

# Prova finale di Ingegneria del software

Prof. Marco Brambilla, Como

#swEngComo

## CONSEGNE

### 1) Requisiti: Fattibilità + DataDictionary + Goal Diagram

- Fattibilità (1 pagina)
- Progetto i\* con OpenOme
- Documento di progetto, con: testo del progetto (da questo documento), fattibilità, sintesi goal e data dictionary (non c'è un formato fisso, è a vostra discrezione)

### 2) Design: UML

- Almeno 2 activity, 2 sequence e 2 state diagrams
- Almeno 1 diagramma per ogni restante tipo di diagramma
- File di progetto di tool UML
- Documento PDF riassuntivo dei diagrammi fatti, commentati

### 3) Implementazione: Java

- Progetto Java completo e funzionante
- Casi di test Junit
- Documento PDF riassuntivo

### Consegna Completa

- Progetto e documento completo e funzionante

## GITLAB

Le consegne sono da effettuare UNICAMENTE tramite Gitlab.

L'uso di Gitlab è richiesto per TUTTA l'attività di sviluppo, non solo per le fasi di consegna.

- Ogni membro del gruppo deve creare un account [GitLab](#)
- Ogni gruppo condivide gli account gitLab dei membri sul foglio di registrazione gruppi
- In risposta riceverete l'accesso a un repository GitLab con la seguente struttura:
  - Requisiti

- Design
- Java

Per le consegne:

- Verificare di aver caricato tutto il codice prima della deadline
- Creare un tag sul repository (nome del tag: “**requisiti**”, “**design**”, “**java**”, “**final**”)
- Non modificare il tag dopo la deadline

Farà fede la data di creazione del tag.

## TEMA

Attraverso un sistema di soglie di allarme a più livelli, la Protezione Civile dà l’allerta per fenomeni di maltempo (bufere, piogge violente, nevicate intense...) e altri rischi (terremoti, ecc.).

Le allerte sono riferite ai codici di avviamento postale (CAP). I dati sono mantenuti costantemente aggiornati sulla base dei calcoli di vari sottosistemi software esterno che elaborano le previsioni per i diversi fenomeni, nell’arco delle successive 24 ore, con granularità 4 ore.

Il sistema registra tutte le previsioni dei fenomeni/eventi e anche l’effettivo accadimento degli stessi. Ogni evento è descritto da un ID, un testo, un CAP e una sorgente. Ogni evento ha inoltre un tipo. Il sistema tiene traccia di tutti gli eventi ricevuti in ordine di tempo. Gli utenti possono cercare gli eventi per istante temporale, per tipo o per CAP.

Il sistema tiene anche traccia degli eventi correnti. Per evento corrente si intende un evento in corso per ogni CAP nel giorno corrente. I restanti eventi passano in archivio storico.

Se per un certo CAP in una certa data sono presenti più eventi, si considera un ordinamento per livello di gravità più elevato.

Esiste un elenco di allarmi in evidenza, che sono quelli di massima gravità per fascia oraria e per CAP. Se in base a previsioni successive un’allerta rientra, questa viene rimossa dalle allerte in evidenza, e si pone in evidenza l’eventuale allerta di livello immediatamente inferiore.

Specificare, progettare e implementare il sistema distribuito necessario, coprendo: sistema centrale, applicazione di previsione, applicazione mobile degli utenti che possono ricevere notifiche degli allarmi.

Definire esplicitamente tutti i formati dei dati scambiati e le modalità di scambio (protocollo).

E’ possibile raffinare i requisiti ed aggiungere ipotesi e assunzioni sul contesto, sensate e in linea con quanto indicato nei requisiti. Tali estensioni devono essere esplicitamente riportate nella documentazione di progetto (sezione specifica requisiti).

# VALUTAZIONE

Il voto viene assegnato al progetto solo in sede di consegna finale. Il voto terrà conto anche delle attività svolte, della collaborazione di tutti i membri, delle registrazioni dell'avanzamento di progetto sulla piattaforma GitLab, oltre al livello di qualità generale delle consegne.