Ministero dell'Istruzione, dell' Università e della Ricerca
ESAME DI STATO DI ISTRUZIONE SECONDARIA SUPERIORE
Indirizzo: ITIA - INFORMATICA E TELECOMUNICAZIONI ARTICOLAZIONE INFORMATICA
Tema di: INFORMATICA e SISTEMI E RETI

Il candidato svolga la prima parte della prova e due tra i quesiti proposti nella seconda parte.

PRIMA PARTE

La ditta InfoService offre servizi di assistenza hardware-software e consulenza informatica in genere.
Essa opera a livello regionale ed al suo interno lavorano una cinquantina di dipendenti che si occupano di settori specifici quali assistenza hardware a dispositivi informatici, configurazione di server e relativi servizi, assistenza software e sviluppo di nuove applicazioni su richiesta dei clienti, personalizzazione di software già esistenti.

Per ottimizzare la gestione degli interventi di assistenza presso i propri clienti, InfoService ha deciso di sviluppare un sistema di ticketing. Il sistema prevede che i clienti, accedendo al portale web attraverso le proprie credenziali, possano richiedere interventi di personale tecnico per la risoluzione di problemi di natura hardware o software relativi ai servizi offerti da InfoService.

La richiesta comporta l'apertura di un ticket nel quale, oltre ai dati del richiedente, già presenti in quanto associati al suo account, il cliente descriverà il problema riscontrato per il quale richiede l'intervento. A seconda della problematica, l'intervento verrà effettuato da remoto oppure presso il cliente. Il personale di InfoService addetto all'helpdesk individuerà il tecnico a cui assegnare il ticket.

I tecnico, effettuato l'intervento, registrerà immediatamente in un report online l'attività svolta e il tempo impiegato: se il problema è stato risolto, provvederà a chiudere il ticket, altrimenti questo resterà aperto in attesa di ulteriori interventi. Il cliente dovrà convalidare il report, avendo anche la possibilità di esprimere un proprio commento.

Il candidato analizzi la realtà descritta e, fatte le opportune ipotesi aggiuntive, individui una soluzione che a suo motivato giudizio sia la più idonea per sviluppare i seguenti punti:

1. il progetto, anche mediante rappresentazioni grafiche, dell'infrastruttura tecnologica ed informatica necessaria a gestire il servizio nel suo complesso, dettagliando:
   a) le risorse hardware ed i servizi software necessari per sviluppare il sistema di ticketing;
   b) le misure che possono essere adottate per gestire con la massima sicurezza le informazioni trattate dal sistema di ticketing;
   c) le modalità con le quali i tecnici provvedono online alla compilazione del report approfondendo:
      - le caratteristiche della connessione alla rete Internet sia della sede centrale di InfoService sia dei dispositivi in dotazione al personale tecnico in trasferta;
      - gli aspetti di sicurezza relativi alla comunicazione tra i dispositivi client in dotazione al personale tecnico e il sistema centrale di InfoService;
      - le modalità attraverso le quali il cliente convalida il report compilato dal tecnico, eventualmente esprimendo il proprio commento;
2. il progetto della base di dati per la gestione del sistema di ticketing: in particolare si richiede il modello concettuale ed il corrispondente modello logico;
3. lo sviluppo in linguaggio SQL delle query che consentono di ottenere le seguenti informazioni:
   - elenco dei ticket attualmente aperti riportando il nome del cliente che li ha aperti, la data di apertura, il tecnico che li sta seguendo;
   - tempo medio di chiusura dei ticket completati in un certo intervallo temporale fornito in ingresso.

SECONDA PARTE

Il candidato risponda a due quesiti a scelta tra quelli sotto riportati.

I. In relazione al tema proposto nella prima parte, si consideri che solo i dirigenti di InfoService possano monitorare l'attività del personale tecnico che effettua interventi di assistenza. Il candidato, dopo aver apportato le opportune modifiche al database sviluppato nella prima parte, progetti l'architettura di massima delle pagine necessarie ad implementare la funzione sul portale web del sistema di ticketing. Codifichi poi in un linguaggio a sua scelta le pagine che consentono al solo personale dirigente di visualizzare le statistiche relative agli interventi di assistenza (come ad es. la seconda query del punto 3 della prima parte).

II. In relazione al tema proposto nella prima parte, il candidato definisca il piano di indirizzamento della rete interna della sede principale di InfoService e le modalità con le quali viene controllato l'accesso di dispositivi wifi alla stessa. Approfondisca quindi i fattori che consentono di garantire la continuità del servizio dettagliando le risorse hardware e i servizi software che ritiene idonei per il caso in questione.

III. Lo sviluppo della rete Internet e l'incremento esponenziale del numero di dispositivi che si prevede verranno ad essa connessi, anche in conseguenza del forte impulso dato in tal senso dall'Internet delle cose (IoT), sta favorendo la diffusione del protocollo IPv6. Si espongano le caratteristiche del suddetto protocollo e le differenze rispetto al protocollo IPv4.

IV. Nell'interazione con un'applicazione web dinamica, l'utente compie azioni che richiedono l'invio di dati al server. Il candidato esamini i metodi attraverso cui è possibile trasferire al server i dati generati lato client dall'utente durante l'uso dell'applicazione, evidenziandone le specificità e i differenti usi. Fornisca al riguardo esempi di casi di utilizzo per le differenti modalità.

Commento

La prova è centrata su un sistema di ticketing che un'azienda informatica ha deciso di sviluppare per ottimizzare la gestione degli interventi di assistenza tecnica ai propri clienti.

Un sistema di ticketing è un servizio che raccoglie e tiene traccia di tutti i contatti, provenienti da vari canali (portale web, e-mail, chat, telefono), che intercorrono tra il personale dell'azienda e i clienti che hanno richiesto assistenza. Si tratta di una problematica complessa ma la traccia propone un sistema di ticketing con un solo canale di comunicazione ("*Il sistema prevede che i clienti, accedendo al portale web attraverso le proprie credenziali, possano richiedere interventi di personale tecnico per la risoluzione di problemi di natura hardware o software relativi ai servizi offerti da InfoService*") e ne descrive il funzionamento in modo molto sintetico, permettendo al candidato di fare ipotesi aggiuntive che semplificano notevolmente la soluzione.

Soluzione prima parte

1. *il progetto, anche mediante rappresentazioni grafiche, dell'infrastruttura tecnologica ed informatica necessaria a gestire il servizio nel suo complesso*

Leggendo attentamente la traccia, si evincono alcune cose molto significative:

- la ditta InfoService offre servizi di assistenza hardware-software e consulenza informatica e opera a livello regionale in un'unica sede
- dispone già di un portale web
- può contare su una cinquantina di dipendenti con un profilo tecnico informatico specializzato
- il sistema di ticketing riguarda esclusivamente l'assistenza tecnica per risolvere problemi di natura hardware o software

Si tratta, dunque, di un'azienda di dimensioni medio-piccole e ragionevolmente con un numero importante ma non elevatissimo di clienti.
Per il proprio sistema informatico, InfoService ha adottato una soluzione "on-site" implementando una sala server all'interno della propria sede. Ha fatto questa scelta tenendo conto che dispone già di figure professionali in grado di amministrare e manutenere una sala server e privilegiando i vantaggi di:

1. essere l'unico gestore dei server
2. poter garantire la sicurezza in modo autonomo e intervenire immediatamente in caso di problemi, guasti e malfunzionamenti
3. poter proteggere la propria rete senza aspettare l'intervento di altri soggetti (ad esempio nel caso di attacchi DDos)

In alternativa avrebbe potuto scegliere una soluzione esterna, affidando la sicurezza di apparecchiature e dati ad un Data Center specializzato, in particolare:

