

Elaborato per l'esame di maturità 2021

Indice

1. Considerazioni e ipotesi aggiuntive.....	2
2. Funzionalità previste.....	2

Richiesta n. 1 della traccia + richiesta di sistemi e reti

3. Infrastruttura	3
3.1 Scelta motivata dei dispositivi hardware e software.....	3
3.2 Ipotesi di configurazione con riferimento alla sicurezza	5
3.3 Descrizione dei protocolli utilizzati	7
4. Modalità di compilazione dei report	8

Richiesta n. 2 della traccia + richiesta di informatica

5. La base di dati	10
5.1 Schema E/R	10
5.2 Schema logico	11
5.3 Schema fisico	11
5.4 Query significative (richiesta n.3 della traccia).....	12
5.5 Implementazione linguaggio back-end (richiesta n.4 della traccia).....	13

Produzioni per altre materie

6. Produzione scritta di italiano	17
7. Produzione scritta di storia	19
8. Produzione scritta di inglese	20

1. Considerazioni e ipotesi aggiuntive:

- La InfoService è un'azienda di dimensioni medio-piccole, opera a livello regionale, di conseguenza è ragionevole pensare che possieda un numero abbastanza contenuto di clienti.
- Tali clienti potranno inviare richieste solo tramite il portale web di InfoService, previa registrazione e accesso. Non sarà possibile quindi richiedere assistenza da telefono o attraverso le email.
- I clienti potranno aprire più ticket nello stesso periodo di tempo (si pensi ad una piccola azienda che si affida a *InfoService* e che ha più problemi su macchine diverse nello stesso momento).
- Sullo stesso ticket potranno essere eseguiti più interventi ad opera di tecnici diversi.
- Ogni intervento dovrà riportare la data e l'ora in cui è stato effettuato.
- Il sistema dovrà inviare una notifica al cliente al momento dell'apertura e della chiusura del ticket.

2. Funzionalità previste:

1. Registrazione
2. Login cliente
3. Logout cliente
4. Login personale di InfoService
5. Logout personale personale di InfoService
6. Segnalazione e descrizione del problema da parte del cliente
7. Apertura del ticket da parte dell'addetto all'helpdesk + notifica al cliente
8. Assegnazione dell'intervento al tecnico
9. Inserimento report da parte del tecnico
10. Chiusura del ticket da parte del tecnico + notifica e invio del report al cliente
11. Convalida del report da parte del cliente
12. Possibilità per il cliente di esprimere un commento sul report del suo ticket.

3. Infrastruttura:

3.1. Scelta motivata dei dispositivi hardware e software

- **Sala server:**

Per l'infrastruttura di rete di *InfoService* si può pensare a due tipi di soluzione:

1. **On-site:** questa soluzione prevede l'allocazione di una sala server interna all'azienda. Tale sala server dovrà essere gestita dal personale di *InfoService*.
I vantaggi di questa soluzione sono:
 - gestione autonoma del server
 - intervento immediato in caso di problemi o malfunzionamenti/guasti
 - costi contenuti
2. **Esterne:** prevede di affidare ad un Data Center esterno la gestione delle apparecchiature e dei dati.
I vantaggi di questa soluzione sono:
 - sorveglianza, manutenzione e sicurezza garantiti da una ditta specializzata.
 - misure di sicurezza avanzate
 - responsabilità a carico della ditta esterna

Per una società come *InfoService*, tenendo conto della già presente preparazione in ambito informatico e sistemistico dei suoi dipendenti, del fatto che opera a livello regionale e con un numero di clienti e volume di dati abbastanza contenuto, è preferibile optare per una soluzione di tipo *on-site*.

Specifiche Hardware:

- Processore Intel Xeon W-2123, 3,6 GHz
- RAM Crucial Ballistix, 3200 MHz, 64GB DDR4
- SSD Sabrent 1TB + HDD Seagate Barracuda, 4TB
- Dissipatore Noctua NH-U9S
- Alimentatore TP-Link RPS150 ridondante
- Gruppo di continuità Tecnoware UPS ERA PLUS 1500

Software sui Server:

- O.S. Linux “ClearOS”
- Software del sistema di ticketing
- Php
- PhpMyAdmin
- MySQL
- Apache web server
- Exim Mail server
- FileZilla FTP server

- **Dispositivi in dotazione:**

Personale addetto all'helpdesk: in questo caso, dato che le macchine dovranno semplicemente restare collegate al portale di InfoService, ricevere le richieste, aprire i ticket e assegnare i tecnici, l'azienda può optare per una soluzione di fascia media:

- PC DELL 7010 SFF
- CPU Intel Core i5-3470
- Ram 8GB
- Hard Disk 500GB
- OS Windows 10 Pro



Tecnici di InfoService: tenendo conto del fatto che parte degli interventi verrà effettuata da remoto, i tecnici avranno bisogno di un dispositivo potente come un pc, ma facile da portare e da utilizzare come un tablet.

Un pc portatile convertibile è la soluzione ideale: in ufficio si può utilizzare come un classico computer portatile, mentre in trasferta lo si può piegare per utilizzarlo come un tablet sfruttando il suo schermo touchscreen.

- HP Pavilion X360
- Intel Core i5-1135G7
- RAM 16 GB
- SSD 1TB
- Scheda grafica Intel Iris Xe
- OS Windows 10 Home
- Schermo 14" Touchscreen



3.2 Ipotesi di configurazione con riferimento alla sicurezza: la DMZ

Per proteggere i server da eventuali attacchi si può implementare una DMZ (*delimitarized zone*).

Con il termine DMZ si intende una sottorete, separata dalla LAN, che contiene ed espone dei servizi ad una rete esterna non sicura (la WAN).

Quest'architettura rende molto più difficile un tentativo di attacco indirizzato alla LAN aziendale, dato che la DMZ che fornisce i servizi non è direttamente collegata alla LAN.

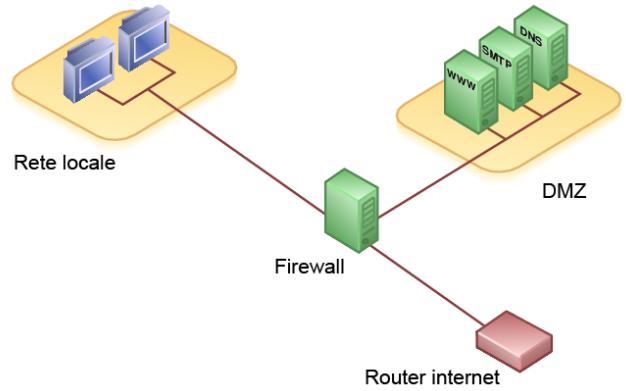
Nel caso in cui, ad esempio, un servizio LAN fosse compromesso e quindi vulnerabile, chi attacca potrebbe raggiungere tranquillamente gli altri host presenti sulla rete.

Se lo stesso problema si verificasse nella DMZ, raggiungere la LAN sarebbe molto più complicato per chi attacca, grazie alla presenza del firewall.

Le DMZ possono avere uno o due firewall

DMZ a singolo firewall:

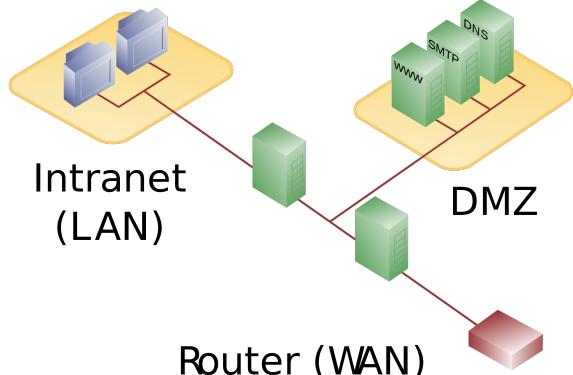
È la soluzione meno costosa ma allo stesso tempo la più carente in termini di sicurezza: avendo un solo firewall che collega la rete WAN esterna con la DMZ e la LAN, basta che questo presenti una falla per generare un rischio per l'intera rete.



DMZ a doppio firewall:

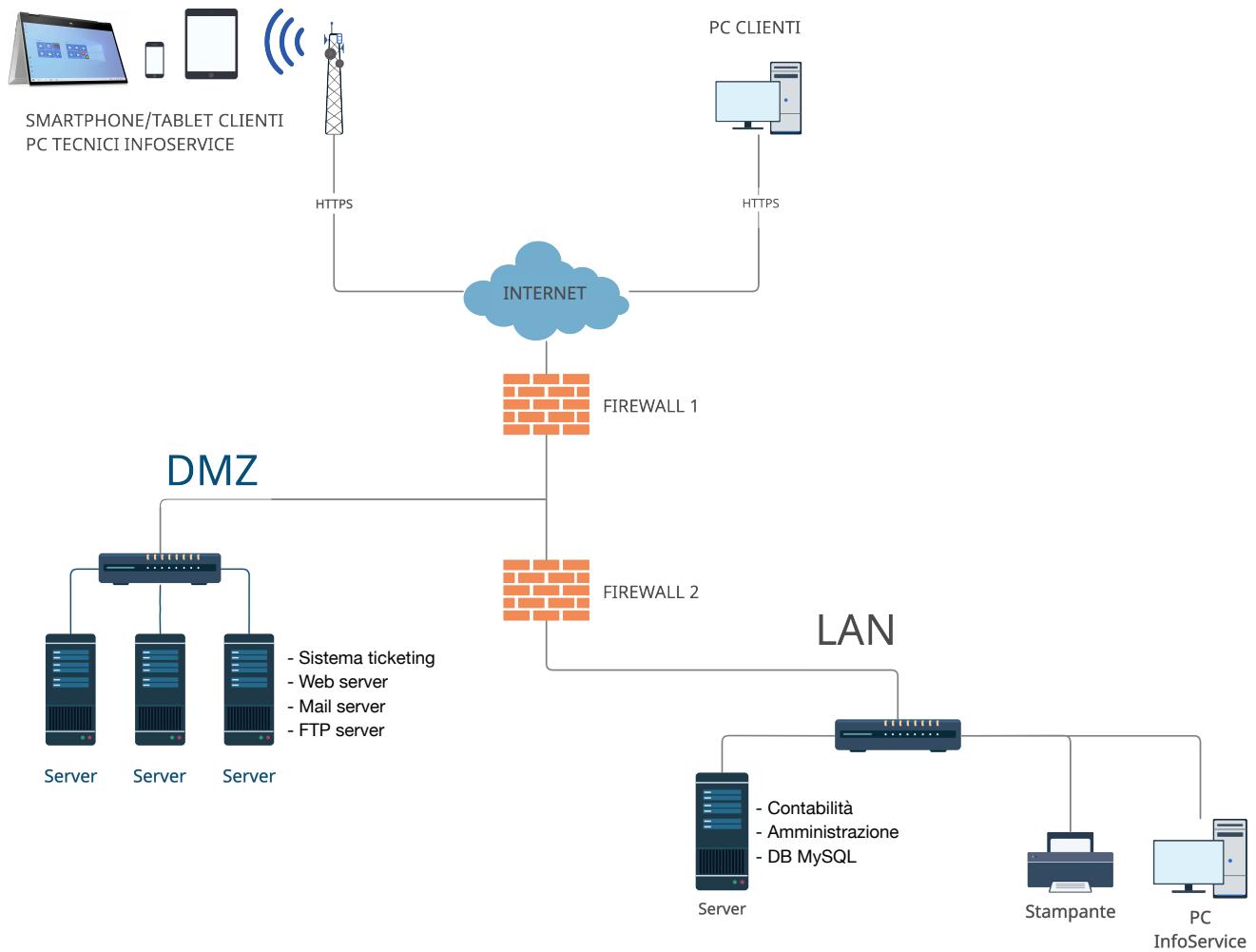
Questa architettura è più costosa ma offre maggior sicurezza: il firewall esterno protegge la DMZ della rete WAN, mentre quello interno separa la DMZ e dalla LAN aziendale.

È consigliabile utilizzare firewall di produttori differenti, altrimenti chi attacca potrebbe sfruttare una singola falla per violare entrambi i firewall.



Per la rete di InfoService prendiamo in considerazione l'architettura a **doppio firewall**, la sicurezza offerta dalla presenza di due firewall differenti è fondamentale.

Rappresentazione grafica dell'infrastruttura di rete di InfoService:



Questa architettura permette di configurare un percorso statico per controllare il traffico tra le reti, come nella seguente tabella:

Gli utenti si trovano...	Accesso alla DMZ	Accesso alla LAN	Accesso a Internet
... su Internet (WAN)	permesso	Non consentito	-
... nella LAN	permesso	-	permesso
... nella DMZ	-	Non consentito	Non consentito

3.3 Descrizione dei protocolli utilizzati:

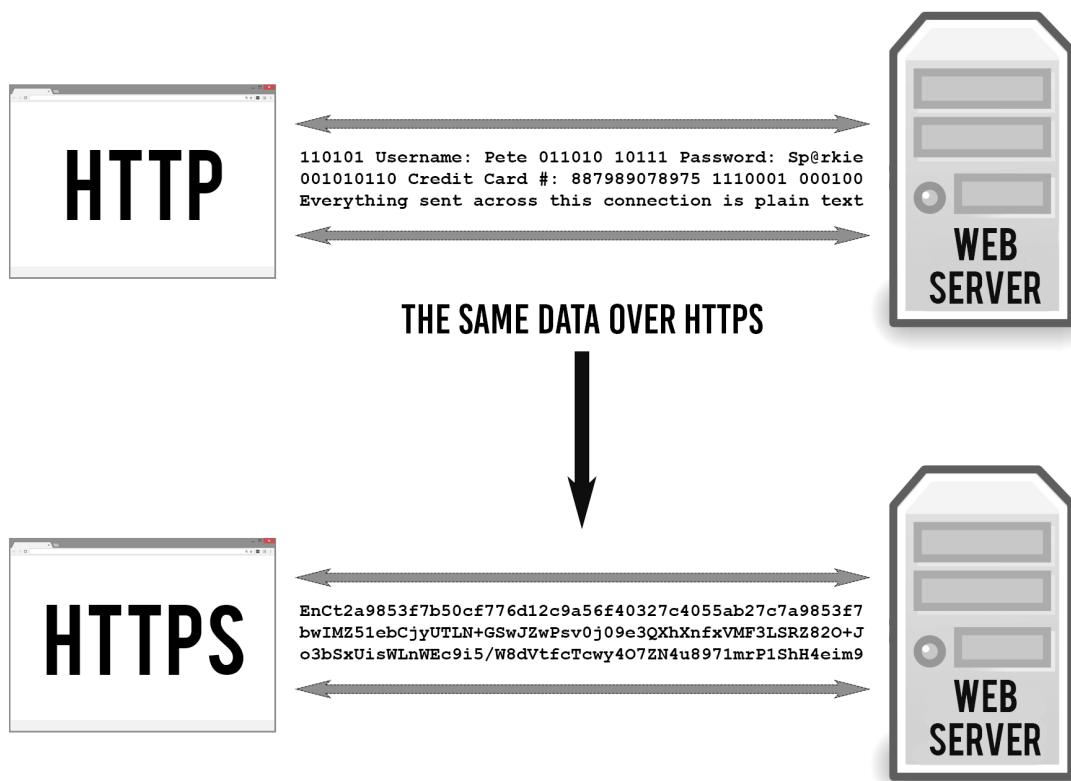
Per garantire la protezione dei dati durante le comunicazioni sulla rete InfoService si avrà del protocollo HTTPS (*Hyper Text Transfer Protocol Secure*), esso opera sul 7° livello della pila ISO/OSI (livello *application*), deriva dall'HTTP, ma a differenza sua, garantisce:

