

## CUESTIONES Y EJERCICIOS

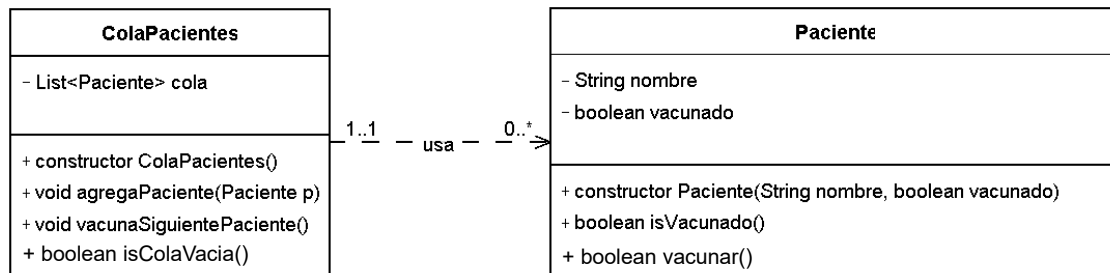
1. Vamos a crear la clase `NombresUnicos` correspondiente al siguiente diagrama UML, teniendo en cuenta que:

- La clase usará una lista de cadenas para almacenar nombres de personas SIN DUPLICADOS.
- El constructor de la clase debe inicializar la propiedad *lista*.
- El método *agregaNombre* recibirá una cadena como parámetro y la añadirá a la lista solo en el caso de que no existiese es nombre previamente en la lista. Para ello se considerarán iguales dos nombres independientemente de la capitalización de las letras. Ej: Ana = ANA = ana
- Además, el método *agregaNombre* escribirá un error en la consola en el caso de que se intente guardar un nombre que ya había sido almacenado previamente.
- El método *ordenaNombres* ordenará alfabéticamente la lista, sin imprimir nada.
- El método *imprimeNombres* utilizará un iterador para imprimir los nombres en el siguiente formato:
  1. Ana
  2. Juan
  3. Pedro
- El método *contiene* devolverá un lógico que indica si la cadena que se recibe como parámetro está contenida o no en la lista.

NombresUnico:
- List<String> lista
+ constructor NombresUnicos() + void agregaNombre(String nombre) + void ordenaNombres() + void imprimeNombres() + boolean contiene(String nombre)

Por último, realiza una clase *PruebaNombresUnicos* con un método *main* que cree un objeto de la clase *NombresUnicos* y pruebe todos sus métodos.

2. Vamos a crear las clases correspondientes al siguiente diagrama UML:



Teniendo en cuenta que:

- En la clase Paciente, el método “vacunar” cambia el valor de la propiedad *vacunado* a *true* si es que el paciente no estaba vacunado. El método devolverá *true* si se consiguió vacunar al paciente o *false* si ya estaba vacunado.
- En la clase ColaPacientes se utilizará una lista para modelar la cola de los pacientes para recibir la vacuna. Esto implica que siempre añadiremos un nuevo paciente al final de la cola.
- El constructor de la clase ColaPacientes debe inicializar la propiedad tipo List.
- El método *vacunaSiguientePaciente* comprobará si hay pacientes en la cola. En el caso de que sí haya pacientes, tomará el primer paciente de la cola y lo intentará vacunar. En el caso de que no se pudiera vacunar se debe imprimir el siguiente mensaje en la pantalla “¡Hay que estar aburrido para querer vacunarse 2 veces!”. Tanto si el paciente se vacunó correctamente como si no se debe extraer el paciente de la cola dejándola lista para atender al siguiente.

Realiza también una clase PruebaColaPacientes con un método *main* que cree un objeto ColaPacientes y le agregue 4 pacientes, 3 de los cuales estarán sin vacunar y 1 vacunado.

Después se intentarán vacunar a todos los pacientes de la cola, para ello debes realizar un bucle que atienda a todos los pacientes de la cola hasta que quede vacía. Finalmente, se imprimirá el mensaje “Todos los pacientes están vacunados” y terminaremos el programa.