Gestione aziendale --- Vetreria Giavenale

Brusarosco Matteo 5^C **2021**

Indice

- Introduzione……..1
- Prima parte……...2
  - Architettura di rete……..2
    - Topologia della rete……...3
    - Tabelle di routing……...4
    - Tabelle di routing delle sedi……..5
    - Configurazione del firewall……...5
    - VPN……..7
  - Comunicazione tra server e client……...7
  - Gestione della sicurezza……...8
  - Linguaggi web……9
  - Database.....10
    - Modello concettuale…..……..10
    - Modello logico…….……11
- Seconda parte……....12
  - Codifica di un segmento dell'applicazione WEB……12

Introduzione

La richiesta è quella di progettare una rete di gestione per un'azienda che vuole, tramite il sito web, esporre agli eventuali clienti tutti i prodotti disponibili alla vendita, e in quale sede è possibile trovarli. L'azienda richiede inoltre un sistema di comunicazione attraverso il quale può fornire ai clienti assistenza e supporto.
Si può quindi ipotizzare che l'azienda abbia la necessità di avere una rete locale LAN con almeno un server web e un mail server, due database(principale e backup) e infine vari spazi per tutti gli altri eventuali dispositivi (pc, stampanti, ecc) utili all'azienda stessa, possibilmente divisi per sezione.
Per ipotesi presuppongo che l'afflusso dei clienti sia abbastanza grande e la quantità dei dipendenti contenuta. Da queste ipotesi ne ricavo le dimensioni e la struttura della rete interna, del server web e le altre caratteristiche necessarie per gestire il carico di richieste.

L'azienda dispone di diverse sedi, disposte in varie città. Tutte le sedi hanno un software che si interfaccia con il database centrale (effettuando l'accesso con autorizzazioni adeguate).

Il software di accesso remoto al database della sede centrale permette di visualizzare tutto quello presente nel magazzino e di aggiungere o rimuovere i prodotti da parte del personale autorizzato.

Prima parte

1.1 Architettura di rete e caratteristiche dei sistemi server

Per la struttura della rete si possono fare alcune ipotesi:

1. Per erogare il servizio web è necessario almeno un server web dotato di ip pubblico e situato in una DMZ. All'interno della DMZ posiziono anche un mail server, che sarà poi utilizzato per le comunicazioni verso l'esterno dal dipartimento dedicato all'assistenza.
2. Per la storicizzazione dei dati è necessario disporre almeno 2 server con database, entrambi con un DBMS come MySql. Uno dei due database verrà poi utilizzato per eseguire un backup periodico (settimanale o giornaliero) del database principale.
3. Tutti gli altri dispositivi come i computer dei dipendenti, le stampanti, ecc, sono all'interno di una loro sottorete, protetta da un network firewall.
4. All'interno di ogni sezione, i dispositivi sono collegati via cavo allo switch disposto per tale sezione.

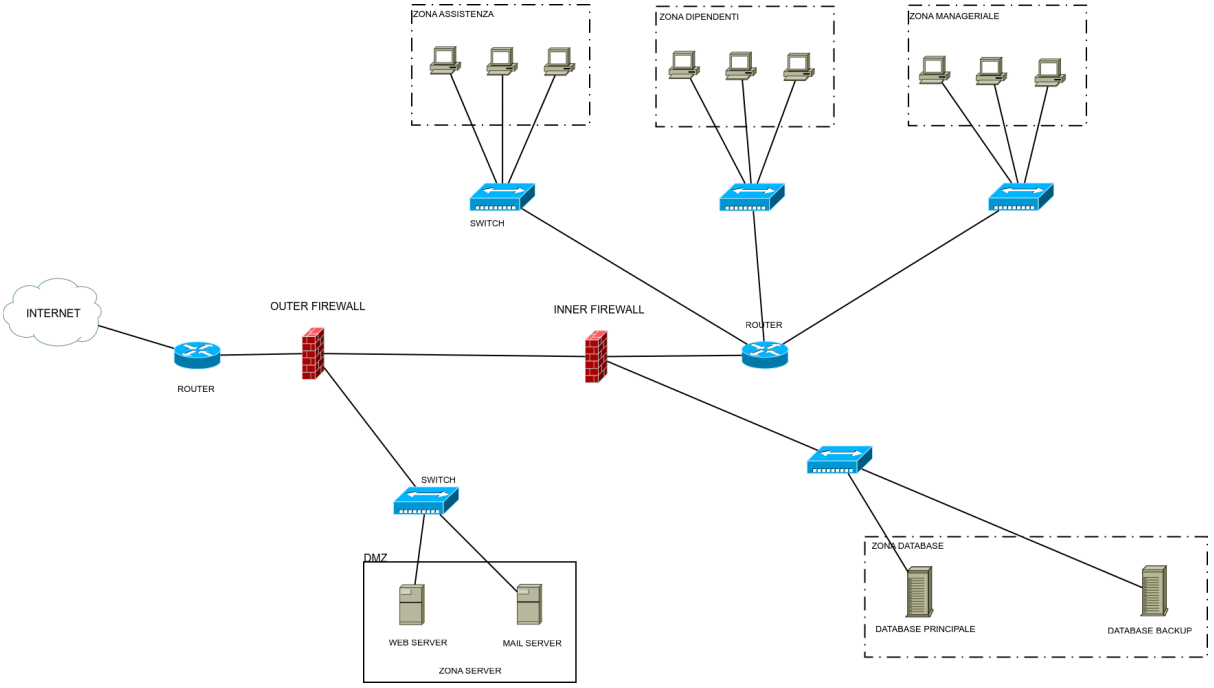
Nella struttura reale della rete, considerando anche l'organizzazione interna e le considerazioni formulate in precedenza, l'azienda può essere divisa in 4 sezioni:

1. **Sezione server**: comprende la DMZ nella quale sono posti i due server che andranno ad interagire con l'esterno cioè il web server e il mail server.
2. **Sezione assistenza**: comprende i dispositivi che andranno ad interfacciarsi con il mail server per comunicare con i clienti.
3. **Sezione dipendenti**: comprende i dispositivi che andranno ad interfacciarsi solamente con i macchinari e la gestione interna. Nessun accesso alla rete esterna.
4. **Sezione manageriale**: comprende i dispositivi degli amministratori aziendali. Ha un collegamento con la rete esterna.
5. **Sezione database**: comprende i database e permette solamente operazioni di lettura ai suddetti archivi. Eventuali modifiche alla base di dati possono essere eseguite dalla sezione dipendenti e manageriale.

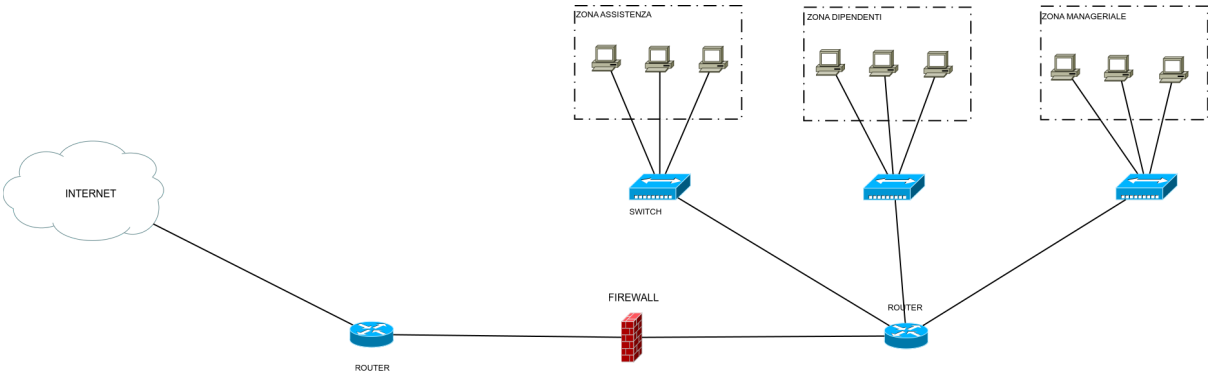
Questa soluzione prevede quindi la divisione della LAN aziendale in molteplici sottoreti, ognuna con le sue caratteristiche, in modo da separare i dispositivi da usare all'interno da quelli che possono interfacciarsi verso l'esterno e avere un maggiore controllo sulle sezioni interne grazie alle regole impostate nei firewall.

Topologia della rete

La soluzione della rete potrebbe quindi essere schematizzata con la figura seguente, dove viene utilizzato un router per la connessione esterna e un altro per le varie sottoreti interne.



La struttura della rete delle sedi, date le differenze strutturali, può essere rappresentata nel modo seguente:



Da notare che, data la mancanza della DMZ, è stato rimosso un firewall.

Tabelle di routing

Dato che l'azienda dispone di varie sedi, si ipotizza che il numero di dipendenti per sede non sia elevato. Si può quindi proporre una soluzione con sottoreti di classe C simile alla seguente:

Proposta con sottoreti di classe C router esterno	
Rete	**IP**
Self	192.168.1.254
Router interno	192.168.1.1
DMZ	192.168.200.0 /24
Sezione Database	192.168.100.0 /24

Proposta con sottoreti di classe C router interno	
Rete	**IP**
Self	192.168.1.1
Router esterno	192.168.1.254
Sezione Dipendenti	192.168.10.0 /24
Sezione Assistenza	192.168.20.0 /24
Sezione Manageriale	192.168.30.0 /24

Nel caso avessi bisogno di più ip (per future espansioni dell'azienda) potrei utilizzare una rete di classe B. In tal caso gli ip delle sottoreti diventerebbero:

Proposta con sottoreti di classe B	
Rete	**IP**
Sezione dipendenti	172.1.0.0 /16
Sezione assistenza	172.2.0.0 /16
Sezione manageriale	172.3.0.0 /16
Sezione Database	172.20.0.0 /16
DMZ	172.50.0.0 /16

Tabelle di routing delle sedi

Le tabelle di routing descritte sopra sono un esempio valido per la rete della sede centrale. Le sedi minori che poi andranno a collegarsi a quella principale presentano una struttura diversa e di conseguenza anche delle tabelle di routing differenti.

Ogni sede mantiene la struttura della sede principale, dalla quale differisce per la mancanza della DMZ, e quindi dei 2 server, e della sezione database.

La tabella di routing di una possibile sede minore, date le considerazioni sopra elencate, può essere:

Proposta con sottoreti di classe C sedi	
Rete	**IP**
Sezione dipendenti	192.168.11.0 /24
Sezione assistenza	192.168.21.0 /24
Sezione manageriale	192.168.31.0 /24

Configurazione dei firewall

Come precedentemente visto, per la struttura della rete si è deciso di utilizzare due firewall:

1. Il primo, posizionato verso l'esterno, protegge tutta la rete e ammette solo comunicazioni verso la DMZ. Questo firewall è di quarto livello cioè stateful (controllo attivo sulle connessioni in entrata ed uscita).
2. Il secondo, posizionato all'interno, ammette comunicazioni solo tra la LAN e la DMZ. Questo firewall è di tipo packet filter.

E' stato deciso di implementare due firewall per garantire un controllo più' granulare degli accessi alla rete e dividere il carico lavoro tra i due.

I firewall installati nella rete seguono le seguenti tabelle ACL. Le tabelle sono state create ipotizzando che:

- Vengano adeguatamente cambiate per le varie sedi, ognuna con i propri ip.
- Il web server ha ip 192.168.200.10, collegato alle porte 80 (http) e 443(https).
- Il mail server ha ip 192.168.200.20, collegato alle porte 143 (IMAP) , 993 (IMAP SSL) e 587(SMTP).
- Il database ha ip 192.168.100.2, collegato alla porta 3306 (default MySql).
- L'unica sezione della LAN che può accedere alla WAN è la sezione manageriale.

Tabella ACL per il firewall esterno:

Regole ACL firewall esterno					
Nr	Azione	IP sorgente	Porta sorgente	IP destinatario	Porta destinatario
1	allow	any	any/TCP	192.168.200.10	80/TCP
2	allow	any	any/TCP	192.168.200.10	443/TCP
3	allow	192.168.200.10	80/TCP	any	any/TCP
4	allow	192.168.200.10	443/TCP	any	any/TCP
5	allow	192.168.30.*	any	any	any
6	allow	192.168.200.20	587/TCP	any	any/TCP
7	allow	any	any/TCP	192.168.200.20	587/TCP
8	deny	any	any	any	any

Tabella ACL per il firewall Interno:

Regole ACL firewall interno					
Nr	Azione	IP sorgente	Porta sorgente	IP destinatario	Porta destinatario
1	allow	any	any/TCP	192.168.200.10	80/TCP
2	allow	any	any/TCP	192.168.200.10	443/TCP
3	allow	192.168.200.10	any/TCP	192.168.100.2	3306/TCP
4	allow	192.168.10.*	any/TCP	192.168.100.2	3306/TCP
5	allow	192.168.30.*	any/TCP	192.168.100.2	3306/TCP
6	allow	192.168.30.*	any/TCP	any	any
7	allow	192.168.20.*	any/TCP	192.168.200.20	993/TCP
8	allow	192.168.20.*	any/TCP	192.168.200.20	143/TCP
9	allow	192.168.20.*	any/TCP	192.168.200.20	587/TCP
10	deny	any	any	any	any

VPN

Per collegare le sedi sparse tra i vari comuni si è deciso di creare una VPN cioè una rete virtuale privata. La VPN permette di unire 2 reti private separate come fosse un'unica LAN e di fatto condividere i dispositivi di rete.

