

PROGETTAZIONE WEB

PROGETTO DIDATTICO

v. 1.1

La realizzazione del progetto, da svolgersi singolarmente, ha lo scopo di applicare le conoscenze acquisite durante il corso allo sviluppo di servizi che, seppur nella loro limitata complessità, implementino delle funzionalità utili in qualche contesto del mondo reale. Le funzionalità devono essere complete. Per esempio, una procedura che controlla il formato della data senza tenere conto degli anni bisestili, non è completa e quindi non può essere utilizzata nel mondo reale. Per poter verificare che un servizio sia completo, il servizio da realizzare deve essere logicamente ben definito: solo in questo caso si può fare un test funzionale, ossia provare se funziona correttamente. Per esempio, se si realizza un gioco, devono essere chiare le regole. Se si realizza un modulo da compilare, deve essere chiaro come compilarlo correttamente. A questo fine è buona norma partire da contesti reali dei quali si ha esperienza diretta.

Per verificare la completezza e la correttezza del servizio realizzato, si richiede di inserire nella pagina *index.html* (o in una pagina ad essa collegata) un piccolo manuale utente (20-25 righe) che spieghi senza utilizzare termini tecnici come usare il servizio. Nel suo insieme, il servizio realizzato deve essere composto da **funzionalità coese**¹, **non ridondanti**², **modulari e leggibili**³. Il servizio può non essere originale (per esempio, il gioco della dama) ma il codice che lo realizza deve essere inedito. Eventuali frammenti minori di codice presi dal web devono essere studiati e ristrutturati secondo i criteri di programmazione e di validazione del corso. Nel caso si adoperi del codice preso dal web senza alcuna modifica, occorre inserire nel codice il sito di provenienza ed evidenziarlo bene tramite dei commenti, dal momento che non sarà preso in considerazione per la valutazione.

¹ Due o più funzionalità sono coese se appartengono al medesimo servizio e possibilmente interagiscono tra loro, come ad esempio le diverse funzioni di un gioco. Una pagina web che include una collezione di tante piccole procedure logicamente indipendenti (ad es. l'ora corrente, le previsioni del tempo, il gioco del tris ed il calcolo del codice fiscale) non presenta funzionalità coese. È buona norma fissare alcuni servizi di una sola tipologia di progetto, e svilupparli in profondità, piuttosto che fare un progetto con “un po’ di tutto, ma abbozzato”.

² Due o più funzionalità sono ridondanti se contengono parti di codice di controllo o di elaborazione/presentazione dati che sono molto simili. È buona norma cercare di accorpare tante procedure simili in un'unica procedura parametrica e condivisa. Ad esempio, in un progetto di verifica dei campi inseriti in form, è buona norma fare procedure condivise per la presenza/assenza di contenuto nei campi, ed anche per la verifica del contenuto specifico. Anche sui contenuti può esserci ridondanza. Ad esempio, è inutile inserire una legge con molti articoli con la medesima struttura che si ripete. Oppure molte immagini in un elenco. Ridurre i contenuti a pochi esempi rappresentativi, specialmente se gestiti con tecnologia lato client. Se gli elementi da inserire devono apparire tutti per questioni di integrità del servizio (ad esempio un menu HTML con tutti i comuni d'Italia), è buona norma suddividere tali elementi in categorie e consentirne il caricamento dinamico (ad esempio, un menù che consente di selezionare la provincia, e poi un secondo menù che si genera dinamicamente con 2-3 comuni di quella provincia, inserendo solo esempi per 2-3 province).

³ Nella fattispecie un servizio è modulare se, a livello di codice, è implementato a funzioni o ad oggetti, e ciascuna funzione (o metodo) è abbastanza corta da potersi leggere grosso modo in una sola videata. Vi sono inoltre un numero minimo di variabili globali. Il codice HTML, JS, CSS è separato in diversi file. I file sono suddivisi in cartelle per tipologia (*img*, *css*, *js*, *php*, *mid*,...), il codice è inserito in file separati rispetto ai dati, vi sono vari file *js*, *css*, *php* ciascuno con funzioni logicamente separate ed indipendenti. La leggibilità del codice è, accanto alla modularità, una caratteristica cruciale nei linguaggi di alto livello per la riusabilità del codice. La leggibilità nella fattispecie riguarda il fatto che le variabili e gli oggetti abbiano come nomi dei sostantivi (eventualmente con aggettivi, quindi non nomi incomprensibili come “a”, “x”), le funzioni siano nominate con dei verbi (o frasi verbali), ed il codice sia indentato.

Le tecnologie⁴ che si possono adoperare nella realizzazione del servizio sono HTML, CSS, JavaScript con DOM Level 3, PHP, MySQL. Il progetto deve potersi eseguire correttamente con i due browser Mozilla Firefox e Google Chrome. Tutto il codice deve poter essere eseguito usando le versioni specifiche dei prodotti software contenuti nel pacchetto dell'insegnamento.

