

**2017- Sessione estiva**  
**Sezione A classe 8 - Ingegnere senior**  
**Giugno 2017 Tracce sorteggiate**

**Prima prova scritta - Settore dell'informazione**

- Il candidato illustri le tecniche e le metodologie di progettazione ad alto livello di astrazione di sistemi digitali integrati in tecnologia CMOS VLSI con riferimento ad un linguaggio di descrizione hardware ad alto livello.
- Con riferimento alla programmazione di applicazioni in Internet, si illustri il concetto di architettura client/server. Si descrivano inoltre le caratteristiche principali dell'applicazione di posta elettronica e del protocollo di comunicazione SMTP.
- Con riferimento allo sviluppo di software nel paradigma ad oggetti, si illustri il concetto di design pattern. Si descrivano inoltre due design pattern tra quelli più comunemente utilizzati nell'ambito dell'ingegneria del software.

**Settore dell'informazione**

- Il candidato scriva una relazione sintetica che illustri possibili forme di presentazione grafica (diagrammi) di dati gerarchici, ossia dati modellabili come alberi (grafi senza cicli). La relazione si deve soffermare in modo particolare sulle proprietà dei digrammi considerati, sui possibili algoritmi di calcolo di tali diagrammi e sulla relativa complessità computazionale.
- Il candidato imposti una relazione di progetto di un servizio on-line per la gestione di un magazzino di prodotti di abbigliamento con più sedi sul territorio nazionale. Il candidato si soffermi anche sull'analisi di alcune attività trasversali, quali la stima dei costi e dei tempi necessari per la realizzazione del software.
- Il candidato imposti una relazione progettuale generale nella quale siano evidenziati i criteri di progetto per il controllo con metodi analitici di sistemi lineari stabili asintoticamente con ritardo finito mediante regolatori con azione Proporzionale+Integrale+Derivativa e azione feedforward.

**Informazione LM32**

**TEMA 1**

La *SmartCityICT s.r.l.* è un'azienda che offre servizi ICT per le Smart City. Tra i progetti più rilevanti che la vedono coinvolta vi è lo sviluppo della *QueueManagementApplication (QMA)*: un'applicazione per la gestione delle code di attesa, che dovrà essere distribuita secondo il paradigma *SaaS (Software as a Service)*.

Dopo una prima analisi sommaria, si è stabilito che l'applicazione QMA dovrà soddisfare i seguenti macro-requisiti:

- presenza di tre profili utente: *administrator, provider, consumer*;
  - *adminstrator*: utente amministratore del sistema con massimi privilegi;

- *provider*: qualunque organizzazione/ente che fornisce un servizio S al pubblico e che desidera utilizzare QMA per la gestione della coda di attesa al servizio S;
- *consumer*: un qualsiasi cliente, di un qualche provider P accreditato in QMA, che intende usufruire dell'applicazione QMA per avere informazioni tempestive sulla lunghezza della coda di attesa per il servizio offerto da P;
- i dati sulla lunghezza della coda nei singoli locali dei provider, devono essere prodotti in modo automatico da un apposito hardware (sensore), connesso alla rete Internet, inserito all'interno del distributore (box) di ticket cartacei;
- la comunicazione con il sistema informativo dell'applicazione QMA deve consistere in semplici richieste *GET* del protocollo *HTTP(S)*;
- ogni sensore deve inviare le seguenti informazioni: codice del servizio, orario di rilevamento, numero totale di persone in coda di attesa, elenco dei prossimi N codici-ticket di clienti in attesa;
- ogni sensore deve effettuare un rilevamento ogni 5 minuti;
- i dati sulle code devono essere disponibili per un periodo di almeno 1 anno;
- l'accesso ai dati sulle code di attesa dovrà avvenire tramite un portale Web e/o Web app, in particolare tutti gli utenti del portale devono poter monitorare facilmente la lunghezza della coda per il servizio di loro interesse.