- **Autenticazione:** la sicurezza che il sito web visitato sia veramente quello richiesto;
- **Riservatezza:** la garanzia della protezione di dati (grazie alla crittografia solo il client e il server possono accedere ai dati);
- **Integrità:** la sicurezza che i dati arrivino integri a destinazione

Per farlo, si avvale di una comunicazione criptata, che può essere decriptata solo da chi possiede un certificato SSL valido.

In questo modo solamente il client e il server sono in grado di conoscere il contenuto della comunicazione.

Inoltre si impedisce ad altri soggetti di leggere e/o modificare i dati che vengono scambiati, come nel caso di un attacco “*man in the middle*”



Ecco la differenza tra una stringa di dati inviata con il protocollo HTTP e una inviata con l'HTTPS (e quindi criptata)

4. Modalità di compilazione del report

Il tecnico, dopo aver eseguito l'intervento, dovrà essere in grado di scrivere un report sul portale web di InfoService, descrivendo in che modo è intervenuto per risolvere il problema, se quest'ultimo è stato risolto, se necessita di ulteriori interventi oppure se si sono verificati problemi non previsti.

Per fare ciò, i tecnici avranno bisogno di una **funzionalità apposita**: le credenziali che gli sono state fornite al momento dell'assunzione in azienda, permettono di accedere ad un'**area riservata** del portale web di InfoService dotata di un'interfaccia differente rispetto a quella dei clienti, da cui il tecnico potrà controllare l'elenco dei ticket a cui è stato assegnato, redigere un report e chiudere temporaneamente il ticket, in attesa dell'approvazione definitiva del cliente.

Il report scritto dal tecnico, una volta inviato, arriverà sul database.

Schermata per la creazione di un report da parte del tecnico:

AGGIUNGI UN NUOVO REPORT:

Report

Tempo impiegato in minuti (min. 15 minuti)

Invia

Come il cliente vede i report relativi ai suoi ticket:

PIETRO@EXAMPLE.EX

LOGOUT

ID	DESCRIZIONE	STATO	DATA APERTURA	DATA CHIUSURA	CLIENTE
1	wifi non funzionante	Chiuso	2021-05-30 03:36:56	2021-05-31 21:47:58	1

ID REPORT: 1

DATA: 2021-05-30 03:43:14

REPORT: EFFETTUATO AGGIORNAMENTO SOFTWARE

TEMPO IMPIEGATO: 15

CONVALIDA: SI

COMMENTO:

ID REPORT: 3

DATA: 2021-05-30 23:13:56

REPORT: EFFETTUATA SCANSIONE ANTIVIRUS

TEMPO IMPIEGATO: 15

CONVALIDA: SI

COMMENTO:

5.0 - La base di dati

Il sistema di ticketing si basa su un'applicazione web, scritta in Php, e installata sul Server web on-site di InfoService.

I clienti e il personale dell'azienda (Dirigenti e Tecnici) possono accedervi con qualsiasi dispositivo, pc, tablet o smartphone, semplicemente utilizzando un browser.

