



REPASO PROGRAMACION– 2º DAM

PRUEBA de EVALUACIÓN INICIAL curso 2023/2024

Nombre y Apellidos:

Puntuación:

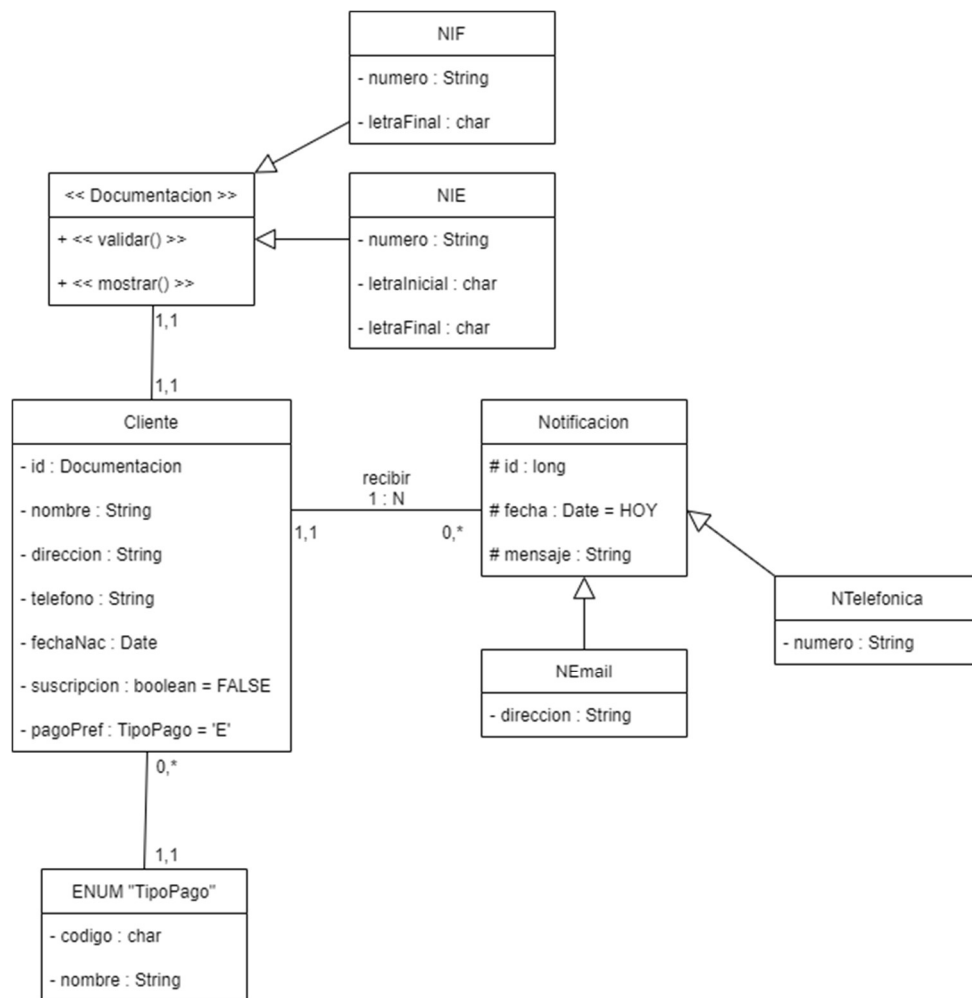
NOTA: Esta prueba corresponde con una evaluación inicial de diagnóstico y no se tendrá en cuenta como instrumento de evaluación del módulo. Servirá como medio para recabar información sobre las cualidades como desarrollador de los alumnos/as. Se pondrá a disposición del alumnado una propuesta de solución a la prueba, con el objetivo de autoevaluar el propio trabajo llevado a cabo, su validez y calidad técnica.

El sistema guarda información sobre *Clientes*:

- Se tiene un campo **id** de tipo *Documentacion*, que es una clase abstracta con métodos abstractos *validar()* y *mostrar()*. De ella heredan las clases *NIF* y *NIE*, que implementan esos métodos propiamente de acuerdo al formato propio de cada uno (ver anexo con formatos de NIF y NIE).
- Un campo **nombre** para el nombre y apellidos del cliente.
- Un campo **direccion** para la dirección postal del cliente.
- Un campo **telefono** para el teléfono del cliente.
- Un campo **fechaNac** para la fecha de nacimiento del cliente. Será a través de este campo del que se establecerá la ordenación de los clientes. Podrá usarse cualquier tipo de dato para la fecha que se estime oportuno, siempre que se muestren con el formato dd/mm/aaaa
- Un campo **suscripcion** que marcará si el cliente se ha suscrito o no. Por defecto, no estará suscrito.
- Un campo **pagoPref** de tipo *TipoPago*, que es una enumeración para los siguientes únicos tipos de pago aceptados en el sistema: 'E' para "efectivo", 'T' para "transferencia" y 'C' para tarjeta de crédito.

