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

AA 2020-2021

Argomento del progetto

Un sistema che possa essere realizzato mediante architettura **SOA/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, analisi dati, AI, ecc.
Esempi di progetti adatti: 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)
  - 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

- Implementazione di un **sottoinsieme** delle funzionalità del sistema analizzato
- L'applicazione deve essere sviluppata per quanto riguarda la parte backend con:
  - Spring
  - il backend deve essere inserito in container **Docker**
  - se si e' in 3 o piu' persone in gruppo dovete utilizzare **Kubernetes**

- per la parte Web:
  - Angular o altri framework a scelta come VueJs, React, JSP oppure HTML5 e JavaScript, scambiando JSON con il backend

- 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.
- integrare un servizio esterno tipo Facebook, Google Apps, per realizzare la registrazione
- Un paio delle funzionalità' devono essere disponibili su **applicazione Android** (nativa) che comunica con JSON con la parte server

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

Opzionale (per arrivare al 30L nel progetto): NEW

- Contattare un altro gruppo e costruire un sistema integrato, loosely coupled, utilizzando i Web Services
  oppure
- implementare un Design Pattern per Microservizi, come Saga.
- Implementare l'architettura a Microservizi, mista, quindi magari senza consistenza dati tra i vari DB (quindi non necessariamente seguendo un design pattern come Saga), ma il backend deve essere inserito in Docker e con utilizzo di Kubernetes.

(la seconda o la terza sono le opzioni consigliate tra le tre alternative)

Ambienti da utilizzare

- IDE: **IntelliJ** o Netbeans o Eclipse e AndroidStudio (obbligatorio). Quindi deve essere implementato e funzionante un prototipo "verticale" del sistema
- Il software sviluppato d**eve** essere inserito nel sistema controllo versione, per esempio **GIT (o Subversion)** e memorizzato in GitHub o simili**.**

Project Management

Il progetto deve essere gestito a livello di Project Mng, con un tool tipo uno dei seguenti:

- https://trello.com/b/I7TjiplA/trello-tutorial
- https://app.teamgantt.com/welcome/step-1
- https://app.clickup.com/

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

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 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 PDF dei diagrammi UML (POCHI)
  - demo **funzionante** del vostro sistema

Traccia di architettura da cui partire per Spike Architetturali per l vostro progetto: