

TRACCE DELLE PROVE D'ESAME ESTRATTE

BANDO N. 367.134 DIITET ISTI CTER

CONCORSO PUBBLICO PER TITOLI ED ESAMI PER L'ASSUNZIONE CON CONTRATTO DI LAVORO A TEMPO PIENO E INDETERMINATO DI UNA UNITÀ DI PERSONALE PROFILO COLLABORATORE TECNICO ENTI DI RICERCA, VI LIVELLO PROFESSIONALE PRESSO L'ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "ALESSANDRO FAEDO" DI PISA AFFERENTE AL DIPARTIMENTO INGEGNERIA - ICT E TECNOLOGIE PER L'ENERGIA E I TRASPORTI DEL CONSIGLIO NAZIONALE DELLE RICERCHE

Prova scritta

Traccia A

In riferimento alle tecnologie basate sui linguaggi Java/javascript, il candidato illustri una tecnologia a sua scelta (tra le principali e più recenti), inquadrandole nell'ambito di scenari applicativi relativi allo sviluppo di applicazioni web con web services

Traccia B

In riferimento alle tecnologie per le basi di dati, il candidato illustri una tecnologia a sua scelta (tra le principali e più recenti), presentando un caso d'uso nell'ambito di scenari applicativi legati alla gestione ed elaborazione di dati multimediali

Traccia C

In riferimento alle tecnologie delle reti WAN e LAN e dei servizi internet, il candidato illustri una tecnologia a sua scelta (tra le principali e più recenti), inquadrandola nell'ambito di uno scenario applicativo di gestione e storage di dati multimediali

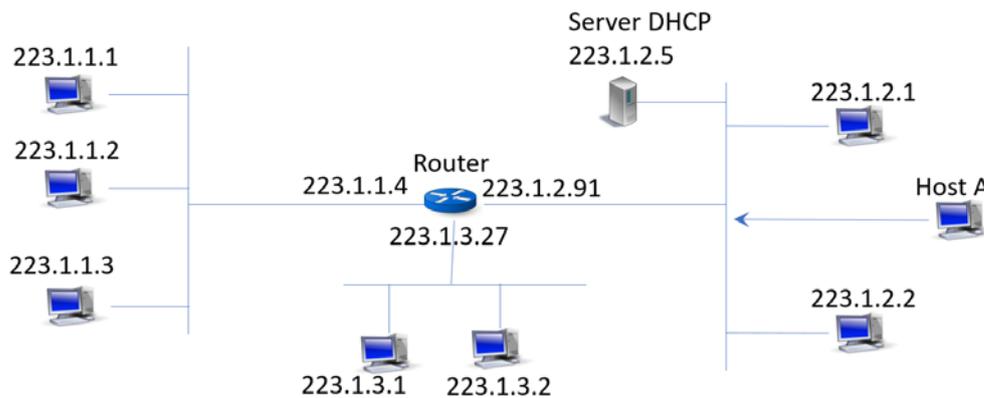
Prova teorico-pratica

Configurazione di reti WAN e LAN e gestione di servizi Internet per grandi infrastrutture dati.

D1. In riferimento alle reti private virtuali (VPN), spiegarne le funzionalità principali e illustrarne un caso d'uso.

D2. In riferimento al protocollo Dynamic Host Configuration Protocol (DHCP), spiegarne le funzionalità principali.