Con l'uso della VPN rendo possibile lo sharing del database tra le varie sedi, permettendo ad esse di sincronizzarsi, visualizzando i prodotti in magazzino e modificandoli (aggiungere e rimuovere).

Il tipo della VPN scelta è la Site-To-Site. Questo tipo di VPN viene generalmente instaurata tra router e router che assumono contemporaneamente il ruolo di client VPN e server VPN.

La VPN collega quindi il router esterno della sede principale con il router esterno delle altre sedi.

1.2 Modalità di comunicazione tra server e dispositivi, protocolli e servizi software per gestire la rete e fornire le pagine

Per quanto riguarda il software lato server e la gestione delle pagine web si può utilizzare XAMPP, una suite che comprende:

- Un DBMS free (MariaDB e SQLite).
- Web server Apache.
- Mail server Mercury.

Come precedentemente scelto, il web server e il mail server saranno posizionati su due macchine differenti e perciò nel caso ipotizzato non verrà utilizzato XAMPP ma direttamente i servizi necessari (Apache per il web server, Mercury per il mail server).

Se volessimo invece utilizzare un unico server e mantenere comunque tutte le funzionalità necessarie, si potrebbe installare la suite completa XAMPP ma, così facendo, si dovrebbero cambiare le tabelle ACL dei firewall.

Per il mail server si è scelto di utilizzare il protocollo **IMAP**, al posto del POP3, come protocollo di ricezione in quanto permette di effettuare procedure di sincronizzazione delle email oltre a fornire l'esecuzione di operazioni in parallelo (utile per l'assistenza dato che viene utilizzato 1 account per tutte le richieste ma viene usato da diverse persone in contemporanea).
Il protocollo di invio rimane invece l'**SMTP**, il protocollo standard per la trasmissione di email.

Il web server utilizza come protocollo di comunicazione l'**HTTP**, il protocollo usato come principale per il trasferimento di informazioni sul web nell'architettura client-server.

I protocolli di comunicazione scelti, impostati così, non offrono alcun livello di protezione e perciò, come vedremo successivamente, saranno implementati con un protocollo di sicurezza.

1.3 Gestione della sicurezza dei sistemi realizzati o utilizzati

La sicurezza è una delle parti fondamentali delle comunicazioni e nella gestione dei dati e perciò si è deciso di non tralasciare nessuna parte.

La sicurezza del sistema realizzato può quindi essere scomposta in 3 sezioni principali:

- **Sicurezza di rete**: Per prevenire attacchi dall'esterno sono stati inseriti nella rete 2 firewall: uno esterno e l'altro interno. Il firewall esterno permette l'accesso solo verso il web server dentro alla DMZ mentre quello interno solo connessioni in uscita verso la DMZ e la rete esterna. Le uniche persone che possono accedere all'esterno della rete sono i dirigenti, quindi le regole del firewall esterno e interno saranno impostate per accettare in uscita solo tutti gli IP appartenenti alla sezione amministrazione.

- **Sicurezza del database**: La sicurezza del database è incentrata sul preservare i dati da possibili problemi di natura casuale oppure intenzionale. E' buona norma includere una backup policy per recuperare i dati in caso di perdita, corruzione o rottura, per eventi esterni, del database. Per il backup e gia' stato predisposto un database secondario nella quale verranno copiati giornalmente, durante le ore notturne, i dati. Per quanto riguarda gli accessi non autorizzati al database c'è il bisogno di realizzare degli account specifici con i giusti permessi. Un altro livello di protezione nel database viene applicato nella memorizzazione delle password, di cui viene salvato l'hash. L'hash viene generato tramite una funzione di php e permette di non salvare le password in chiaro all'interno del database. Non salvare le password in chiaro torna utile nel caso ci sia una "fuga" di dati per qualche breccia, dato che i malintenzionati ottengono solamente una serie di caratteri e numeri casuali al posto della password vera.

- **Sicurezza nelle comunicazioni**: Per le comunicazioni viene utilizzato il TLS unito ai protocolli di comunicazione citati prima:

    - Per il web server HTTP over TLS, chiamato anche HTTPS.
    - Per il mail server IMAP over SSL.
    - SMTP over TLS.

SSL e TLS permettono una comunicazione sicura dalla sorgente al destinatario su reti TCP/IP fornendo autenticazione (cioè verificare l'identità del destinatario/mittente), integrità dei dati (verificare che i dati siano quelli originali e non manomessi da nessuno) e riservatezza (cifratura dei dati per garantire che solo il destinatario legga il contenuto).