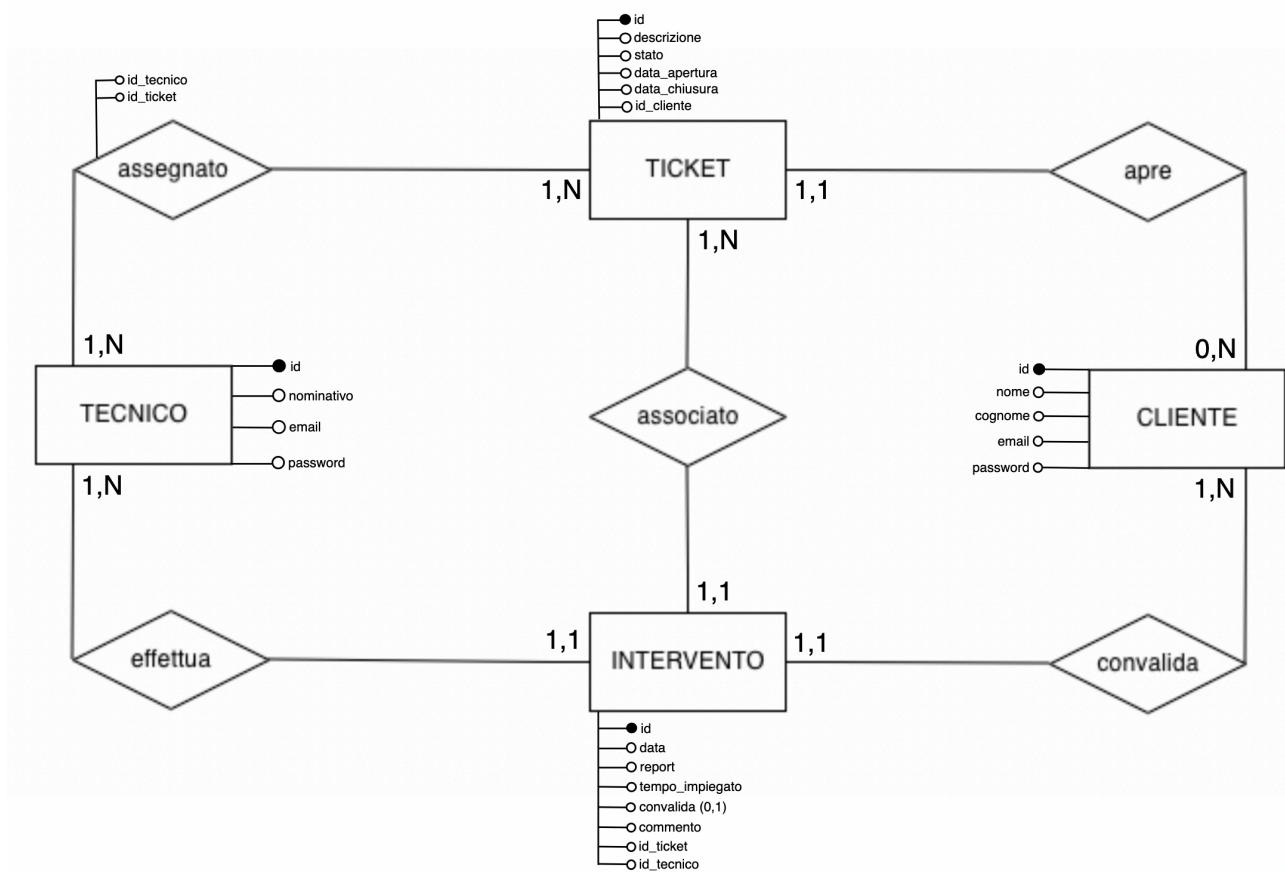
Il software offre funzionalità che non sono fruibili da tutti gli utenti: in base al ruolo di chi accede, infatti, il sistema abilita determinate funzionalità e ne disabilita altre.

Tutte le informazioni relative a clienti, tecnici, ticket ed interventi dovranno essere memorizzate su una base di dati, gestita da un DBMS (DataBase Management System)