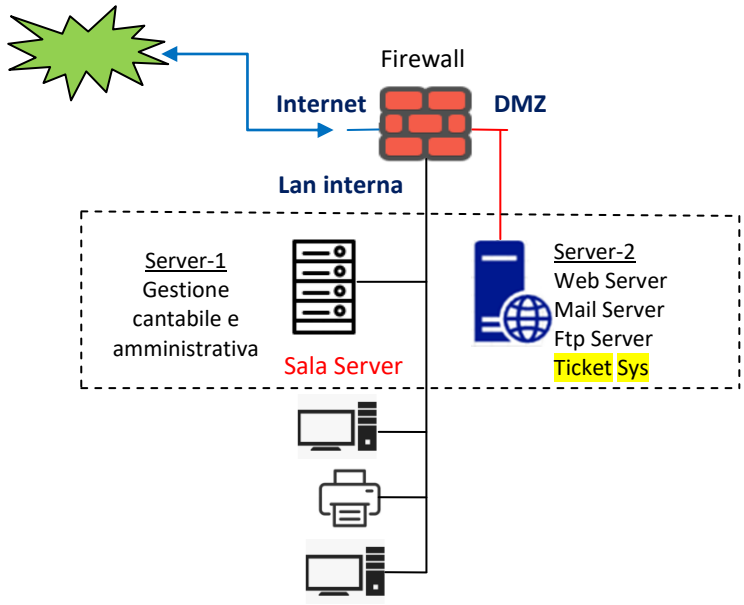
- una soluzione di **colocation**
  Un server di proprietà dell'azienda viene ospitato nel Data Center di un provider. Il Data Center garantisce la sorveglianza, la manutenzione hardware e la sicurezza informatica. L'azienda cura invece la manutenzione sistemistica attraverso un software di controllo remoto
- una soluzione **Cloud server**
  Un server virtualizzato mette a disposizione le proprie risorse in termini di RAM, memoria di massa e processore. L'utente decide di volta in volta le risorse di cui ha realmente bisogno per svolgere una determinata attività.

Queste soluzioni sono molto innovative e flessibili e presentano diversi punti di forza:

1. costi di manutenzione e gestione ridotti
2. sistemi ridondanti per garantire l'accesso alla rete, l'elettricità e il controllo del clima
3. connettività di rete costante e di qualità tramite cavi in fibra ottica ultraveloci
4. misure avanzate per la protezione dai rischi fisici (controllo accessi, fuoco, acqua, ecc) e la protezione dei software e dei dati (antivirus e firewall, backup quotidiani, Disaster Recovery Plan)
5. personale ICT molto esperto
6. spese interamente deducibili

Ovviamente la scelta tra soluzioni on-site e soluzioni esterne è legata alla natura e alle esigenze di ciascuna azienda.

In questo caso InfoService ha scelto la soluzione on-site ritenendo irrinunciabili i vantaggi che derivano dall'avere il controllo completo del proprio sistema informatico.



Rete aziendale InfoService

InfoService dispone di due server, il Server-1 dedicato alla gestione contabile ed amministrativa dell'azienda e il Server-2 che gestisce il sito web aziendale, la posta elettronica e il servizio FTP e sul quale viene implementato anche il sistema di ticketing (TicketSys)

Il Server-2 è raggiungibile dall'esterno tramite Internet e per questo è esposto ad eventuali attacchi hacker. Per impedire che un eventuale attacco possa compromettere l'intera rete aziendale, è stata creata una zona protetta (DMZ) che ospita il Server-2. La protezione è assicurata da un firewall con tre interfacce (3 legs ovvero "a tre gambe") che collegano rispettivamente:

a) la rete **Internet** (Wan)
b) la **Lan interna** (Il Server-1, i pc, le stampanti, ecc.)
c) la **DMZ** con il Server-2 esposto ad Internet

Il firewall permette al personale che lavora sulla Lan interna di collegarsi ad Internet e di accedere velocemente alla DMZ ma controlla fortemente il traffico dalla DMZ alla Lan interna, proteggendo quest'ultima da intrusioni indesiderate sul Server-2.

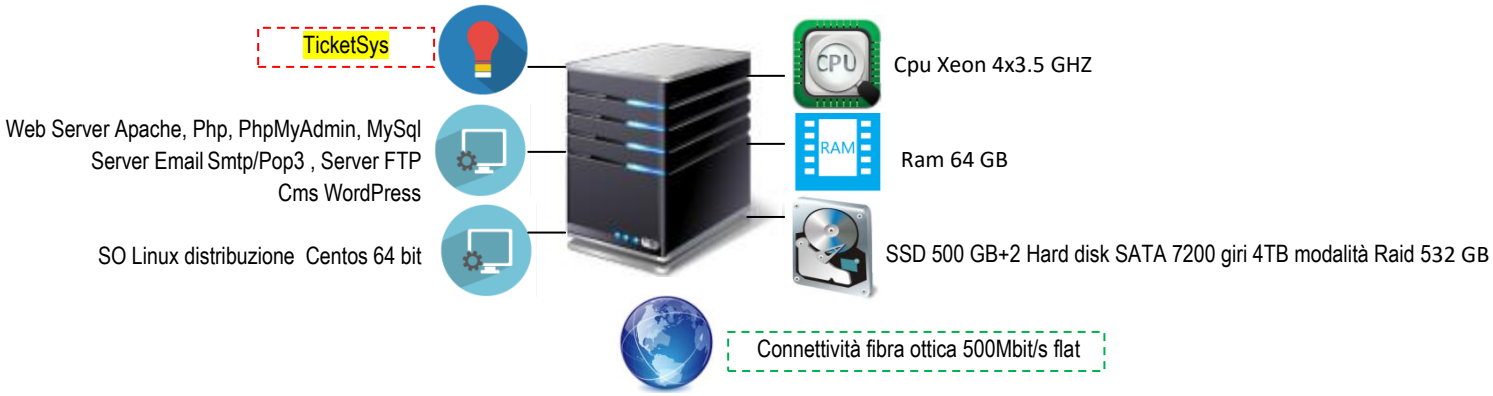
La configurazione hardware e software di massima del Server-2 è la seguente:

Hardware	**Software**
CPU Xeon 4 x 3.5 GHz	SO Linux distribuzione CentOS 64 bit
Ram 64 GB	Web Server Apache, Php, PhpMyAdmin, MySql
SSD 500 GB+2 Hard disk SATA 7200 giri 4TB modalità Raid 5	Mail Server Zimbra
	Ftp Server ProFTP
	CMS Wordpress, per la creazione e gestione del sito web aziendale
Connettività	TicketSys : sistema di ticketing
500 Mbit/s Flat fibra ottica	

Per realizzare il sistema di ticketing InfoService ha sviluppato il software **TicketSys**, un'applicazione web based, leggera, veloce e flessibile, scritta principalmente in Php, e "appoggiata" su un database MySql. Le pagine web sono costruite con un approccio "responsive" ovvero con un design che ne consente una fruizione ottimale su ogni tipo di dispositivo, dagli schermi dei pc desktop ai tablet e agli smartphone.

Per questi motivi i tecnici InfoService e i clienti possono accedere al sistema di ticketing da qualsiasi dispositivo utilizzando semplicemente un browser web.

L'accesso avviene utilizzando un link presente sull'home page del sito aziendale.



Configurazione Hardware/software Server-2

Senza descriverle per brevità, le misure per la sicurezza della Sala Server e del Server-2 sono le seguenti:

Sicurezza Sala Server	Sicurezza Server-2
• Gruppo di continuità elettrica • Condizionamento del clima • Sistema antincendio • Finestre e porte blindate, accessi controllati	▪ Alimentatori ridondanti ▪ Dischi Raid ▪ Dischi Hot Swap
	▪ Firewall ▪ Antivirus
	▪ Backup su HD esterno ▪ Backup incrementali sul cloud ▪ Piano di protezione dagli attacchi DDoS ▪ Disaster Recovery Plan
	▪ Amministratori esperti!

