

Linee guida per il progetto di laboratorio del corso di Tecniche e architetture avanzate per lo sviluppo del software

AA 2019-2020

Argomento del progetto

Un sistema che possa essere realizzato mediante architettura **SOA e Microservizi**, utilizzando gli standard dei Web Services, nella quale la parte principale dell'applicazione dovrà essere realizzata con architettura **3 Tiers** con interfaccia utente, business logic e Data base.

Non vanno bene: programmi di grafica, giochi, simulazioni, ecc.

Esempi di progetti: negozi virtuali, sistemi di gestione biblioteca, banca, agenzie di viaggio, prenotazioni mediche, car-sharing, TO-BYKE, AIRb&b ecc

Come e con che strumenti

Metodologia

Il progetto deve essere realizzato cercando di seguire il più possibile la metodologia Agile dell'**Xtreme programming** (vista a lezione). Quindi le fasi del progetto **non** devono essere separate e sequenziali:

- Descrizione informale delle funzionalità del sistema (requisiti), con **scenari di uso, user stories**, del sistema da parte dei potenziali utenti (breve documento condiviso online)
- Creazione di mockups con uno strumento a scelta tra quelli disponibili, tipo:
<https://moqups.com/>
<https://balsamiq.com/products/mockups/>
<https://gomockingbird.com/home>
- Dalla descrizione degli scenari devono essere ricavati:
 - insiemi di classi e loro responsabilità (API), con il metodo delle **CRC cards** (su carta), tenendo traccia su carta di ipotesi successive, con traccia di esecuzioni simboliche per verificare la completezza dell'analisi, cioè la presenza delle classi e responsabilità necessarie alla realizzazione del sistema
 - Use-cases in UML (solo un sottoinsieme di quelli possibili) con strumento di UML tipo GenMyModel o altri strumenti gratuiti.
 - Diagramma delle classi in UML (obbligatorio) e di attività'

- Un paio di diagrammi di collaborazione o sequenza (a scelta) corrispondenti ad un paio di use cases più complessi (obbligatori). NB: questi diagrammi corrispondono alle esecuzioni simboliche fatte con le CRC cards, quindi vanno fatti **dopo** la fase delle CRC cards.
- Se **utili** diagrammi di stato (facoltativi)

Implementazione/design

- L'applicazione deve essere sviluppata per quanto riguarda la parte backend con:
 - Spring
 - se si e' piu' di 3 anche con una parte in JEE
- per la parte Web:
 - JSP oppure HTML5 e JavaScript, Angular o altri framework a scelta, scambiando JSON con il backend
- Identificazione di uno o piu' servizi esterni che possano essere implementati come **Web Services** REST e SOAP e che comunichino con una componente del sistema principale
- identificazione di un servizio del sistema accessibile da una **Android app** e quindi sviluppo della applicazione (semplice) che contatta il server per ricevere informazioni generate da quel servizio, fornito da servizi REST.

Eventualmente ritorno alla fase di analisi per modificare le classi e loro responsabilità

Project Managment

- Il progetto deve essere gestito a livello di Project Mng, con un tool tipo uno dei seguenti:
 - <https://trello.com/b/l7TjplA/trello-tutorial>
 - <https://app.teamgantt.com/welcome/step-1>
 - <https://taiga.io> [con github]

aggiornando (CONTROLLARE) il project plan che definite all'inizio del progetto.

Ambienti, tecnologie da utilizzare

- Implementazione di un **sottoinsieme** delle funzionalità del sistema analizzato e progettato, in Java e utilizzando **Spring**
- IDE: **Netbeans** o IntelliJ o **Eclipse** e AndroidStudio (obbligatorio). Quindi deve essere implementato e funzionante un prototipo "verticale" del sistema, cioè devono essere presenti tutti e tre i livelli: interfaccia, EJB e DB.
- Se possibile, deploy della applicazione su OpenShift che comunica con l'applicazione Android (mediante JSON)

- Il software sviluppato **deve** essere inserito nel sistema controllo versione disponibile all'interno di Netbeans e/o Eclipse o IntelliJ, per esempio **GIT (o Subversion)** e memorizzato in GitHub o simili.
- integrare un servizio esterno tipo Facebook, Google Apps, utilizzando REST WS
- il progetto dovrebbe essere sviluppato seguendo le linee guida dell'**Xtreme programming** e inoltre cercando di costruire un vero e proprio gruppo di progetto, come descritto in Managing for Excellence.
- Un paio delle funzionalita' devono essere disponibili su applicazione Android (nativa) che comunica con JSON con la parte server

Opzionale (per arrivare al 30L nel progetto):

- Contattare un altro gruppo e costruire un sistema integrato, loosely coupled, utilizzando i Web Services

Suggerimenti:

- Il sistema deve essere complesso per rendere meglio la fase di analisi OO e di Design, la parte implementata dovrà coprire solo un sottoinsieme di funzionalità di tutto il sistema !
- Contattare un altro gruppo e costruire un sistema integrato, loosely, utilizzando JMS o i Web Services
- Alla presentazione del progetto portate:
 - documento di descrizione del sistema
 - i fogli sui quali avete scritto durante le fasi di design e l'analisi, così riuscite a descrivere l'evoluzione di queste fasi
 - le stampe dei diagrammi UML
 - demo **funzionante** del vostro sistema

