

## TRACCIA A

### Esercizio 1

Tesla, leader mondiale dell'industria automobilistica elettrica, vuole integrare il proprio software gestionale con un modulo di gestione della produzione delle proprie batterie di ultimissima generazione.

In particolare, ha diviso la gestione in tanti piccoli moduli, e a ciascuno di essi corrisponde un unico microservizio.

Il modulo che si richiede di sviluppare si occupa della **produzione** delle batterie.

Il nome del progetto dovrà essere **teslabattery\_<iniziale\_cognome><iniziale\_nome>** (es. Mario Rossi creerà il progetto **teslabattery\_rm**), e lo stesso dovrà essere indicato come database nel file di properties.

Per costruire una batteria sono necessarie diverse **componenti** (es. il litio, gli acidi, ecc). Ovviamente, alcune sostanze sono nocive, e l'accesso ad esse deve essere indicato come *pericoloso*.

Una batteria viene costruita utilizzando una **formula**, un'entità che indica

- la **catena di montaggio** da utilizzare
- le componenti necessarie
- la quantità di componenti da utilizzare
- l'unità di misura della quantità (pezzi, millilitri, grammi, ecc)

Ogni **catena di montaggio** ha un nome e un tempo di completamento massimo della produzione, è composta da diversi **robot** (intesi come bracci meccanici o macchinari specifici) e funziona senza alcun intervento umano.

Ogni **robot** avrà quindi una mansione specifica (all'interno di un set predefinito di mansioni), un ordine posizionale nella catena di montaggio e altre caratteristiche proprie.

Quando viene avviato un nuovo **ciclo produttivo**, questo contiene

- un riferimento univoco alla catena di montaggio
- una data di inizio
- uno "status" che indica lo stato in cui si trova la produzione in un momento specifico (ad esempio, AVVIO\_PRODUZIONE, MONTAGGIO\_INVOLUCRO, COMPOSIZIONE\_STRATI, AGGIUNTA\_ACIDI, TEST\_QUALITA, COMPLETATO, FALLITO)
- una data di status che indica il momento in cui è avvenuto l'ultimo passaggio di stato (viene quindi aggiornato ad ogni cambio stato)
- una data di fine.

Ad ogni **ciclo produttivo** viene infine assegnato un singolo **supervisor**, che si assicura che il ciclo non si blocchi e che i parametri di qualità rilevati da sensori e tester restino sempre nei range di eccellenza.

Per questo microservizio, Tesla pretende inoltre che

- La cancellazione su ogni controller venga disabilitata, e che solo gli utenti con ruolo “supervisor” possano accedere agli endpoint
- Sia abilitato Swagger

Identifica inoltre diverse fasi, ad ognuna dei quali associa un peso economico

- FASE 1 (600000 €) – Back End
  - Tutti i model, i dto, le conversioni e i repository delle entità devono essere completati
  - Security attiva
  - Business logic (dal controller al repository)
    - **Obiettivo minimo:** devono essere funzionanti i controller di componenti, formula, catena di montaggio. La base dati può essere inizializzata in automatico dal microservizio.
    - **Obiettivo massimo:** devono essere funzionanti, oltre ai precedenti, anche robot, catena di montaggio / robot e ciclo produttivo. La base dati deve essere inizializzata tramite appositi script di generazione tabelle e relazioni.
    - Ogni controller deve fornire i metodi CRUD di
      - get
      - getAll
      - post
      - put
      - delete
  - Test unitari
    - **Obiettivo minimo:** di ogni metodo CRUD di almeno un controller tra catena di montaggio e ciclo produttivo
    - **Obiettivo massimo:** di ogni metodo CRUD di ogni controller
- FASE 2 (300000 €) – Statistiche BE
  - Si creino dei metodi (dal repository al controller) che forniscano parametri sulla qualità dei cicli produttivi. In particolare Tesla vuole sapere
    - **Obiettivo minimo**
      - Numero di cicli COMPLETATI ogni mese
      - Numero di cicli FALLITI ogni mese
      - Numero di cicli TOTALI ogni mese
    - **Obiettivo massimo**
      - Percentuale di successo dei cicli produttivi (COMPLETATI su TOTALI)
      - Numero di cicli produttivi falliti, raggruppati per stato del ciclo
      - Numero di componenti sprecate nel mese (intese come numero di componenti associate a cicli falliti)

- FASE 3 (100000 €) – Front End – **DA IMPLEMENTARE SOLO SE COMPLETATE LE PRIME 2 FASI CON OBIETTIVO MASSIMO**
  - Si crei una pagina web che consenta la visualizzazione dei vari cicli produttivi, l'avvio di un nuovo ciclo e la cancellazione.
    - **Obiettivo minimo:** CRUD funzionante
    - **Obiettivo massimo:** visualizzazione delle statistiche
  - Per la scrittura del FE si utilizzino le librerie fornite da Tesla
    - Ajax-core.js
    - Base-service.js
  - Possono inoltre essere utilizzate le librerie
    - Qualsiasi libreria js o framework css utile ai supervisor per avere un'esperienza utente ottimale
    - Al bisogno, qualsiasi libreria di chart