Per quanto riguarda i backup, si è pensato di eseguire uno script, eseguito automaticamente tramite i crontab linux.
Lo script esegue un dump del database e lo importa nel database di backup ogni notte, cioè nel momento con meno affluenza (e meno interazioni con il db).

Un ulteriore livello di sicurezza è stato implementato nel sito web per prevenire tutti quegli attacchi conosciuti anche come SQL injection. Questo tipo di attacco informatico sfrutta il mancato controllo dell'input utente per inserire ed eseguire delle stringhe di codice SQL. Queste stringhe malevole possono portare ad un furto di informazioni o alla cancellazione del database. Per prevenire questo attacco è stato fatto un accurato controllo dell'input utente e sono state sfruttate le funzioni offerte da PHP per effettuare una sanificazione delle query dirette al database.

1.4 Linguaggi di programmazione dinamica per il web

Una pagina web dinamica è una pagina web il cui contenuto può variare in base a degli eventi o a input inseriti da parte dell'utente.
I linguaggi di programmazione che permettono di creare pagine web dinamiche possono essere divisi in 2 fondamentali categorie:

- **Client-side**: Il client, utilizzando il web browser come interprete, elabora gli script HTML inseriti nelle pagine. L'elaborazione lato client alleggerisce il server ma , dato che non permette di eseguire calcoli complessi e richieste verso un DBMS, viene impiegato solamente per modifiche all'interfaccia utente, calcoli semplici (leggeri) e lettura/scrittura dei cookies. Il linguaggio più' utilizzato client-side è Javascript.
- **Server-side**: Il server esegue tutte le elaborazioni del caso e invia al client la pagina finita e personalizzata in base all'input inserito. Le elaborazioni client side svolgono funzioni come:
    - Query e operazioni verso un database.
    - Leggere e scrivere file sul server.
    - Interagire con altri server.
    - Processare l'input utente.
    - Eseguire calcoli pesanti che potrebbero rallentare il client.

Tra i linguaggi server-side, quelli più' utilizzati per costruire pagine web dinamiche sono PHP e Java.

Nel web moderno le due categorie di linguaggi vengono utilizzate contemporaneamente, lasciando a javascript tutta la parte grafica e alcuni controlli sui dati, e a PHP/Java la parte di connessione al database ed esecuzione di algoritmi complessi.

Nella progettazione del sito web dell'azienda in questione, è stato deciso di utilizzare PHP per il server-side.
E' stato scelto PHP perché, essendo nato specialmente per il web, fornisce una vasta serie di funzioni che semplificano e velocizzano lo sviluppo delle pagine.

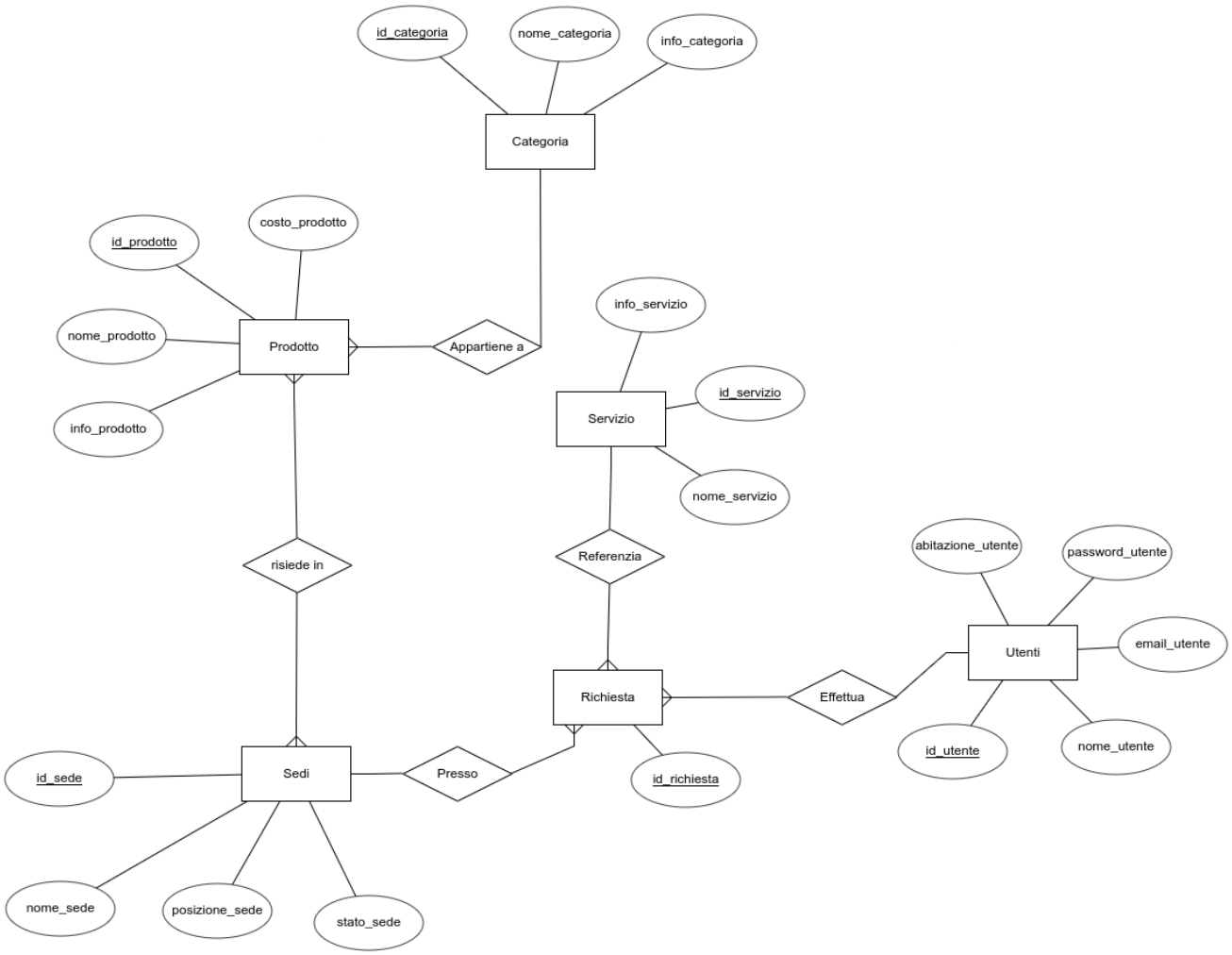
Lato client è stato invece scelto Javascript che, tramite il framework JQuery, permette di eseguire delle semplici animazioni e modifiche del DOM.

1.5 modello concettuale e logico del database

L'idea di fondo dello schema del database dovrebbe procedere in questo modo: un cliente può, tramite il sito web, richiedere un servizio (ad esempio consegna a domicilio, assistenza, ecc) ed ad una richiesta viene associata una sede di riferimento, che si occuperà del cliente. Nello stesso momento il cliente, sempre tramite il sito web, può visualizzare tutti i prodotti presenti nelle varie sedi. Del prodotto si può vedere di che categoria fa parte, in che sede è posizionato (anche più sedi contemporaneamente) e il suo costo.

La quantità legata al prodotto non viene visualizzata nel sito web ma è utile nella gestione del magazzino da parte delle sedi.

Modello concettuale (E/R):



Lo schema logico risulterà quindi come segue:

PRODOTTO (id_prodotto, nome_prodotto, info_prodotto, costo_prodotto, categoria)
PK = id_prodotto
FK = categoria riferito a **CATEGORIA**(id_categoria)

SEDE (id_sede, nome_sede, posizione_sede, stato_sede)
PK = id_sede

PRODOTTO_RISIEDE(id_collegamento, id_prodotto, id_sede)
PK = id_collegamento
FK = id_prodotto riferito a **PRODOTTO**(id_prodotto)
FK = id_sede riferito a **SEDE**(id_sede)

SERVIZIO (id_servizio, nome_servizio, info_servizio)
PK = id_servizio

UTENTE (id_utente, nome_utente, email_utente, password_utente, abitazione_utente)
PK = id_utente

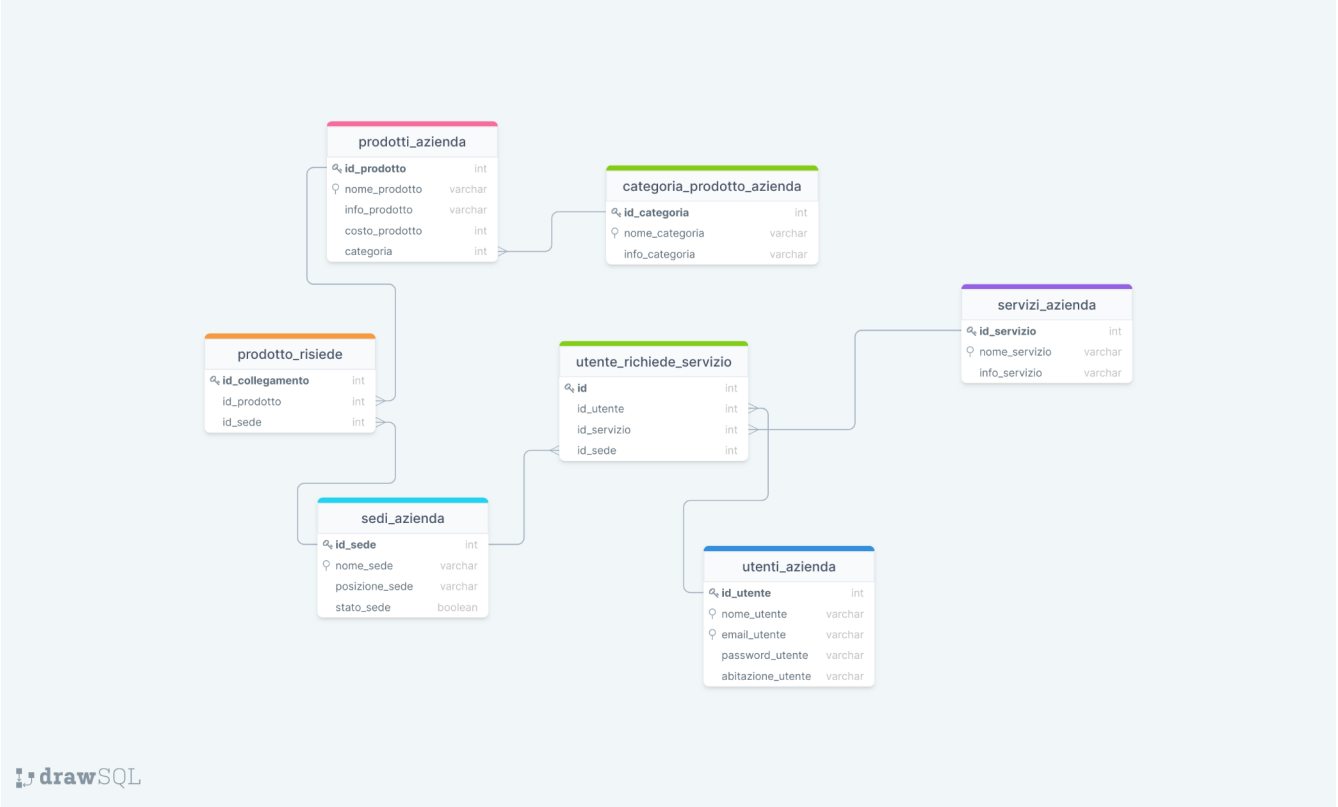
UTENTE_RICHIEDE_SERVIZIO (id_richiesta, id_utente, id_servizio, id_sede)
PK = id_richiesta
FK = id_utente riferito a **UTENTE**(id_utente)
FK = id_sede riferito a **SEDE**(id_sede)
FK = id_servizio riferito a **SERVIZIO**(id_servizio)

CATEGORIA (id_categoria, nome_categoria, info_categoria)
PK = id_categoria

Lo schema logico è stato creato partendo da alcune ipotesi:

- Data la relazione molti-a-molti delle tabelle **Prodotto** e **Sede,** è stata aggiunta la tabella **Prodotto_risiede** ipotizzando che un prodotto possa essere presente in più' sedi contemporaneamente.
- La tabella prodotto_risiede non ammette valori a NULL, quindi prima di creare un'istanza c'è bisogno di verificare l'esistenza del prodotto e della sede.
- La tabella utente_richiede_servizio non ammette valori a NULL, quindi c'è bisogno di verificare l'esistenza della sede, dell'utente e del servizio.

Modello logico:



Seconda parte

Codifica di un segmento dell'applicazione web che interagisca con la base di dati

Il sito web della realtà creata, dato l'uso di cui se ne vuole farne, esegue varie interazioni con il database.
Di seguito sono riportati dei brevi estratti di codifica in linguaggio PHP utili a soddisfare la richiesta.

