERRO P
NG05 GATO P | NM34 PERRO P | NZ73 GATO N | OC67 PERRO N | OT99 GATO N | PD14 GATO N
PI37 GATO P | PR46 GATO N | QN23 GATO P | QO70 PERRO N | QX12 PERRO N | QY85 PERRO P
RE91 GATO N | SH78 GATO P | SI61 GATO N | SO14 PERRO N | SO41 PERRO P | UO55 PERRO N
VO45 PERRO P | VQ99 PERRO P | VV63 GATO P | XS64 PERRO P | YC17 PERRO P | YG11 PERRO N
YG44 PERRO N | YJ32 GATO P | YM01 PERRO N | YS72 GATO P | ZF29 GATO N | ZQ89 GATO N
ZR07 PERRO P |

```

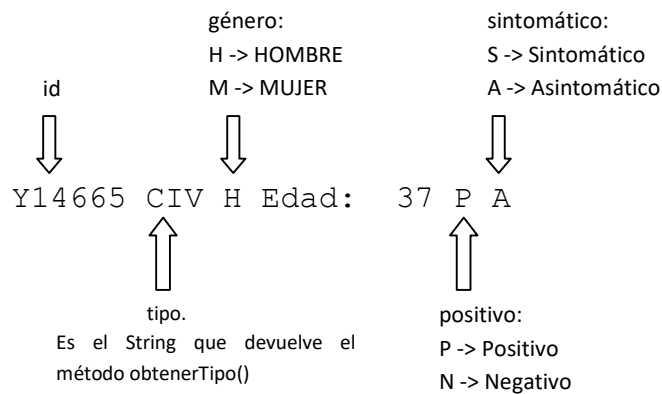
Tanto el listado de personas como el de mascotas se muestra ordenado alfabéticamente por id.

En el listado de personas se muestran 4 personas por fila separadas por el carácter |

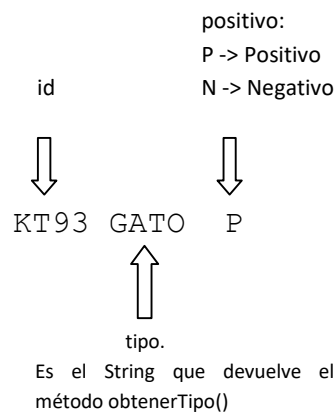
En el listado de mascotas se muestran 6 mascotas por fila separadas por el carácter |

Los datos que aparecen en el listado corresponden a lo siguiente:

Para cada persona:



Para cada mascota:



Método mostrarEstadísticas()

Finalmente este método calculará y mostrará diferentes estadísticas sobre la incidencia de la pandemia en la población.

Los datos a mostrar son los que se muestran en el ejemplo.

La estructura y contenido de la salida por pantalla debe ser igual a la mostrada.

Siguiendo con el ejemplo de población anterior, en este caso el efecto de la simulación ha sido el siguiente:

```
Resultados
-----
Total Personas:      15869   Contagiados:  8090 50,98%   Contagiados Asintomáticos:  4036 49,89%

Contagios por tramos de edad
-----
 0 - 20 ->   1502  18,57% Asintomáticos ->   770  51,26%
21 - 40 ->   1873  23,15% Asintomáticos ->   917  48,96%
41 - 60 ->   1925  23,79% Asintomáticos ->   929  48,26%
61 -    ->   2790  34,49% Asintomáticos ->  1420  50,90%

Contagios en sanitarios y cuerpos de seguridad del estado
-----
Total Sanitarios:    220   Contagiados:   115  52,27% MED:    42 ENF:    42 CEL:    31
Total Seguridad:     60   Contagiados:    29  48,33% PL:     6 PN:     7 GC:    16

Contagios en mascotas
-----
Total Mascotas:      55   Contagiadas:   28  50,91% PERROS:   18 GATOS:   10
```