5a.- un ***Stock*** debe ter os métodos *getStock(codigo) , addStock(codigo, num) throws VerboseException e setStock(codigo, num) throws VerboseException* permitindo almacenar e atopar rapidamente cantas unidades numéricas temos de un obxecto identificado por codigo. O Stock en ningún caso pode ser negativo, lanzándose unha VerboseException si se intenta. Se pide desenvolver **Stock**.

5b.- Crear a clase ArtigoStock, que almacena e permite atopar rapidamente cantas unidades temos no almacén de cada artigo empregando a estrutura de almacenamento dinámica en memoria máis axeitada.

6.- Crear a **clase *ClienteBuilder*** que implanta o ***patrón Builder*** para os obxectos Cliente. Un *Builder* e calquera clase que dispoña dun método ***build()*** que permita construír e retornar un obxecto. Debedes:

- Definir ***Builder*** empregando tipos xenéricos (Generics) para a súa correcta definición, xa que o método ***build()*** de Builder debe ser capaz de retornar un obxecto da clase concreta para a que se defina o Builder.
- Definir un Builder para Cliente (ClienteBuilder). ClienteBuilder é un Builder cos métodos setNome(String nome), setApelidos(String apelidos) .... etc. Estes métodos retornarán true si a información é correcta e false en outro caso. O método build comprobará que os atributos son correctos e si é o caso retornará un novo obxecto Cliente con esa información. Si algún atributo non é correcto, se xerará unha ***VerboseException*** que terá a información sobre os erros de validación a nivel INFO e unha explicación de por qué é erroneo a nivel de Verbosity ERR.
  - Se debe validar o DNI segundo as normas de numeración para os DNI e NIE de España,
  - Se debe validar a data de nacemento que ten que ser anterior ao 01/01/2006
  - Se debe validar o e-mail, que debe ser string@string
  - Se debe validar o código postal, que debe constar de 5 díxitos no que os dous primeiros van de 01 a 52
- ClienteBuilder terá un metodo **void notifyErrors() throws VerboseException** que lanzará unha VerboseException cos erros analizados ata o momento, ou non fará nada si non temos erros.
- ClienteBuilder debería ter un método void reset(); que limpa os erros rexistrados ata o momento

01 Araba/Álava	02 Albacete	03 Alicante	04 Almería	05 Ávila	06 Badajoz	07 Illes Balears	08 Barcelona	09 Burgos	10 Cáceres	11 Cádiz
12 Castellón	13 Ciudad Real	14 Córdoba	15 Coruña	16 Cuenca	17 Girona	18 Granada	19 Guadalajara	20 Gipuzkoa	21 Huelva	22 Huesca
23 Jaén	24 León	25 Lleida	26 La Rioja	27 Lugo	28 Madrid	29 Málaga	30 Murcia	31 Navarra	32 Ourense	33 Asturias
34 Palencia	35 Las Palmas	36 Pontevedra	37 Salamanca	38 S.C. Tenerife	39 Cantabria	40 Segovia	41 Sevilla	42 Soria	43 Tarragona	44 Teruel
45 Toledo	46 Valencia	47 Valladolid	48 Bizkaia	49 Zamora	50 Zaragoza	51 Ceuta	52 Melilla			

A saída do seguinte programa debería ser :

```
public static void main(String[] args) {
    VerboseException.setLevel(ERR);

    ClienteBuilder cb=new ClienteBuilder();
    cb.setDni("3sr073289X");
    cb.setNome("Juan");
    cb.setApelidos("González Perez");
    cb.setEmail("correo");
    cb.setTelefono("+34567444443434");
    try {
       Cliente c=cb.build();
       System.out.println("Creado o cliente "+c);
    } catch(VerboseException e) {
       System.out.println(e.getMessage());
    }
 }
```

```
DNI erróneo
        ERROR: O DNI debe constar de un número de 8 díxitos e unha letra
Email erróneo
        ERROR: O email debe ter a forma usuario@dominio
Teléfono erróneo
        ERROR: O número de teléfono debe constar de 9 díxitos
```