Per la protezione dei dati relativi al sistema di ticketing così come per la protezione dei dati del sito web aziendale, sul web server Apache è stato abilitato il protocollo HTTPS.
HTTPS, acronimo di HyperVarchar Transfer Protocol over SSL, è una variante sicura e certificata del protocollo HTTP. Come HTTP lavora con un'architettura client/server in cui un client (il browser di un utente) esegue una richiesta e il server (il sito web) restituisce la risposta, ma usando il livello SSL, acronimo di Secure Sockets Layer, cripta i dati e li rende indecifrabili. In questo modo, HTTPS garantisce che solamente il client e il server siano in grado di conoscere il contenuto della comunicazione e impedisce ad altri soggetti di leggere e modificare i dati che vengono scambiati

Un ruolo importante per la protezione dei dati relativi al sistema di ticketing è svolto anche dalla qualità del software TicketSys. I dati sono più sicuri se il software è ben architettato, prevede un efficace sistema di autenticazione ed è scritto utilizzando gli accorgimenti utili ad impedire eventuali attacchi esterni attraverso i form dell'interfaccia web.

Autenticazione

In un qualsiasi sistema software il meccanismo di autenticazione è fondamentale per la sicurezza e la complessità delle strategie adottate varia in base al contesto, all'importanza dei dati da proteggere e alla pericolosità e all'impatto di eventuali violazioni.

I metodi di autenticazione sono molteplici e vanno dal login con username e password ai sistemi di riconoscimento biometrici (impronte digitali, voce, ecc) e all'uso di smart card e di token

Per accedere al sistema, TicketSys prevede una semplice procedura di login che verifica le credenziali dell'utente e riconosce il ruolo (cliente, dirigente, addetto helpdesk o tecnico) che egli riveste. A seconda del ruolo, l'utente potrà utilizzare tutte le funzionalità del software, o potrà utlilizzarne solo alcune.

Un filtro importante per la sicurezza, è ovviamente previsto al momento della registrazione dell'utente che non è automatica ma viene convalidata da InfoService sulla base del controllo della mail aziendale.

TicketSys prevede che le password siano robuste (con una lunghezza adeguata e composta da maiuscole, minuscole, numeri e simboli di interpunzione) e vengano cambiate obbligatoriamente ogni 6 mesi

Uso di sessioni

1. Quando un utente viene autorizzato ad accedere al sistema tramite la procedura di login, viene impostata la variabile di sessione "autorizzato". Per evitare che un malintenzionato possa accedere direttamente ad una pagina php senza passare per il login, l'accesso a tutte le pagine php di TicketSys viene protetto da un semplice script che esegue il test della variabile di sessione "autorizzato".
   Nella pagina di login, se l'utente viene accettato, si imposta $_SESSION["autorizzato"]= "SI";
   All'inizio di ciascuna pagina php il codice:

```
<?php
    session_start();
    if (!isset($_SESSION["autorizzato"])) {
        die("Non autorizzato! Accesso negato");// oppure  header('Location:login.php');
    }
?>
```

   blocca l'utente che non ha eseguito il login (oppure lo ridirige alla pagina di login)
2. Le sessioni vengono correttamente create e distrutte
3. Vengono utilizzati gli accorgimenti per far fronte alle vulnerabilità delle sessioni php. Questo punto è abbastanza complesso e lascio agli studenti più motivati il compito di approfondire i problemi (Session Hijacking e Session Fixation) e le buone pratiche php per farvi fronte

Validazione dell'input

I campi dei form HTML che permettono la creazione, gli aggiornamenti e la chiusura dei ticket attraverso l'interazione tra clienti e tecnici, se non vengono correttamente validati, rappresentano una seria minaccia per la sicurezza. In particolare sono soggetti al problema dell'SQL-injection che consiste nel codice malevolo che, attraverso i form e sfruttando le query SQL, un hacker può inviare al database MySql. Per contrastare l'SQL-injection e più in generale per validare i dati inseriti in un form e inviati dal browser al server via GET o via POST, TicketSys utilizza le buone pratiche di eseguire sempre il typecast dei numeri e di sottoporre le stringhe al filtro di escaping dei caratteri speciali.

Esempio di un form con i campi "numero" e "descrizione" che tramite submit vengono inviati al server via GET

```
$conn=mysqli_connect(($host, $username, $password, $database);
$numero=(int)$_GET["numero"];          //typecast
$descrizione=$_GET["descrizione"];
$descrizione= mysqli_real_escape_string( $conn, $descrizione);      //escaping caratteri speciali
…
```

Le buone pratiche per la validazione dell'input non si esauriscono ovviamente con quelle appena viste. I programmatori esperti adottano altre tecniche specifiche per far fronte ai tanti problemi di sicurezza presenti in un ambiente aperto e ostile come il WEB.

In ogni caso le misure di protezione del software, per quanto complesse e sofisticate, aumentano la sicurezza ma non riescono a garantire il rischio zero e la difesa da tutti gli attacchi possibili; per questo motivo resta fondamentale proteggere adeguatamente il server e utilizzare una connessione cifrata HTTPS.

2. *il progetto della base di dati per la gestione del sistema di ticketing: in particolare si richiede il modello concettuale ed il corrispondente modello logico;*

Ipotesi aggiuntive

La descrizione del sistema di ticketing è molto sintetica e non parla esplicitamente dei costi degli interventi di assistenza tecnica. Parla di tempo impiegato ma non cita i costi di trasferta e dei materiali (accessori, pezzi di ricambio, cavi, ecc) utilizzati per l'assistenza hardware.

Per semplicità, ipotizziamo che i clienti e InfoService hanno stipulato un contratto che per ciascun ticket, riguardante uno o più interventi, prevede la fatturazione:

1. dei costi di manodopera in base ai tempi impegati
2. dei costi di trasferta in base alle distanze
3. dei materiali eventualmente utilizzati per l'assistenza hardware

Con questo tipo di contratto, una volta ricevuta la richiesta di assistenza, InfoService interviene senza dover fare un preventivo di spesa e aspettare l'approvazione del cliente.
Ipotizziamo inoltre che per interagire con i tecnici, i clienti possano utilizzare solo messaggi testuali (non è prevista la possibiltà di inviare allegati con grafici, immagini, screenshot e video)