Collegamento al database e funzioni per eseguire query:

```
<?php
        $conn = new mysqli("localhost", "username", "password", "db_name" );
        $statement = $conn->prepare("query");
        $statement->bind_param("tipoInput", $variabili);
        $statement->execute();
        $result = $statement->get_result();
        $conn->close();
?>
```

Login utente:

```
<?php
        if($_SESSION['nomeUtente'] == ""){
            if(isset($_POST['loginBtn'])){
                $nomeUtente = $_POST['nomeTxt'];
                $passw = $_POST['passwordTxt'];
                $conn = new mysqli($servername, $username, $password, $db_name);
                $query = "select * from utenti_azienda where nome_utente=?";
                $statement = $conn->prepare($query);
                $statement->bind_param("s", $nomeUtente);
                $statement->execute();
                $result = $statement->get_result();
                if($result->num_rows > 0){
                    $row = $result->fetch_assoc();
                    if(password_verify($passw, $row['password_utente'])){
                        $_SESSION['nomeUtente'] = $nomeUtente;
                        echo "<h1>Login effettuato con successo</h1>";
                        header("location: paginaPersonale.php");
                        exit();
                    }else{
                        echo "<h1>password errata</h1>";
                        echo "<p><a href='loginForm.php'>Clicca qui per riprovare</a></p>";
                    }
                }else{
                    echo "<h1>Nome utente errato</h1>";
                    echo "<p><a href='loginForm.php'>Clicca qui per riprovare</a></p>";
                }
                $conn->close();
            }
        }else{
            header("location: paginaPersonale.php");
            exit();
        }
    ?>
```

Visualizzazione prodotti presenti nelle sedi:

```
<?php
            echo "<h1>Barattoli</h1>";
            $conn = new mysqli($servername, $username, $password, $db_name);
            $query = "select nome_prodotto, info_prodotto, nome_sede, costo_prodotto from
                        prodotto_risiede
                        join prodotti_azienda pa on pa.id_prodotto = prodotto_risiede.prodotto
                        join categoria_prodotto_azienda cpa on cpa.id_categoria = pa.categoria
```

```
                join sedi_azienda sa on sa.id_sede = prodotto_risiede.sede
                where nome_categoria='Barattoli' group by nome_prodotto,
                nome_sede;";
        $statement = $conn->prepare($query);
        $statement->execute();
        $result = $statement->get_result();

        $tmp = "";
        while($row = $result->fetch_assoc()){
            if($row['nome_prodotto'] == $tmp){
                echo " ," . $row['nome_sede'];
            }else{
                echo "</p>";
                echo "<h3>" . $row['nome_prodotto'] . "</h3><p>" . $row['info_prodotto'] .
                    "</p>";
                echo "<p> Costo: <b>". $row['costo_prodotto'] ."€</b></p>";
                echo "<p><b>Disponibile presso:</b> " . $row['nome_sede'];

            }
            $tmp = $row['nome_prodotto'];
        }
        $conn->close();
    ?>
```

Inserimento nel database della richiesta di un servizio:

```
<?php
        if(isset($_SESSION['nomeUtente'])){
            //variabili di appoggio e inserimento nel DB
            $nomeUtente = $_SESSION['nomeUtente'];
            $servizio = $_POST['requestType'];
            $id_servizio = -1;
            $id_utente = -1;
            $id_sede = -1;

            $conn = new mysqli($servername, $username, $password, $db_name);
            $query = "select id_servizio from servizi_azienda where nome_servizio=?";

            //prendo id servzio
            $stmt = $conn->prepare($query);
            $stmt->bind_param("s", $servizio);
            $stmt->execute();
            $ris = $stmt->get_result();
            if($ris->num_rows > 0){
                $row = $ris->fetch_assoc();
                $id_servizio = $row['id_servizio'];
            }
```

```
        $query2 = "select id_utente from utenti_azienda where nome_utente=?";

        //prendo id utente
        $stmt = $conn->prepare($query2);
        $stmt->bind_param("s", $nomeUtente);
        $stmt->execute();
        $ris = $stmt->get_result();
        if($ris->num_rows > 0){
            $row = $ris->fetch_assoc();
            $id_utente = $row['id_utente'];
        }

        $query = "select id_sede from sedi_azienda where posizione_sede=?";
        $sede = $_POST['posizioneSl'];

        //prendo id sede
        $stmt = $conn->prepare($query);
        $stmt->bind_param("s", $sede);
        $stmt->execute();
        $ris = $stmt->get_result();

        if($ris->num_rows > 0){
            $row = $ris->fetch_assoc();
            $id_sede = $row['id_sede'];
        }

        $query = "insert into utente_richiede_servizio (utente, servizio, sede) VALUES
                    (?,?,?)";
        $stmt = $conn->prepare($query);
        $stmt->bind_param("sss", $id_utente, $id_servizio, $id_sede);
        $stmt->execute();

        $conn->close();

    } else{
        echo "<h1>Devi loggarti per visualizzare questa pagina</h1>";
    }
?>
```