

SCRITTO IN ITINERE DI
IUM, IUM+TWEB, Ist. di Tecnologie Web, SWAM, Laboratorio di
Servizi Web, Servizi Web
27 maggio 2019 - PROFF. ARDISSONO-PATTI

Tempo: 1:50 hr. IUM+TWEB; 50 min. per IUM; 1 hr per SWAM, Ist. Tec. Web, Lab. Servizi Web e Servizi Web
Non sono ammessi libri di testo e appunti. Utilizzare per le risposte solo i fogli che vi verranno consegnati.

Si prega di SVILUPPARE GLI ESERCIZI DELLA PARTE 1 E DELLA PARTE 2 IN FOGLI PROTOCOLLO SEPARATI per permettere la correzione in parallelo.

1-IUM PATTI

1. (6 punti) **Modelli di interazione:** Spiega il concetto di *affordance*, e fornisci un esempio di oggetto dotato di *affordance* e un esempio di oggetto senza *affordance*, spiegandone il perché. Discuti quindi l'importanza di questo concetto per affrontare le difficoltà che si presentano in alcune fasi dell'interazione fra utente e sistema schematizzata secondo il modello di Norman.
2. (6 punti) **Usabilità e accessibilità:** Definisci le nozioni di *accessibilità* e *usabilità* e mettile a confronto discutendone le relazioni nella loro applicazione al design di sistemi interattivi.
3. (3 punti) **Conoscere l'utente:** Descrivere quali indicazioni fornisce lo studio dell'attenzione al progettista di sistemi interattivi in relazione agli aspetti di usabilità.

2-TWEB ARDISSONO

1. **XML (6 punti):** si definisca un XMLSchema che specifichi documenti XML che descrivono il menu di un ristorante. La radice dei documenti sia un elemento menu, caratterizzato dai seguenti sotto-tag:
 - primi <!-- elenco non vuoto di elementi piatto, strutturati come segue:
 - nome <!-- elemento che indica il nome del piatto – stringa di caratteri -->
 - ingredienti <!-- elenco non vuoto di ingredienti – ogni ingrediente e' rappresentato da una stringa di caratteri -->
 - vegan <!-- elemento che specifica se il piatto e' vegan o meno - booleano - -->

-->

- secondi <!-- elenco non vuoto di elementi piatto (si veda la definizione di piatto) -->
- pagamento <!-- modalita' di pagamento: elemento opzionale che prende valori in {contanti, POS} -->

Disegnare il DOM di un tipico documento XML valido rispetto alla grammatica specificata.

2. **Applicazione Web in ambiente java – parte teorica (4 punti):** Si descriva in dettaglio il pattern architetturale MVC per il web, eventualmente con l'aiuto di un disegno, e se ne discutano i vantaggi e gli svantaggi.

3. **Applicazione Web in ambiente java – esercizio (5 punti):** Scrivere il codice di una Servlet Java che gestisce richieste GET
`http://localhost:8080/Info?operazione=o` come segue:

1. se l'operazione richiesta è "dati", la Servlet invia il controllo alla pagina "showData.jsp", che gestisce la richiesta.
2. se l'operazione richiesta è "exit", la Servlet scrive un saluto all'utente prendendo il suo nome da sessione utente; poi invalida la sessione utente e infine invia il controllo alla pagina "quit.html".
3. per ogni altro tipo di richiesta, la Servlet invia il controllo alla pagina "error.jsp", specificando che il messaggio da visualizzare su browser è "Richiesta non valida".

NB: Si implementino la servlet e error.jsp.
NON si implementino showData.jsp né quit.html.