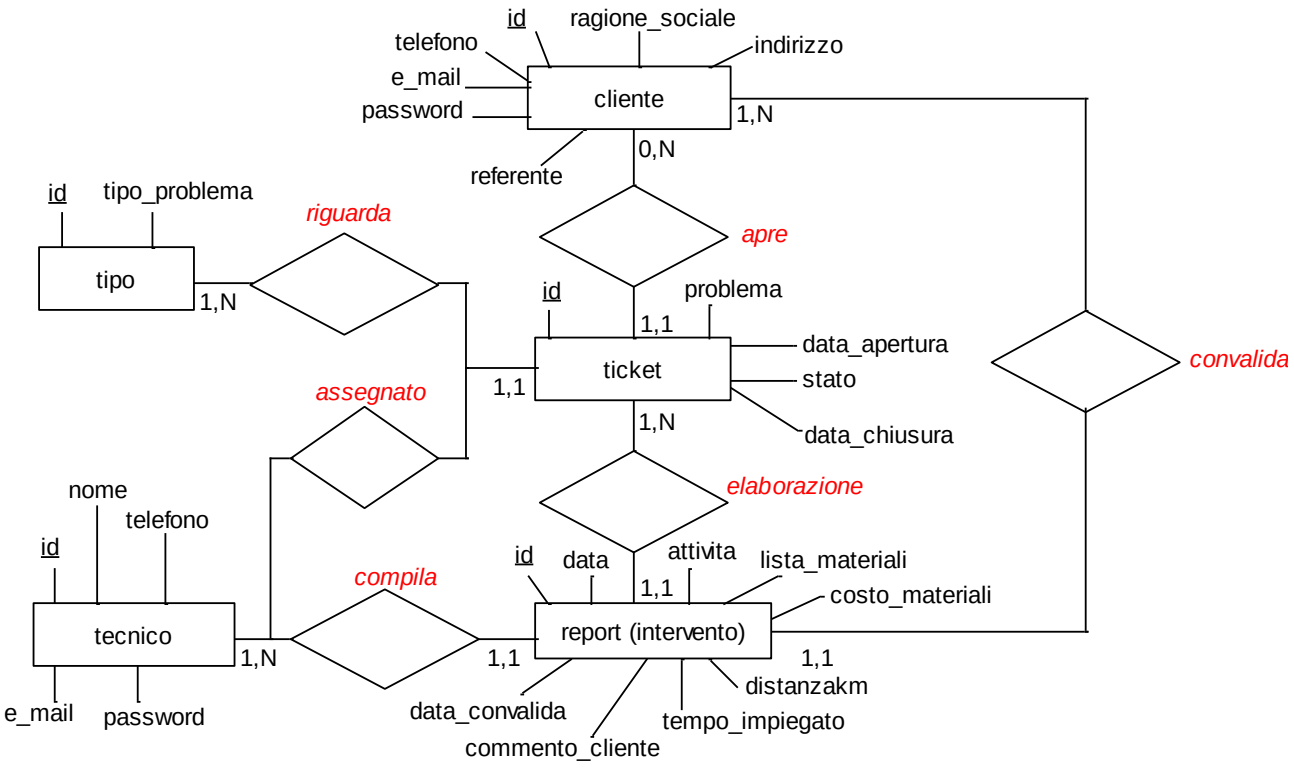
Analisi del sistema di ticketing

- Eseguendo il Login, il cliente accede a TicketSys, apre un ticket e comunica il problema riscontrato. Il sistema con una mail avverte il cliente che il ticket è stato preso in carico
- Il personale di InfoService addetto all'helpdesk, utilizzando un algoritmo di ottimizzazione degli interventi e in base al problema riscontrato, individua il tecnico a cui assegnare il ticket
- Il tecnico effettua l'intervento, da remoto o presso il cliente, e compila immediatamente un report online registrando:
  - le attività svolte
  - il tempo impiegato espresso in ore e/o frazioni di ore
  - la lista eventuale dei materiali utilizzati e il loro costo complessivo
  - la distanza chilometrica eventualmente percorsa

  Se il problema è stato risolto, chiude il ticket altrimenti esegue altri interventi
- Il cliente è costantemente aggiornato sull'evoluzione del ticket e prende visione dei report degli interventi. Convalida ciascun report esprimendo eventualmente un commento

Diagramma E/R

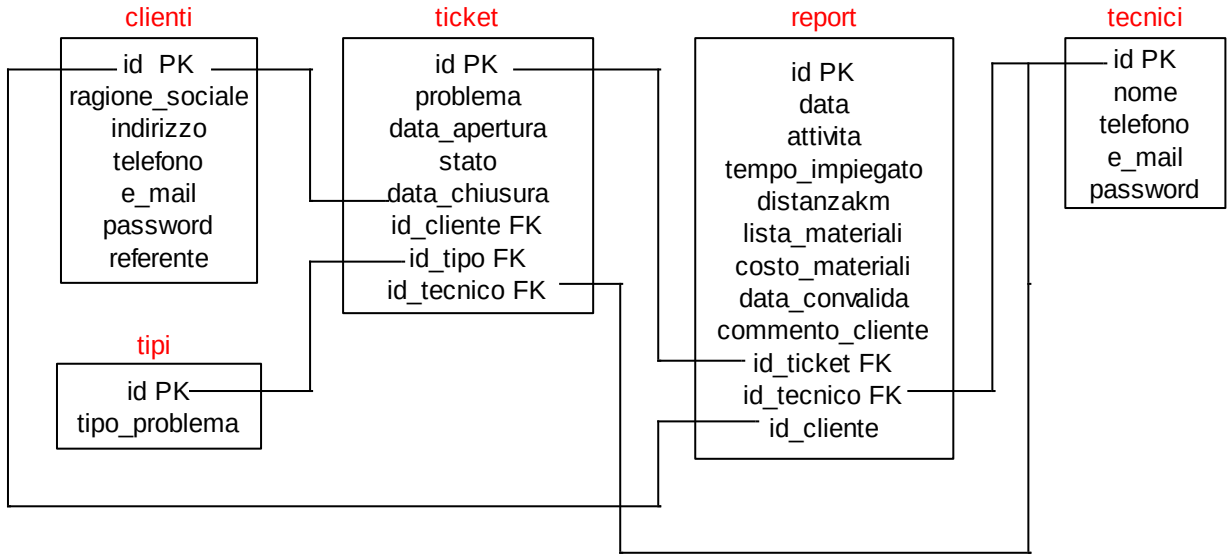
- si prendono in considerazione gli attributi più importanti delle varie entità
- ad ogni intervento corrisponde un report
- lo stato del ticket vale 0 se il ticket è aperto, vale 1 se il ticket è stato chiuso



Lettura del diagramma E/R

- Ogni cliente può aprire da 0 a N ticket, ogni ticket può essere aperto da un cliente
- Ogni ticket riguarda un tipo (di problema), ogni tipo può riguardare da 1 a N ticket
- Ogni ticket viene assegnato ad un tecnico, ad ogni tecnico possono essere assegnati da 1 a N ticket
- Per ogni ticket vengono elaborati 1 o più report (a fronte di altrettanti interventi), ogni report riguarda un ticket
- Ogni report viene compilato da un tecnico, ogni tecnico può compilare da 1 a N report
- Ogni report viene convalidato da un cliente, ogni cliente può convalidare da 1 a N report

Schema logico



Definizione della struttura delle tabelle

Si fa riferimento a MySql. Nome del database: ticketsys

tabella	campi	key	tipo	lung.	Descrizione/note
tipi	id	PK	int	5	Contatore auto_increment
	tipo_problema		varchar	40	Ad esempio: rottura hardware, malfunzionamento software, guasto rete…
tecnici	id	PK	int	5	Contatore auto_increment
	nome		varchar	40	
	telefono		varchar	30	
	e-mail		varchar	30	Viene utilizzata come username nel login
	password		varchar	32	Codifica MD5()
clienti	id	PK	int	5	Contatore auto_increment
	ragione_sociale		varchar	60	
	indirizzo		varchar	80	
	telefono		varchar	30	
	e-mail		varchar	30	Viene utilizzata come username nel login
	password		varchar	32	Codifica MD5()
	referente		varchar	40	
ticket	id	PK	int	8	Contatore auto_increment
	problema		text		
	data_apertura		date		
	data_chiusura		date		
	stato		int	1	0→ ticket aperto 1→ ticket chiuso Quando il tecnico chiude il ticket (stato=1) viene valorizzata la data_chiusura
	id_cliente	FK	int	5	Fa riferimento a clienti.id
	id_tipo	FK	int	5	Fa riferimento a tipi.id
	id_tecnico	FK	int	5	Fa riferimento a tecnici.id

tabella	campi	key	tipo	lung.	Descrizione/note
report	id	PK	int	8	Contatore auto_increment
	data		date		
	attivita		text		Riservato dal software al tecnico
	tempo_impiegato		decimal	5,2	Riservato dal software al tecnico
	distanzakm		int	5	Riservato dal software al tecnico
	lista_materiali		text		Riservato dal software al tecnico
	costo_materiali		decimal	8,2	Riservato dal software al tecnico
	data_convalida		date		Riservato dal software al cliente
	commento_cliente		text		Riservato dal software al cliente
	id_ticket	FK	int	8	Fa riferimento a ticket.id
	id_tecnico	FK	int	5	Fa riferimento a tecnici.id
	id_cliente	FK	int	5	Fa riferimento a clienti.id

Query MySql per la creazione del database e delle tabelle

```
create database ticketsys

create table if not exists tipi
( id int(5) auto_increment primary key,
  tipo_problema varchar(40) not null);

create table if not exists tecnici
( id int(5) auto_increment primary key,
  nome varchar(40) not null,  telefono varchar(30),
  e_mail varchar(30),  password varchar(32));

create table if not exists clienti
( id int(5) auto_increment primary key,
  ragione_sociale varchar(60) not null,
  indirizzo varchar(80),  telefono varchar(30),
  e_mail varchar(30),  password varchar(32),
  referente varchar(40));

create table if not exists ticket
( id int(8) auto_increment primary key,
  problema text not null,
  data_apertura date,
  stato int(1),
  data_chiusura date,
  id_cliente int(5),  id_tipo int(5),  id_tecnico int(5),
  foreign key (id_cliente)  references clienti(id),
  foreign key (id_tipo)  references tipi(id),
  foreign key (id_tecnico)  references tecnici(id) );

create table if not exists report
( id int(8) auto_increment primary key,
  data date,
  attivita text,
  tempo_impiegato decimal(5,2),
  distanzakm int(5),
  lista_materiali text,
  costo_materiali decimal(8,2),
  data_convalida date,
  commento_cliente text,
  id_ticket int(8),id_tecnico int(5), id_cliente int(5),
  foreign key (id_ticket)  references ticket(id),
  foreign key (id_tecnico)  references tecnici(id),
  foreign key (id_cliente)  references clienti(id)  );
```

3. lo sviluppo in linguaggio SQL delle query che consentono di ottenere le seguenti informazioni:

- *elenco dei ticket attualmente aperti riportando il nome del cliente che li ha aperti, la data di apertura, il tecnico che li sta seguendo;*
- *tempo medio di chiusura dei ticket completati in un certo intervallo temporale fornito in ingresso.*

Query "elenco dei ticket…"

```
select ticket.id as "N. Ticket", DATE_FORMAT(ticket.data_apertura,"%d/%m/%Y") as "Data apertura",
clienti.ragione_sociale as "Nome Cliente",tecnici.nome as "Tecnico InfoService"
from ticket, clienti, tecnici
where ticket.stato= 0
and ticket.id_cliente=clienti.id
and ticket.id_tecnico=tecnici.id
order by data_apertura
```

N. Ticket	Data apertura	Nome Cliente	Tecnico InfoService
324	03/03/2020	TeramoColor	Guido Lavespa
367	09/03/2020	Rosa rossa Spa	Paolo Rossi
392	12/03/2020	Cementi Pescara srl	Remo Labarca
404	16/03/2020	Ingegneria DB Abruzzo	Guido Lavespa

NB: la funzione DATE_FORMAT() permette di stampare la data nel formato gg/mm/aaaa

Query "tempo medio di chiusura…"

Il tempo medio di chiusura di un ticket viene espresso in giorni.

1. Si crea una tabella temporanea "*temp*" nella quale si memorizza la differenza espressa in giorni tra data_chiusura e data_apertura di ciascun ticket completato in un certo intervallo temporale
2. si esegue la query che calcola la media richiesta

Ad esempio se l'intervallo fornito in ingresso è il mese di maggio 2020 (dal 01/05/2020 al 31/05/2020)

```
create table temp as (
        Select DATEDIFF(data_chiusura,data_apertura) as differenza
        from ticket
        where stato=1
        and data_chiusura>="2020/05/01" and data_chiusura<="2020/05/31")

Select ROUND(avg( differenza ),2)  AS "Ticket completati a maggio 2020: tempo medio di chiusura in giorni"
from Temp
```

Ticket completati a maggio 2020: tempo medio di chiusura in giorni
3.33

NB: la funzione DATEDIFF() calcola la differenza in giorni tra due date; la funzione ROUND() arrotonda il tempo medio calcolato a 2 cifre decimali

Soluzione seconda parte

I. *In relazione al tema proposto nella prima parte, si consideri che solo i dirigenti di InfoService possano monitorare l'attività del personale tecnico che effettua interventi di assistenza. Il candidato, dopo aver apportato le opportune modifiche al database sviluppato nella prima parte, progetti l'architettura di massima delle pagine necessarie ad implementare la funzione sul portale web del sistema di ticketing. Codifichi poi in un linguaggio a sua scelta le pagine che consentono al solo personale dirigente di visualizzare le statistiche relative agli interventi di assistenza (come ad es. la seconda query del punto 3 della prima parte).*

Il sistema di ticketing si basa sulla web application TicketSys, scritta in Php, Database MySql, e installata sul Server web on-site di InfoService. I clienti e il personale dell'azienda (Dirigenti, Addetti helpdesk, Tecnici) possono accedervi con qualsiasi dispositivo, pc, tablet o smartphone, semplicemente utilizzando un browser.

TicketSys è un software articolato e complesso ed offre molte funzionalità che però non sono fruibili da tutti gli utenti. In base al ruolo di chi accede, riconosciuto nella pagina di login, il sistema abilita determinate funzionalità e ne vieta altre.

Nel nostro caso, solo quando riconosce il ruolo di "Dirigente", abilita l'utente ad utilizzare la funzionalità "Statistiche interventi assistenza".

Nel database ticketsys, occorre creare la tabella "dirigenti"

tabella	campi	key	tipo	lung.	Descrizione/note
dirigenti	id	PK	int	5	Contatore auto_increment
	nome		varchar	40	
	telefono		varchar	30	
	e-mail		varchar	30	Viene utilizzata come username nel login
	password		varchar	32	Codifica MD5()

```
create table if not exists dirigenti
( id int(5) auto_increment primary key,
  nome varchar(40) not null,
  telefono varchar(30),
  e_mail varchar(30),
  password varchar(32));
```

Se si esegue il login e si sceglie il ruolo Dirigente (il valore passato via POST è "D")

nella pagina php che controlla le credenziali, oltre alla variabile di sessione "autorizzato", viene impostata la variabile di sessione "ruolo" che assume il valore "D"

Codice "login.html"

```
<!doctype html>
<head>
    <title>Login/TicketSys</title>
    <link rel="stylesheet" href="mystyle.css" rel="stylesheet">
</head>
<script type="text/JavaScript">
    function valida()
    {
                var e_mail = document.form1.e_mail.value;
                var password = document.form1.password.value;
                if (e_mail == "" || password== ""){
                            alert("Credenziali incomplete!");
                            return false;
                }
     }
</script>
<body>
     <div class="container">
                <h1>InfoService TicketSys</h1>
                <?php
                      $oggi=getdate();
                      echo "Teramo ".$oggi["mday"]."-".$oggi["mon"] ."-".$oggi["year"];
                ?>
                <h2>Login</h2>
                <form  onsubmit="return valida()" name="form1" method='POST' action="controllacredenziali.php">
                    <div class="box">
                            <input type="radio" name="ruolo" value="C" checked><label>Cliente</label><br>
                            <input type="radio" name="ruolo" value="T"><label>Tecnico</label><br>
                            <input type="radio" name="ruolo" value="A"><label>Addetto helpdesk</label><br/>
                            <input type="radio" name="ruolo" value="D"><label>Dirigente</label><br/><br/>
                            <label>E-mail utente</label><br/>
                            <input type='text' name='e_mail'  maxlength="30" /><br/>
                            <label>Password<label><br/>
                            <input type='password' name='password' maxlength="16" /> <br/><br/>
                            <input type='submit' value='Accedi'/>
                            <input type='reset' Value='Reset' />
                            <br/><br/><a href="#" style="text-decoration:none">Registrati</a>
                    </div>
                </form>
                <br/>
                <br/>
     </div>
</body>
</html>
```

Codice "mystyle.css"

```
.container
{		margin:0 auto;
		width:800px;
		text-align:center;
}
.box
{		width: 304px;
		margin: 0px auto;
		text-align: left;
		padding: 20px;
		background-color: #ffffff;
		color: #333;
		border: 1px solid #444444
}
input[type='text'],  input[type='password']
{		padding:5px;
		font-size:18px;
		border:1px solid #999999;
		width:300px;
		margin-bottom:10px;
		border: 1px solid #444444;
		background: #ffff00;
}
input[type='submit'], input[type='reset']{
		width:148px;
		font-weight: bold;
		padding: 12px 15px;
		background: #000080;
		color: #ffffff;
		font-size: 18px;
		border: 1px
}
h1{color:#F00000;}
```

Codice "controllacredenziali.php"

```
<?php
		session_start();
		$conn=mysqli_connect("localhost","root","mypassword","ticketsys");
		$e_mail=$_POST["e_mail"];
		$password=$_POST["password"];
		$ruolo=$_POST["ruolo"];
		$e_mail=mysqli_real_escape_string( $conn,$e_mail); //Vs MySql injection
		$s="select * from dirigenti where e_mail='$e_mail' and  password=md5('$password')";
		$q=mysqli_query($conn,$s);
		$numerorighe=mysqli_num_rows($q);
		if($numerorighe>0) {
			$_SESSION["autorizzato"] = "SI";
			$_SESSION["ruolo"] = $ruolo;
			header('Location:index.php');    //viene ridiretto alla pagina Home
		}
		else {
			header('Location:login.php');     // viene ridiretto alla pagina di login
		}
?>
```

Effettuato il login, la variabile di sessione "ruolo" viene utilizzata in ciascuna pagina per decidere cosa l'utente può fare e cosa non può fare. Anticipando un frammento del codice **"statistiche.php"** proposto di seguito, la variabile di sessione "ruolo" viene testata e se non vale "D", l'utente viene "respinto".

```
session_start();
…
if ($_SESSION["ruolo"]!="D")  {
      header('Location:index.php');// se l'utente non è un dirigente viene ridiretto alla pagina Home
   }
…
```

Codice "statistiche.php"

Esempio di codice Php (molto semplice e testato con pochissimi dati) che consente di accedere alla pagina "Statistiche interventi di assistenza", richiamata dalla pagina Home index.php, esclusivamente ai Dirigenti.

La pagina visualizza:

- il tempo medio di chiusura dei ticket completati nel 2020
- il numero e il tempo medio di chiusura dei ticket completati nel 2020 suddivisi per tecnico

```
<?php
   session_start();
   if (!isset($_SESSION["autorizzato"])) {
            // se l'utente tenta di accedere alla pagina senza autenticazione viene rediretto alla pagina di login
            header('Location:login.php');
    }
    if ($_SESSION["ruolo"]!="D")  {
            header('Location:index.php');// se l'utente non è un dirigente viene ridiretto alla pagina Home
    }
?>
<!doctype html>
<head>
    <meta charset="UTF-8">
    <meta http-equiv="Content-type" content="text/html; charset=UTF-8">
    <title>Statistiche interventi di assistenza</title>
    <link rel="stylesheet" href="mystyle2.css" rel="stylesheet">
 </head>
 <style>
       td{border:solid 1px}
 </style>
<body>
    <h1>InfoService TicketSys</h1>
    <?php
            $oggi=getdate();
            echo "Teramo ".$oggi["mday"]."-".$oggi["mon"] ."-".$oggi["year"];
     ?>
    <h2> Statistiche interventi di assistenza</h2>
    <?php
            /*
            Stampiamo ad esempio.
            1 il tempo medio di chiusura dei ticket completati nel 2020
            2 il numero e il tempo medio di chiusura dei ticket completati nel 2020 suddivisi per tecnico */
            $conn=mysqli_connect("localhost","root","mypassword","ticketsys");
            // Query statistica 1
            $s="create table temp as (
                        Select DATEDIFF(data_chiusura,data_apertura) as differenza
                        from ticket
                        where stato=1
                        and data_chiusura>='2020/01/01' and data_chiusura<='2020/12/31')";
```

```
            $q=mysqli_query($conn,$s);
            $s="Select ROUND(avg( differenza ),2) from temp";
            $q=mysqli_query($conn,$s);
            $r=mysqli_fetch_array($q);
            echo"Tempo medio di chiusura dei ticket completati nel 2020: ";
            echo  $r[0]."  giorni";
            $q=mysqli_query($conn,"DROP TABLE temp"); //elimino la tabella temp
            // Query statistica 2
            $s="select tecnici.nome,count(*),ROUND(avg(DATEDIFF(data_chiusura,data_apertura)),2)
                        from ticket inner join tecnici on id_tecnico=tecnici.id
                        where stato=1 and data_chiusura>='2020/01/01' and data_chiusura<='2020/12/31'
                        group by id_tecnico";
            $q=mysqli_query($conn,$s);
            ?>
            <br/><br/>Numero e  tempo medio di chiusura dei ticket completati nel 2020 suddivisi per tecnico<br/><br/>
            <table>
               <tr bgcolor="#FFFF00">
                  <td>Tecnico</td><td>Numero ticket completati</td><td>Tempo medio di chiusura in giorni</td>
             </tr>
            <?php
            while($r=mysqli_fetch_array($q))  {
                        echo"<tr><td>$r[0]</td>
                        <td style='text-align:center'>$r[1]</td >
                        <td style='text-align:center'>$r[2]</td></tr>";
            }
            echo"</table>";
            mysqli_close($conn);
            ?>
</body>
</html>
```

InfoService TicketSys

Teramo 6-3-2021

Statistiche interventi di assistenza

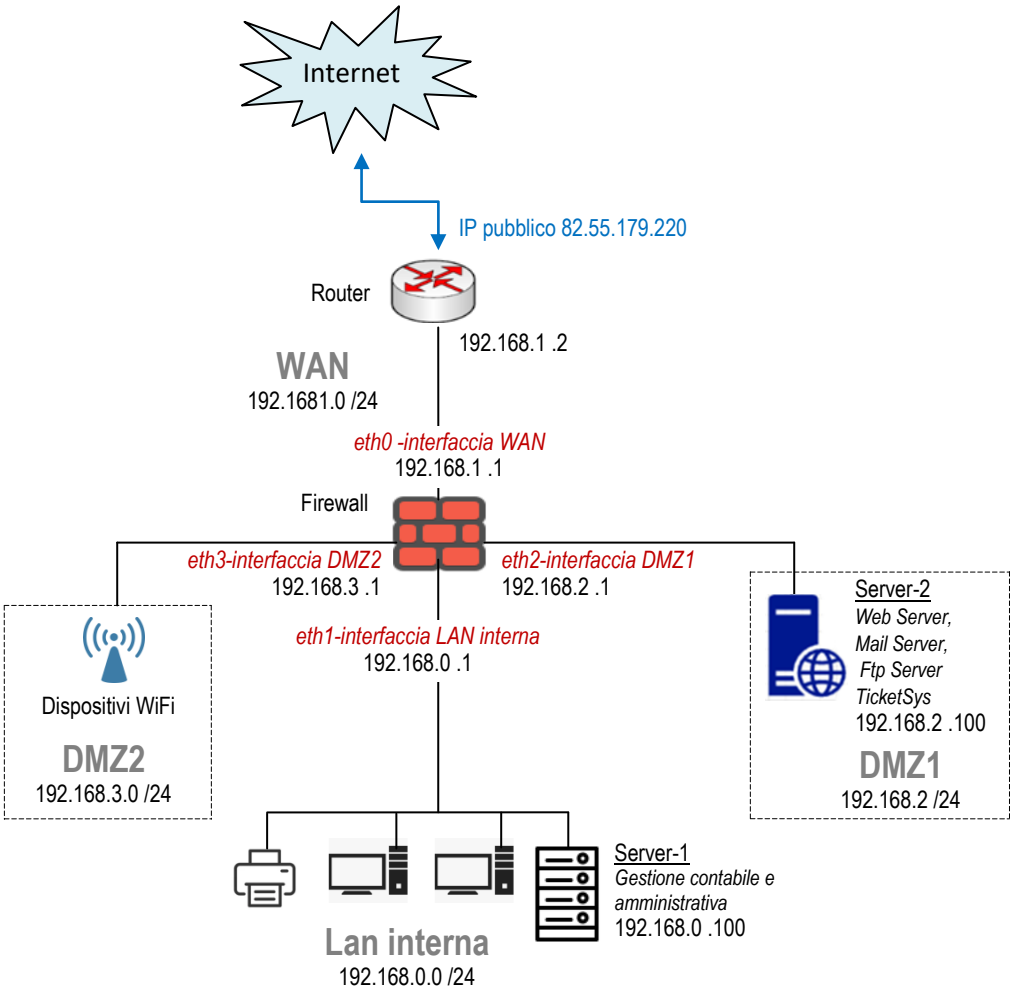
Tempo medio di chiusura dei ticket completati nel 2020: 2.60 giorni

Numero e tempo medio di chiusura dei ticket completati nel 2020 suddivisi per tecnico

Tecnico	Numero ticket completati	Tempo medio di chiusura in giorni
Paolo Rossi	1	3.00
Guido Lavespa	2	3.50
Remo Labarca	2	1.50

II. *In relazione al tema proposto nella prima parte, il candidato definisca il piano di indirizzamento della rete interna della sede principale di InfoService e le modalità con le quali viene controllato l'accesso di dispositivi wifi alla stessa. Approfondisca quindi i fattori che consentono di garantire la continuità del servizio dettagliando le risorse hardware e i servizi software che ritiene idonei per il caso in questione*.