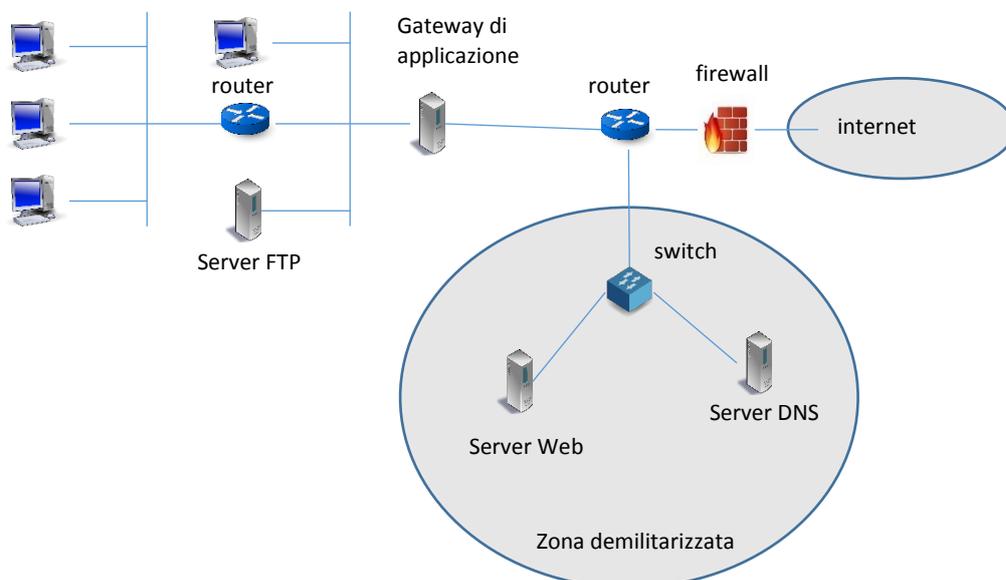
Si consideri un host A che si connette ad una rete locale organizzata come in figura, e in cui il server DHCP ha indirizzo 223.1.2.5. Si spieghi come si sviluppa il protocollo e quali messaggi vengono scambiati tra l'host A e il server DHCP.



D3. Illustrare l'uso di un firewall per la protezione di una rete. Si consideri poi un firewall su un router che connette un'organizzazione di indirizzi 232.128.16/24 con l'esterno, la cui lista di controllo degli accessi è mostrata nella tabella seguente. Si chiede di commentare la tabella e di discutere quale tipo di traffico può essere scambiato tra la rete dell'organizzazione e l'esterno. Nella tabella si assume che le regole siano ordinate per priorità decrescente.

Azione	Indirizzo sorgente	Indirizzo destinazione	Protocollo	Porta sorgente	Porta destinazione	bit di flag
consenti	232.128.16/24	Al di fuori di 232.128.16/24	UDP	>1023	53	----
consenti	Al di fuori di 232.128.16/24	232.128.16/24	UDP	53	>1023	----
consenti	232.128.16/24	Al di fuori di 232.128.16/24	TCP	>1023	80	Qualsiasi
consenti	Al di fuori di 232.128.16/24	232.128.16/24	TCP	80	>1023	ACK
blocca	tutto	tutto	tutto	tutto	tutto	tutto

D4. Si discuta l'architettura di rete presentata in figura, discutendo il ruolo di ogni componente.



D5. In riferimento al Network Address Translator (NAT), se ne discuta l'uso e le funzionalità. Inoltre, si consideri un NAT su un gateway la cui sottorete locale è 188.188.1.0/24, dove il gateway ha indirizzo 188.188.1.0, e la porta del gateway collegata ad internet ha indirizzo 200.210.230.161. Nella rete locale sono presenti due host con indirizzo 188.188.1.10 e 188.188.1.20 che, a partire da un certo tempo t, hanno stabilito le seguenti connessioni TCP (le connessioni sono elencate nello stesso ordine col quale sono state create):

IP sorgente	IP Destinazione	Porta sorgente	Porta destinazione
188.188.1.10	131.114.4.92	1900	3120
188.188.1.20	131.114.4.92	1901	110
188.188.1.10	188.188.1.20	944	25
188.188.1.10	188.188.1.0	950	80
188.188.1.20	131.114.4.92	1230	110
188.188.1.20	131.114.4.92	1300	256

Progettazione, implementazione e gestione di infrastrutture IT distribuite basate su virtualizzazione per la gestione di grandi quantità di dati; installazione e gestione di infrastrutture HW/SW e gestione di database

D6. Si descriva brevemente quando è vantaggioso passare alla virtualizzazione e quando non lo è.

D7. Si descriva brevemente il tipo di approccio alla virtualizzazione di Tipo 2 (o hosted).

D8. Un elemento fondamentale nella virtualizzazione dei server è lo storage. Si descriva brevemente il tipo di storage Nas o Network Attached Storage.

D9. Le caratteristiche dei sistemi NoSQL vengono spesso sintetizzate con l'acronimo BASE: si esponano brevemente ed esaurientemente i concetti espressi dall'acronimo.

D10. Descrivere l'approccio agentless per monitorare le *performances* delle infrastrutture software, discutendone i pro e i contro.