Si richiede di:

- definire un elenco dettagliato di requisiti per l'applicazione QMA aderenti ai macro-requisiti sopra menzionati;
- fornire un'architettura di alto livello della *QueueManagementApplication* che garantisca una buona scalabilità del servizio;
- indicare una strategia di gestione della persistenza dei dati che garantisca anch'essa il requisito di scalabilità, illustrando la struttura della(e) basi di dati impiegata(e);
- definire il formato XML e/o JSON dei messaggi inviati dai sensori al sistema informativo dell'applicazione QMA;
- fornire un elenco dei principali casi d'uso per l'utente provider e per l'utente consumer;
- fornire i diagrammi delle classi UML per il livello di accesso ai dati e per il livello della logica di business di una possibile implementazione del portale Web in architettura 3-tier;
- specificare il tipo di tecnologia utilizzata;
- definire un insieme di test tesi a verificare l'efficacia e l'efficienza dell'applicazione QMA.

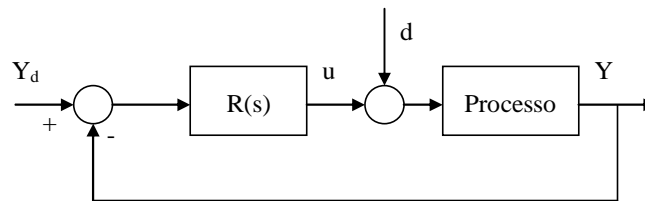
## TEMA 2

Ad uno studio di Ingegneria viene commissionato il progetto del sistema di controllo per un processo descritto dal seguente modello linearizzato:

$$Y(s) = \frac{k_m(1+T_3s)}{(1+T_1s)(1+T_2s)} e^{-sL} U(s)$$

I parametri del modello sono:

$$T_1 = 7.0 s; T_2 = 0.5 s; T_3 = 0.6 s; L = 2 s; k_m = 2.0$$



Progettare un sistema di controllo a controreazione che garantisca le seguenti specifiche:

- astatismo rispetto a disturbi a gradino;
- errore nullo per riferimento a gradino;
- tempo di risposta 7 s;
- sovraelongazione minore del 5%.

Valutare la possibilità di inserire un regolatore feedforward per migliorare le prestazioni del sistema di controllo.

### TEMA 3

Ad uno studio di Ingegneria viene commissionato il sistema di gestione di un parcheggio a pagamento.

Il parcheggio ha due ingressi e due uscite, ciascuno gestito da sbarre.

Gli ingressi hanno ciascuno una sbarra; l'uscita ne ha una coppia per impedire che due veicoli possano uscire con lo stesso biglietto.

Il numero di posti liberi del parcheggio è memorizzato in una variabile, il cui valore è mostrato da un display.

Alla pressione di un pulsante la sbarra di ingresso relativa si alza e si richiude dopo un certo tempo; la sbarra non si richiude se c'è un ostacolo; se non ci sono posti disponibili la sbarra non si alza. All'ingresso del veicolo viene decrementato il valore dei numeri di posti disponibili.

L'uscita funziona nel seguente modo:

all'inserimento del biglietto la sbarra più interna si alza e rimane aperta fino a che il veicolo è entrato nello spazio compreso fra la sbarra interna e quella esterna; questa condizione è segnalata da un sensore;

quando il veicolo è entrato nello spazio intermedio, dopo che la sbarra interna si è chiusa, si apre la sbarra esterna, che rimane aperta fino a quando il veicolo non è transitato. In ogni caso, le sbarre non si abbassano se è presente un ostacolo, segnalato da un sensore. La sbarra interna non si apre se la sbarra di uscita è aperta.

All'uscita del veicolo viene incrementato il numero di posti disponibili.

In caso di richieste di ingresso o di uscita simultanee devono essere conteggiati correttamente i posti liberi.

Si richiede di:

- descrivere il funzionamento del sistema mediante una macchina a stati finiti;
- implementare la logica di controllo in SFC e in LADDER.