El sistema también almacena información sobre las *Notificaciones* recibidas por los clientes:

- Se tiene un campo **id** para identificar unívocamente a cada una de ellas.
- Un campo **fecha** para la fecha de la notificación, que por defecto será el día actual.
- Un campo **mensaje** con la transcripción del mensaje de la notificación.
- Hay 2 tipos de notificaciones, las realizadas por correo electrónico, *NEmail*, de las que se guarda el campo **direccion** con el email del destinatario de la notificación, y las realizadas telefónicamente, *NTelefonica*, de las que se guarda el campo **telefono** con el número de teléfono del destinatario de la notificación.



Se pide implementar el diagrama de clases al completo de la figura anterior, de acuerdo a las especificaciones siguientes:

- Todas las entidades contarán con un constructor por defecto y otro con todos los atributos propios de cada entidad. Tener en cuenta las relaciones de herencia establecidas en el diagrama.
- Todas las entidades contarán con métodos *getter/setter* para cada atributo de la entidad, así como una redefinición del método *toString()* para formatear la salida.
- Habrá que codificar la relación *recibir* (1:N) entre las entidades *Cliente* y *Notificacion*. Queda a elección del alumno/a el diseño que estime mejor o más oportuno.

Incluir en la clase *Cliente* un método estático de nombre *nuevoCliente()*, que no necesita argumentos y devuelva un objeto *Cliente* completo y en cuyo cuerpo se pida al usuario por pantalla que introduzca valores para todos y cada uno de los campos de esa entidad, de acuerdo al tipo de dato de cada uno de ellos. Ese método no retornará hasta que el usuario complete adecuadamente todos los valores, mostrando por pantalla, en caso necesario, los errores al usuario. La declaración de ese método sería así:

```
public static Cliente nuevoCliente();
```

Incluir en la clase *Cliente* un método *infoClientes()* para exportar la información de una colección de clientes que se pasa como argumento. Este método recorre cada objeto de la colección y exporta a un fichero de caracteres de nombre *clientes.txt* la información de cada uno (uno por cada línea del fichero) en el siguiente formato:

```
nombre ( id ) fechaNac Tfno: telefono Dir: direccion
```

Incluir en la clase *Cliente* un método *notificacionesEn()* que, pasado un año como parámetro, devuelva un array de 12 posiciones (una para cada mes) con el nº total de notificaciones (por email o telefónicas) que haya recibido ese cliente.

Todos los métodos implementados deberán gestionar adecuadamente todas las excepciones que puedan originarse y manejarlas correctamente para evitar que el programa principal termine inesperadamente.

En un programa *main* principal definir las variables que se crea oportuno para utilizar todos los métodos implementados anteriormente.

Realizar la entrega de un proyecto Java desarrollado en Eclipse a través de la exportación del mismo en un fichero .zip nombrado con el nombre del alumno/a. Subirlo a la plataforma de aula virtual del módulo, en la url: <https://aulasvirtuales.educastur.es/mod/assign/view.php?id=382248>

Anexo

¿Qué formato tiene un NIF válido?

El NIF (Número de Identificación Fiscal) tiene una longitud total de nueve caracteres. El NIF contiene ocho dígitos numéricos (del cero al nueve) y un último dígito de control. El dígito de control del NIF es alguna de la mayoría de las letras del abecedario (de la "A" a la "Z") salvo alguna de las ellas, que nunca se utiliza.

¿Qué formato tiene un NIE válido?

El NIE (Número de Identidad de Extranjero) consta de nueve caracteres divididos en tres partes. El primer carácter del NIE debe ser una de últimas letras del abecedario ("X", "Y" o "Z"). En la segunda parte, des del segundo carácter del NIE hasta el penúltimo encontramos números del cero al nueve. Para terminar, el último dígito es el dígito de control que, como el NIF, puede ser alguna de la mayoría de las letras del abecedario.