



**PUBBLICAZIONE AI SENSI DELL'ART. 19 DEL D.LGS. N. 33 DEL 14 MARZO 2013,
MODIFICATO DALL'ART. 18 DEL D.LGS N. 97 DEL 25 MAGGIO 2016**

BANDO N. 367.410 CTER AREA PA

CONCORSO PUBBLICO PER TITOLI ED ESAMI PER L'ASSUNZIONE CON CONTRATTO DI LAVORO A TEMPO PIENO E INDETERMINATO DI DUE UNITÀ DI PERSONALE PROFILO COLLABORATORE TECNICO ENTI DI RICERCA, VI LIVELLO PROFESSIONALE PRESSO L'AREA DELLA RICERCA DI PALERMO DEL CONSIGLIO NAZIONALE DELLE RICERCHE

TRACCE PROVA PRATICA DEL 20.06.2023

TRACCIA ESTRATTA

Traccia n. 3

Si vuole sviluppare una web application per la gestione di un magazzino.

1. Si sviluppi lo schema logico tramite linguaggio SQL del database dei prodotti che preveda almeno due entità: Prodotto che conserva le informazioni legate a marca e modello del prodotto e Magazzino che conserva l'anagrafica del singolo magazzino, in una relazione denominata in `In_Stock` di tipo N:M che conservi al suo interno le quantità in giacenza per ogni singolo prodotto in ogni singolo magazzino.
2. Assumendo di avere a disposizione una tabella Prodotto, almeno con i campi id, marca, modello, Magazzino con i campi id, denominazione e indirizzo, e una tabella di relazione `In_Stock` con il campo quantità, sviluppare il codice lato server che gestisca una richiesta web alla URL `https://<nome server>/giacenza` contenente il nome di un magazzino. Il codice dovrà gestire il caso di magazzino non trovato inviando in risposta il rendering di una pagina HTML contenente un apposito messaggio ovvero dovrà inviare una pagina contenente una tabella con le giacenze dei singoli prodotti presenti nel magazzino.
3. Si realizzi una pagina HTML costituita da un form contenente un menu dropdown che verrà popolato all'atto del caricamento della pagina e che conterrà una lista di prodotti richiesti con apposita richiesta web alla URL `https://<nome server>/prodotti` da cui si assume che si ottenga un file `prodotti.json` la cui struttura è:

```
{"prodotti":      «identificativo», "marca": «nome marca», "modello": «mode110>}, ...}
```



La pagina conterrà il codice Javascript per invocare il server alla URL <https://<nome server>/giacenza> con una richiesta web contenente una stringa con il nome del magazzino selezionato.

4. Si scriva lo script di shell che, assumendo di aver creato l'applicazione lato server con un progetto Nodejs denominato magazzino, esegua l'installazione dei pacchetti necessari ed avvii il server.
5. Si implementi in Java, usando ove possibile i Design Pattern, del codice che consenta di identificare le famiglie di prodotti della stessa marca, distribuiti all'interno di diversi magazzini. Si utilizzino i commenti del codice per descrivere la struttura delle classi realizzate.

TRACCE NON ESTRATTE

Traccia n. 1

Si vuole sviluppare una web application per la gestione dell'autenticazione e dei profili di utenza per un portale web.

1. Si sviluppi lo schema logico tramite linguaggio SQL del database dell'utenza che preveda almeno due entità: Utente che conserva le informazioni di autenticazione con password crittografata in SHA-512 e Anagrafica che conserva i dettagli anagrafici dell'utente stesso, in relazione 1:1.
2. Assumendo di avere a disposizione una tabella Utente, almeno con i campi id, username, password, e Anagrafica con almeno i campi id e nome e cognome, sviluppare il codice lato server che gestisca una richiesta web alla URL <https://<nome server>/login> contenente una stringa con la mail dell'utente e una seconda stringa contenente la password, inviata in chiaro, da confrontare con il contenuto del campo password che è invece crittografato in SHA-512. Il codice dovrà gestire il caso di utente non trovato o di password errata inviando in risposta il rendering di una pagina HTML contenente un apposito messaggio ovvero dovrà inviare una pagina contenente un messaggio di benvenuto all'utente con il proprio nome e cognome, in caso di login correttamente effettuato.
3. Si realizzi una pagina HTML costituita da un form contenente due campi di testo per l'inserimento di una mail con verifica lato client della sua correttezza sintattica e di una password con caratteri nascosti. La pagina conterrà il codice Javascript per invocare il server alla URL <https://<nome server>/login> con una richiesta web contenente una stringa con la mail dell'utente e una seconda stringa contenente la password, inviata in chiaro.



4. Si scriva lo script di shell che, assumendo di aver creato l'applicazione lato server con un progetto Nodejs denominato authentication, esegua l'installazione dei pacchetti necessari ed avvii il server.
5. Si implementi in Java, usando ove possibile i Design Pattern, del codice che consenta di effettuare una copia dei dati anagrafici e di quelli di autenticazione di un utente, rendendo il codice indipendente dalle classi. Si utilizzino i commenti del codice per descrivere la struttura delle classi realizzate.

Traccia n. 2

Si vuole sviluppare una web application per la gestione del personale dipendente di un'azienda.

1. Si sviluppi lo schema logico tramite linguaggio SQL del database degli impiegati che preveda almeno due entità: Impiegato che conserva le informazioni anagrafiche e del dipendente e Ruolo che conserva i dettagli legati ai ruoli aziendali come la denominazione, il reparto di appartenenza, le mansioni, il salario base per ruolo e così via, in una relazione denominata Carriera di tipo N:M che conservi al suo interno le date di inizio e fine del ruolo aziendale per ogni singolo dipendente.
2. Assumendo di avere a disposizione una tabella Impiegato, almeno con i campi id, cognome, nome, indirizzo e codice_fiscale, Ruolo con i campi descrizione e dipartimento, e una tabella di relazione Carriera con i campi dal e al, sviluppare il codice lato server che gestisca una richiesta web alla URL <https://<nome server>/carriera> contenente il cognome ed il nome dell'impiegato considerato. Il codice dovrà gestire il caso di impiegato non trovato inviando in risposta il rendering di una pagina HTML contenente un apposito messaggio ovvero dovrà inviare una pagina contenente una tabella con i ruoli ricoperti da ciascun impiegato il cui nome e cognome corrispondono a quanto richiesto.
3. Si realizzi una pagina HTML costituita da un form contenente un menu dropdown che verrà popolato all'atto del caricamento della pagina e che conterrà una lista di cognomi e nomi richiesti con apposita richiesta web alla URL <https://<nome server>/impiegati> da cui si assume che si ottenga un file impiegati.json la cui struttura è:

```
{"impiegati":      «identificativo», "nome": «cognome e nome>», ...j}
```

La pagina conterrà il codice Javascript per invocare il server alla URL <https://<nome server>/carriera> con una richiesta web contenente una stringa con il cognome e nome dell'impiegato selezionato.

4. Si scriva lo script di shell che, assumendo di aver creato l'applicazione lato server con un progetto Nodejs denominato personale, esegua l'installazione dei pacchetti necessari ed avvii il server.



5. Si implementi in Java, usando ove possibile i Design Pattern, del codice che consenta di aggiungere o eliminare un ruolo a un impiegato, a seconda del periodo in considerazione. Si utilizzino i commenti del codice per descrivere la struttura delle classi realizzate.