La rete aziendale InfoService contiene un server pubblico e consente l'accesso dei dispositivi wifi.
La configurazione della rete atta a garantire la sicurezza, in particolare ad evitare qualsiasi possibilità che un un attacco esterno possa compromettere la Lan interna, si basa sull'utilizzo di un firewall con 4 interfacce:



rete aziendale InfoService con piano di indirizzamento

La rete aziendale InfoService è quindi segmentata con 4 reti/interfacce: WAN, LAN interna, DMZ1 e DMZ2.

La LAN interna, che contiene il Server-1 per la gestione contabile e amministrativa e le stazioni di lavoro del personale, è la rete più protetta. Da essa è possibile accedere ad Internet e alle due zone DMZ (in particolare i programmatori possono accedere in modo veloce al Server-2): non è invece assolutamente raggiungibile da altre reti e solo l'amministratore, dai server interni, può avviare una connessione in grado di accedervi.
Per le dimensioni dell'azienda (una cinquantina di dipendenti) è sufficiente una sottorete /24 con 254 host utilizzabili

DMZ1 ospita il Server-2 che è pubblico su Internet e consente a chiunque di accedere sulla porta TCP 80

Per la natura di InfoService e per le caratteristiche dei servizi aziendali offerti, la rete WiFi (realizzata con *Access point* con standard di conformità IEEE 802.11 a/b/g/n, banda di frequenza a 2,4 Ghz e 5 Ghz, sicurezza WEP, WPA, WPA2) è riservata ai visitatori e ai dipendenti e DMZ2 è progettata esclusivamente per fornire ai dispositivi wifi la connettività a Internet. Con tale modalità, per l'accesso al WiFi è sufficiente l'autenticazione con una wireless password condivisa.

Il firewall è configurato in modo tale che:

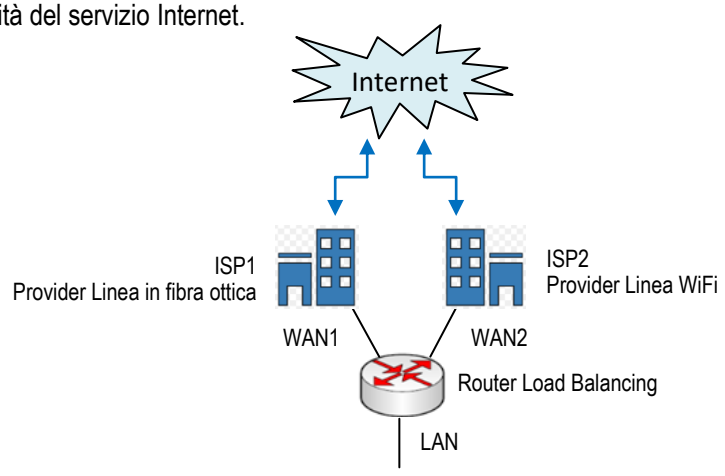
- la LAN interna può accedere ad Internet, DMZ1 e DMZ2
- DMZ1 può accedere a Internet ma non alla Lan interna
- DMZ2 può accedere a Internet ma non alla Lan interna
- fornisce l'accesso a Internet agli indirizzi IP, compressi nell'intervallo 192.168.3.0/24, che gli Access Point assegnano ai client wireless
- Il Web Server sul Server-2 (IP 192.168.2.100) è accessibile da Internet alla porta 80

Per quel che riguarda la continuità del servizio, la professionalità degli amministratori di rete e le misure di sicurezza adottate assicurano la migliore prevenzione dei rischi e minimizzano il danno in caso di eventi negativi.
In particolare:

- la continuità elettrica viene assicurata dal gruppo di continuità
- la sala server è climatizzata e protetta da intrusioni indesiderate e da eventi catastrofici quali incendi e allagamenti
- i server sono dotati di hardware ridondato (due alimentatori, dischi rigidi SATA raid 5) per far fronte a guasti o malfunzionamenti. I dischi sono Hot Swap per consentirne la sostituzione senza dover spegnere i server e interrompere il servizio
- la sicurezza logica è assicurata da antivirus, segmentazione della rete e firewall perimetrale
- il backup dei dati, indispensabile per la continuità operativa ed obbligatorio per legge (l'articolo 32 del GDPR recita: "*il titolare deve mettere in atto misure adeguate per garantire la sicurezza dei dati*"), è attuato giornalmente e in orari prestabiliti da un software che esegue il backup su hard disk esterno e su uno spazio cloud
- il piano di protezione dagli attacchi DDoS consente di contrastare eventuali attacchi massivi senza dover interrompere il normale svolgimento delle attività lavorative
  (Un attacco DDoS invia alla risorsa web aggredita una quantità di richieste tale da impedirgli, dopo un certo tempo, di funzionare correttamente)
- il disaster recovery plan prevede, in caso di perdita di dati e blocco delle attività, il ripristino di applicazioni e dati entro tempi brevissimi

Per garantire la continuità di Internet non è invece sufficiente affidarsi a un Provider di qualità.
Anche le tecnologie più sicure non sono esenti da rischi e problemi e si possono comunque verificare black-out e disservizi.

Nei casi in cui la connettività deve essere assicurata sempre (always-on) si può adottare la soluzione di utilizzare una seconda linea, preferibilmente con tecnologia di trasmissione differente dalla prima.
Nel nostro caso la linea principale in fibra ottica a 500Mb/s viene affiancata da una linea WiFi fornita da un altro Provider; le due linee convergono in un unico router multi-Wan con la funzionalità di "Load Balancing" che permette di condividere dinamicamete le due linee e di ottimizzare la connettività totale.
Con questa configurazione se "cade" la linea principale in fibra ottica, l'altra, seppure in modo meno performante, assicura la continuità del servizio Internet.



continuità del servizio Internet

III. *Lo sviluppo della rete Internet e l'incremento esponenziale del numero di dispositivi che si prevede verranno ad essa connessi, anche in conseguenza del forte impulso dato in tal senso dall'Internet delle cose (IoT), sta favorendo la diffusione del protocollo IPv6. Si espongano le caratteristiche del suddetto protocollo e le differenze rispetto al protocollo IPv4*.

Esistono due versioni del protocollo IP: la versione IPv4 e la versione IPv6.

Caratteristiche principali di IPV4

- la versione IPv4 è di gran lunga la versione più diffusa e trasporta oltre il 90% del traffico Internet attuale
- l'indirizzo IP è formato da 32 bit
- i 32 bit sono suddivisi in blocchi di 8 e la rappresentazione è decimale, con 4 cifre separate da un punto. In decimale il formato è XXX.XXX.XXX.XXX dove XXX è un numero compreso tra 0 e 255 (essendo 2^8= 256 le combinazioni che si possono ottenere con 8 bit)
  Esempio di indirizzo IP IpV4: *192.168.1.254*
- Il numero di indirizzi IPv4 disponibili è di 2^32 ovvero circa 4 miliardi

Gli indirizzi IPv4 disponibili sembrano tantissimi, in realtà, in conseguenza anche del numero sempre maggiore di dispostivi IoT che si connettono ogni giorno ad Internet, presto saranno assolutamente insufficienti.
Per questo motivo è nato lo standard IPv6, che in futuro sostituirà l'IPv4.