Ogni progetto deve avere una componente lato client e una componente lato server. Per progetti maggiormente orientati al lato client, per esempio giochi interattivi, la componente lato server si può limitare alla gestione della registrazione degli utenti e della classifica. Per progetti maggiormente orientati al lato server, per esempio bacheche elettroniche, sistemi di gestione delle discussioni online, sistemi di noleggio e di e-commerce, la componente lato client deve gestire in modo amichevole e interattivo l'interfaccia con l'utente.

Si raccomanda una distribuzione dei componenti che sia portabile⁵, ed organizzata nel seguente modo. Tutti i file dovranno essere inclusi in una cartella principale nominata con *matricola cognome* dello studente; a partire dalla pagina principale, nominata come *index.html* (o *index.php*) dovrà essere possibile la navigazione in tutte le pagine del progetto inclusa la documentazione (esclusivamente in formato html) che dovrà spiegare in modo semplice e non tecnico i servizi sviluppati e le modalità di uso.

Il codice HTML 5 e CSS 3 (sia statico che dinamicamente generato, estratto tramite apposita bookmarklet) verrà validato durante la discussione del progetto.

Attenzione: è necessario fare in modo che non ci siano warning generati dai validatori HTML e CSS o, in alternativa, giustificare con un commento la loro presenza. Gli strumenti di validazione sono disponibili su <http://validator.w3.org> e <http://jigsaw.w3.org/css-validator>. Non usare proprietà non standardizzate, tranne in casi concordati.

Per eventuali componenti eseguibili con plug-in aggiuntivi occorrerà riferire l'URL da cui possono essere scaricati tali applicativi; la parte relativa allo sviluppo di componenti lato client al di fuori di HTML+CSS+JavaScript non verrà comunque valutata e pertanto non dovrà essere inserita nella documentazione o nei commenti, anche se potrà essere inclusa nel progetto. A titolo indicativo, le dimensioni di un progetto sono adeguate⁶ se comparabili alle dimensioni dei progetti mostrati durante i laboratori.

⁴ I progetti sono discussi in una qualsiasi delle aule didattiche o laboratori di facoltà adibiti a sede di esame, dove non si dispone dei diritti di amministrazione necessari per installare nuovi pacchetti applicativi o creare nuovi utenti in ambiente Linux o Windows. Le prove dei progetti avvengono usando l'utente *studenti* nella piattaforma Windows, con i pacchetti di installazione scaricabili dal sito dell'insegnamento. Anche se si dispone di un portatile, il progetto deve essere eseguito usando la macchina presente nell'aula e con il pacchetto software dell'insegnamento. Eventuali esigenze specifiche possono essere concordate preliminarmente.

⁵ Un modulo è portabile se è in grado di funzionare su diverse piattaforme e partendo da cartelle diverse del file system. A tale scopo, non deve esserci alcun percorso assoluto nei riferimenti ai componenti all'interno di tutte le pagine (quindi nessun riferimento a percorsi al di fuori della cartella principale) a meno di ancorare ad indirizzi web per riferire le fonti adoperate. Inoltre, gli applicativi (quali browser, Apache, PHP, MySQL) devono essere considerati con i percorsi e le impostazioni di default. Eseguire un test funzionale del proprio progetto adoperando il pacchetto dell'insegnamento. Eventuali impostazioni specifiche dovranno essere concordate. Occorre fornire lo script SQL per generare il db, le tabelle ed i dati di partenza.

⁶ Considerando codice e dati non ridondanti. Il codice deve essere indentato in modo da evidenziarne la struttura. Per dubbi sulle dimensioni venire a ricevimento.

Modalità di consegna e di valutazione dei progetti

Per consegnare il progetto in versione definitiva comprimere⁷ la cartella principale e caricare il file zip usando il seguente form:

<https://forms.gle/8wRD1ri8iW1eie8i8>

La consegna del progetto deve avvenire entro il terzo giorno solare che precede la data dell'esame (es. esame 17, consegna nel giorno 14). **I progetti caricati dopo la scadenza non saranno presi in considerazione.** Il progetto è valutato⁸ il giorno dell'esame, mediante un colloquio orale nel quale sono considerati i seguenti aspetti:

1. Servizio logicamente definito
2. Test funzionale sui due browser
3. Validazione HTML/CSS
4. Assenza di errori JavaScript/PHP
5. Dimensione del progetto
6. Coesione funzionalità
7. Codice non ridondante
8. Codice inedito
9. Leggibilità e modularità
10. Utilità
11. Portabilità

⁷ In formato “.zip”, ottenendo per esempio 12345_rossi.zip. Eliminare ogni frammento di codice o file non adoperato. A seguito della consegna, si riceverà entro 48 ore una e-mail di conferma della ricezione. Un progetto consegnato deve essere discusso nel primo appello disponibile, altrimenti viene rimosso dall'archivio delle consegne.

⁸ Il candidato dovrà portare una copia dei file consegnati, preparandosi all'occorrenza a caricarli personalmente sulla macchina disposta a tale scopo.