Sviluppo di applicazioni software con linguaggio Java e Javascript mediante l'uso di ambienti integrati di sviluppo (es. Eclipse o NetBeans); Sviluppo di applicazioni Web, in particolare basate su Web Services; Sviluppo di App per dispositivi mobili per dati multimediali

D11. L'interfaccia *Collection* è alla base della gerarchia di tutte le collezioni di Java; da questa interfaccia derivano tra le altre l'interfaccia *Set* e *List*. Descrivere la differenza tra *Set* e *List* specificando quando è più opportuno l'uso dell'una o dell'altra. Indicare quale è il risultato dell'esecuzione del metodo *main* della classe *Student*, di seguito riportata, giustificando la risposta.

```
public class Student {
    private final String name;
    private final String city;
    private final int age;
    public Student(String name, String city, int age) {
        this.name = name;
        this.city = city;
        this.age = age;
    }
    public String getCity() {return city;}
    public int getAge() {return age;}
    public String getName() {return name;}
    public static void main(String[] args) {
        Set<Student> students = new HashSet<>();
        Student a = new Student("Mario Rossi", "Pisa", 25);
```

```

        Student b = new Student(a.getName(), a.getCity(), a.getAge());
        Student c = new Student("Paolo Verdi", "Firenze", 35);
        students.add(a);
        students.add(a);
        students.add(b);
        students.add(c);
        for ( Student student : students ) {
            System.out.printf("%s, %s, %s\n",
                student.getName(),
                student.getCity(),
                student.getAge());
        }
    }
}

```

D12. Indicare e motivare quale è il risultato dell'esecuzione del metodo *main* della classe *Show* di seguito riportata. Il metodo *apply* della classe suddetta fa uso delle novità introdotte con Java 8 come *Stream API* e *Lambda Expression*. Si chiede di modificare tale metodo in modo da produrre il medesimo risultato, senza utilizzare i costrutti introdotti con Java 8.

```

public class Show {
    private final String title;
    private final String genre;
    private final int rate;
    public String getTitle() { return title; }
    public String getGenre() { return genre; }
    public int getRate() { return rate; }
    public Show(String title, String genre, int rate) {
        this.title = title;
        this.genre = genre;
        this.rate = rate;
    }
    public static void main(String[] args) {
        List<Show> shows = new ArrayList<>();
        shows.add(new Show("The Big Bang Theory", "Comedy", 8));
        shows.add(new Show("Modern Family", "Comedy", 8));
        shows.add(new Show("Lost", "Fantasy", 7));
        shows.add(new Show("Games Of Thrones", "Fantasy", 9));
        shows.add(new Show("Breaking Bad", "Crime", 9));
        shows.add(new Show("Sherlock", "Crime", 8));
        shows.add(new Show("Westworld", "Fantasy", 8));
        apply(shows, "Fantasy", 8);
    }
    private static void apply(List<Show> shows, String genre, int rate) {
        shows.stream().filter(show -> show.getRate() >= rate)
            .filter(show -> show.getGenre().equals(genre))
            .map(Show::getTitle)
            .findFirst()
            .ifPresent(System.out::println);
    }
}

```

D13. Un'università italiana ha un database MySQL composto dalle seguenti tabelle:

Studenti: studenti iscritti all'università

Professori: professori che insegnano all'università

Corsi: i corsi disponibili

Esami: esami sostenuti dagli studenti

Di seguito le tabelle in formato SQL:

<pre>CREATE TABLE IF NOT EXISTS `studenti` (`id` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT, `nome` VARCHAR(45) NOT NULL, `citta` VARCHAR(45) NOT NULL, `data_nascita` DATE NOT NULL, `data_iscrizione` DATE NOT NULL, `data_laurea` DATE NULL, `voto_laurea` TINYINT UNSIGNED NULL, `lode` TINYINT UNSIGNED NOT NULL DEFAULT '0', PRIMARY KEY (`id`)) ENGINE = InnoDB;</pre>	<pre>CREATE TABLE IF NOT EXISTS `professori` (`id` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT, `nome` VARCHAR(45) NOT NULL, PRIMARY KEY (`id`)) ENGINE = InnoDB;</pre>
<pre>CREATE TABLE IF NOT EXISTS `corsi` (`id` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT, `nome` VARCHAR(45) NULL, `professore` BIGINT UNSIGNED NOT NULL, PRIMARY KEY (`id`), INDEX `fk_corsi_professori_idx` (`professore` ASC), CONSTRAINT `fk_corsi_professori` FOREIGN KEY (`professore`) REFERENCES `professori` (`id`) ON DELETE CASCADE ON UPDATE NO ACTION) ENGINE = InnoDB;</pre>	<pre>CREATE TABLE IF NOT EXISTS `esami` (`studente` BIGINT UNSIGNED NOT NULL, `corso` BIGINT UNSIGNED NOT NULL, `data` DATE NOT NULL, `voto` TINYINT UNSIGNED NOT NULL, PRIMARY KEY (`corso`, `studente`), INDEX `fk_esami_studenti_idx` (`studente` ASC), CONSTRAINT `fk_esami_studenti` FOREIGN KEY (`studente`) REFERENCES `studenti` (`id`) ON DELETE RESTRICT ON UPDATE NO ACTION, CONSTRAINT `fk_esami_corsi` FOREIGN KEY (`corso`) REFERENCES `corsi` (`id`) ON DELETE RESTRICT ON UPDATE NO ACTION) ENGINE = InnoDB;</pre>

Il candidato scriva le query SQL (con riferimento al suddetto database MySQL) per estrarre i seguenti dati:

1. Estrarre l'elenco degli studenti non laureati. Riportare nome, città, data iscrizione degli studenti selezionati.
2. Estrarre l'elenco dei 5 anni che hanno avuto il maggior numero di laureati. Riportare l'anno e il numero di laureati dell'anno.
3. Estrarre l'elenco dei professori che hanno verbalizzato almeno 20 esami ai corsi dei quali sono docenti. Riportare il nome del professore e il numero di studenti che hanno sostenuto gli esami con i professori selezionati.

Infine, considerando le seguenti tabelle:

Tabella professori

id	nome
1	Enrico Fermi
2	Vint Cerf

Tabella corsi

id	nome	professore
1	Fisica	1
2	Reti	2
3	Fisica 2	1

Tabella esami

studente	corso	data	voto
1	1	09/06/2010	25
8	1	12/09/2010	23
3	3	01/07/2017	28

Tenendo presente le tabelle in formato SQL sopra indicate, il candidato discuta il risultato della seguente query:

1. `delete from professori where id = 1;`

D14. Una società vuole rendere disponibile tramite web-service i dati dei propri prodotti contenuti all'interno del loro database. Utilizzando le specifiche HTTP e REST progettare l'API per compiere le varie operazioni CRUD (Create, Read, Update, Delete) relativa alla tabella da esportare tramite web-service. In particolare, viene richiesto al candidato di descrivere in maniera tabellare il metodo HTTP, URL e codici di risposta per ognuna delle seguenti operazioni:

- Creazione prodotto
- Cancellazione prodotto
- Modifica prodotto
- Modifica parziale prodotto
- Prelevare singolo prodotto
- Prelevare lista prodotti

Si precisa che:

- Metodo HTTP: indicare in questa colonna il verbo o metodo HTTP opportuno per la funzione che si intende implementare (Esempio: GET, POST ecc...)
- URL: indicare l'URL corrispondente alla funzione. Trattandosi del disegno di una API è sufficiente indicare una URL relativa (Esempio: /api/prodotti), racchiudendo eventuali variabili nel *path* tra parentesi graffe.
- Codici di risposta: indicare i codici di risposta corretti per successo o fallimento dell'operazione. Per fallimento non si consideri il caso in cui il server risponda con un errore di tipo 500 e considerare solo i codici di errore 400 (Bad Request), 404 (Not Found) e 409 (Conflict), indicando dove utilizzarli e con quale significato.

Supponendo che la tabella dei prodotti contenga i seguenti campi:

- modello
- marca
- categoria
- prezzo

utilizzando i metodi HTTP e le URL indicate nella tabella e il JSON come formato dei dati, si esponga un esempio di interazione con il server per la modifica di un prodotto.

D15. In riferimento alle funzioni pure o impure nei linguaggi funzionali, indicare le differenze tra le due tipologie e indicare i vantaggi dell'una rispetto all'altra. Inoltre, utilizzando il linguaggio Javascript fare un esempio di funzione pura e uno di funzione impura.