Caratteristiche principali di IPV6

- l'indirizzo IP è formato da 128 bit
- i 128 bit sono suddivisi in blocchi di 16 e la rappresentazione è esadecimale (8 cifre esadecimali separate da :)
  Esempio di indirizzo IP IpV6→ *A041:0C8D:96BC:1010:F0F0:7DD4:A24C:6276*
- il numero di indirizzi disponibili, pari a 2^128, è praticamente inesauribile
- grazie all'enorme spazio di indirizzamento, IPv6 consente una vera connessione end-to-end e rende superfluo il ricorso alla NAT (Network Address Translation) degli indirizzi IP privati in pubblici. In altre parole, ogni dispositivo collegato ad una LAN può essere indirizzato direttamente tramite il proprio IP privato
- utilizza IPSec (Internet Protocol Security), che consente di crittografare e autenticare i pacchetti inviati, rendendo la rete molto più sicura

IPv6, essendo un'evoluzione tecnologica del Protocollo IP, è decisamente migliore di IPv4 in termini di efficienza e sicurezza.

Purtroppo, però, le due versioni non sono compatibili e questo rallenta la diffusione del nuovo protocollo: la transizione allo standard futuro procede lentamente e potrà essere completata soltanto quando tutti i provider, server, sistemi e dispositivi utilizzeranno IPv6.

IV. Nell'interazione con un'applicazione web dinamica, l'utente compie azioni che richiedono l'invio di dati al server. Il candidato esamini i metodi attraverso cui è possibile trasferire al server i dati generati lato client dall'utente durante l'uso dell'applicazione, evidenziandone le specificità e i differenti usi. Fornisca al riguardo esempi di casi di utilizzo per le differenti modalità.

Un'applicazione web dinamica utilizza un modello di architettura client-server in cui un browser web lato client interagisce con l'applicazione lato server.

Un'applicazione web viene scritta utilizzando due tipi di linguaggio: i linguaggi lato client e i linguaggi lato server. In questo contesto, per generare i dati lato client, facciamo riferimento ad un form HTML (eventualmente utilizziamo anche CSS, JavaSript, JQuery, Ajax), e lato server utilizziamo il linguaggio PHP.

Quando un utente esegue il submit di un form, i dati vengono trasferiti al server attraverso una richiesta HTTP GET (metodo GET) o HTTP POST (metodo POST). Il server elabora la richiesta e fornisce la risposta, generalmente una pagina HTML o un file JSON o in altro formato.

PHP utilizza due variabili "superglobali" $_GET[] e $_POST[] per ricevere i dati provenienti dal client.

- $_GET[] è un array che contiene tutte le variabili che PHP ha ricevuto dal client tramite una richiesta GET
- $_POST[] è un array che contiene tutte le variabili che PHP ha ricevuto dal client tramite una richiesta POST

Esempio metodo GET

Codice lato client: prova1.html

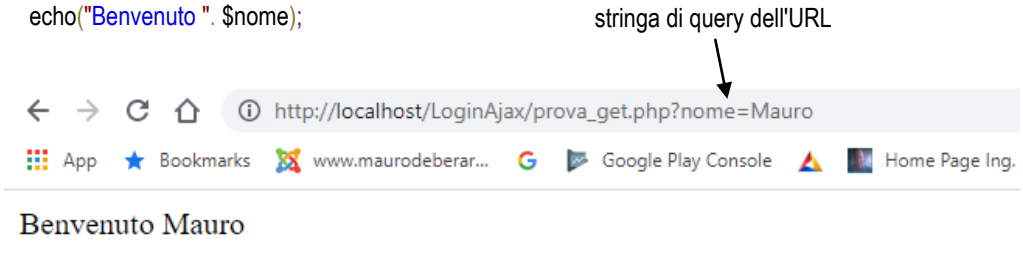
```
<!doctype html>
<html>
<head><title>Sistemi e Reti </title></head>
<body>
    <form action="prova_get.php" method="GET">
        <label>Inserisci il tuo nome:</label>
        <input type="text"  name="nome" />
        <input type="submit">
    </form>
</body>
</html>
```

La pagina contiene al suo interno un campo di input tipo text chiamato "nome" e un campo submit.
Il metodo scelto è GET (valore di default)

Se clicchiamo sul tasto di submit viene effettuata una richiesta HTTP GET al file "prova_get.php" e nella stringa di query dell'URL viene aggiunto un parametro con il valore inserito nel campo "nome".

Codice lato server: prova_get.php

```
<?php
    $nome=$_GET["nome"];
    if($nome=="") {
            echo("Non hai inserito alcun nome");
    }
    else {
            echo("Benvenuto ". $nome);
    }
?>
```

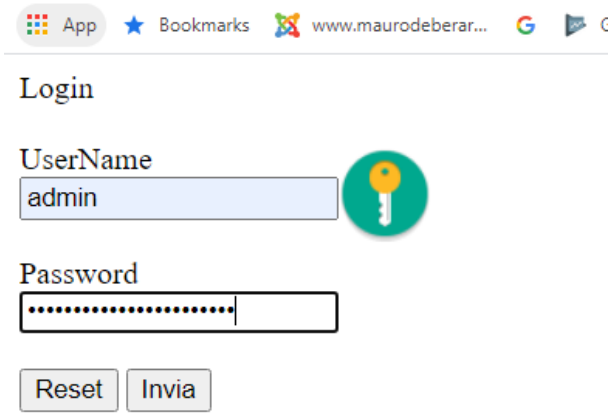


Esempio metodo POST

Codice lato client: prova2.html

```
<!doctype html>
<html>
<head><title>Sistemi e Reti </title></head>
<body>
      <form action="prova_post.php" method="POST">
            <label>Login</label></br><br/>
            <label>UserName</label><br/>
            <input type="text"  name="username" /><br/><br/>
            <label>Password</label><br/>
            <input type="password"  name="password" /><br/><br/>
            <input type="reset" value="Reset"> <input type="submit">
     </form>
</body>
</html>
```

La pagina contiene al suo interno un campo di input tipo text chiamato "username", un campo di input tipo password chiamato "password", un campo di reset e uno di submit.
Il metodo scelto è POST

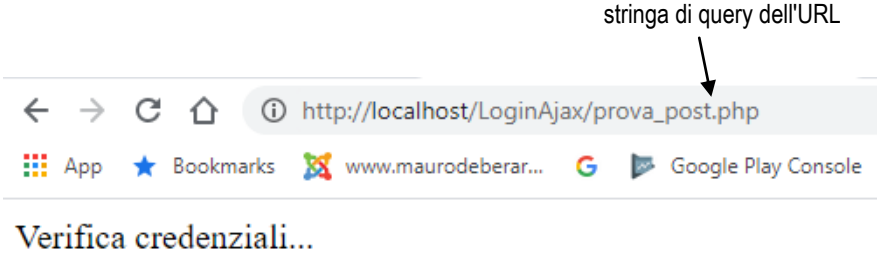


Se clicchiamo sul tasto di submit viene effettuata una richiesta HTTP POST al file "prova_post.php"

Con il metodo POST, a differenza del metodo GET, i parametri della richiesta non vengono aggiunti nella stringa di query dell'URL. Il metodo POST si utilizza generalmente quando si inviano al server dati personali o sensibili.

Codice lato server: prova_post.php

```
<?php
    $un=$_POST["username"];
    $pw=$_POST["password"];
    if($un==""  || $pw=="") {
            echo("Credenziali incomplete!");
    }
    else {
            echo("Verifica credenziali...");
    }
?>
```



Nei due semplicissimi esempi appena visti, i dati generati lato client vengono trasmessi al server in modo sincrono. Significa che l'utente compila un form e lo trasmette al server cliccando su "Invia": il server, in base ai dati ricevuti, risponde caricando una nuova pagina.

In alternativa, si va affermando sempre più la modalità asincrona di Ajax, una tecnica basata su JavaScript, che permette uno scambio di dati in background fra web browser e server.

Con AJAX, JavaScript inoltra una richiesta GET o POST al server, riceve ed elabora la risposta e aggiorna la pagina corrente.

L'utente non si accorge che qualcosa è stato trasmesso al server e l'applicazione risulta più fluida, veloce e interattiva.