

A la nostra aplicació hem de crear instàncies de la classe `Email`, que són immutables. Aquesta classe conté alguns camps obligatoris i d'altres que són optatius (per simplificar les coses podem suposar que són tots de tipus `String`). Un "esquema" d'aquesta classe podria ser:

```
public final class Email {
    private final String from;
    private final List<String> to;
    private final String subject;
    private final String body;
    private final List<String> ccTo;

    public Email(
        String from, List<String> to, String subject, String body) {

        this(from, to, subject, body, new ArrayList<>());
    }

    public Email(
        String from, List<String> to, String subject, String body,
        List<String> ccTo) {

        this.from = from; this.to = to; this.subject = subject;
        this.body = body; this.ccTo = ccTo;
    }

    // more methods
}
```

A més, en aquest disseny inicial, els usuaris de la classe hauran de fer servir llistes pels paràmetres que són múltiples (fins i tot pel cas més simple quan només hi ha un destinatari).

Per això, es desitja una API més còmode per a crear el correus de manera que podem anar acumulant paràmetres i, en el moment d'anar a construir-la, comprovar que tenim tota la informació. Aquesta nova API seria

```
Email email = Email.builder()
    .from("juanmanuel.gimeno@udl.cat")
    .to("xyz@alumnes.udl.cat")
    .to("pqr@alumnes.udl.cat")
    .subject("Còpies a les activitats")
    .body("Com ja sabeu, la còpia a ...")
    .ccTo("capest.inf@eps.udl.cat")
    .make();
```

Les "regles" per aquesta nova forma de crear els Emails, seran:

- els mètodes poden cridar-se en qualsevol ordre
  - Excepte **builder**, que ha de ser el primer i **make** que ha de ser el darrer
- **from**, **subject** i **body** s'han de cridar exactament una vegada amb paràmetres no nuls ni buits
- **to** s'ha de cridar com a mínim una vegada, però es pot cridar varies
- **ccTo** és optatiu i pot cridar-se varies vegades

En cas de no respectar-se aquestes regles, es llençarà l'excepció no comprovada (*unchecked exception*) `EmailBuilderException`.

```
public class EmailBuilderException extends RuntimeException {  
    public EmailBuilderException(String message) {  
        super(message);  
    }  
}
```

Si es produeixen varis errors, només cal que l'excepció que es llença indiqui un d'ells.

El que es demana és:

1. (2 punts) Indiqueu quin patró cal aplicar i perquè serveix per a resoldre el problema que ens trobem.
2. (8 punts) Implemeteu la nova API per a crear instàncies d'Email.

### Simplificacions addicionals

- No cal que comproveu que pels atributs que es corresponen a adreces de correu, aquestes estiguin ben formades. Amb comprovar, com s'ha dit abans, que no són nul·les ni buides ja n'hi ha prou.

### Instruccions per l'activitat:

- Resoleu individualment i a ma l'exercici (recordeu, un dels objectius de les activitats és que arribeu "entrenats" als exàmens) que entregareu escanejat.
- Poseu en comú una solució, a partir de les solucions individuals i, si voleu, havent programat i testejat la solució
- El codi que corregiré/puntuaré serà la solució consensuada (que ja no estarà a mà) i que entregareu amb un PDF amb els comentaris que creieu oportuns (**no vull un projecte sinó un PDF**) sobre les decisions de disseny/implementació que heu pres, possibles alternatives, etc, etc.