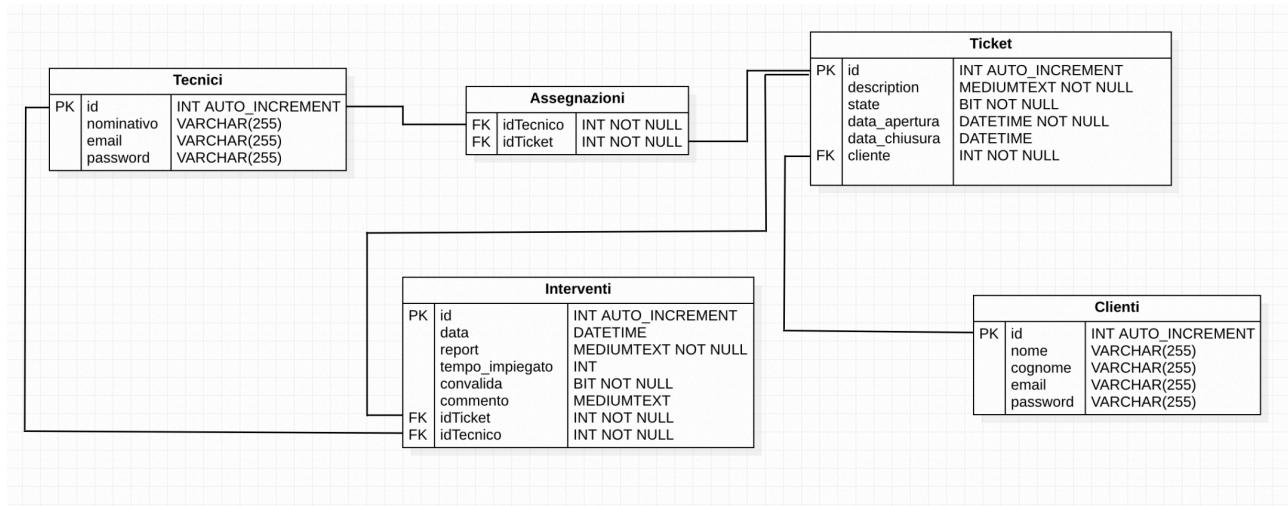
Il DBMS utilizzato da InfoService è MySQL.

Di seguito presenterò il modelli concettuali e logici che ho creato durante la progettazione di questo database.

5.1 - Schema E/R



5.2 - Schema logico



5.3 - Schema fisico

```

create database InfoService;
use InfoService;

DROP TABLE IF EXISTS clienti;
CREATE TABLE clienti
(
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(255) NOT NULL,
    cognome VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL,
    password VARCHAR(255) NOT NULL
);

DROP TABLE IF EXISTS tecnici;
CREATE TABLE tecnici
(
    id INT AUTO_INCREMENT PRIMARY KEY,
    nominativo VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL,
    password VARCHAR(255) NOT NULL
);

DROP TABLE IF EXISTS ticket;
CREATE TABLE ticket
(
    id INT AUTO_INCREMENT PRIMARY KEY,
    descrizione MEDIUMTEXT NOT NULL,
    stato BIT NOT NULL, /* 1 = aperto | 0 = chiuso */,
    data_apertura DATETIME NOT NULL, /* formato YYYY-MM-DD HH-MM-SS */
    data_chiusura DATETIME,
    id_cliente INT NOT NULL,
    FOREIGN KEY (id_cliente) REFERENCES clienti(id)
);

DROP TABLE IF EXISTS interventi;
CREATE TABLE interventi
(
    id INT AUTO_INCREMENT PRIMARY KEY,
    data DATETIME NOT NULL,
    report MEDIUMTEXT NOT NULL,
    tempo_impiegato INT NOT NULL, /* in minuti */,
    convalida BIT NOT NULL, /* 1 = convalidato | 0 = non convalidato */,
    commento MEDIUMTEXT,
    id_ticket INT NOT NULL,
    id_tecnico INT NOT NULL,
    FOREIGN KEY (id_ticket) REFERENCES ticket(id),
    FOREIGN KEY (id_tecnico) REFERENCES tecnici(id)
);

DROP TABLE IF EXISTS associazioni;
CREATE TABLE associazioni
(
    id_tecnico INT NOT NULL,
    id_ticket INT NOT NULL,
    CONSTRAINT id PRIMARY KEY (id_tecnico,id_ticket),
    FOREIGN KEY (id_tecnico) REFERENCES tecnici(id),
    FOREIGN KEY (id_ticket) REFERENCES ticket(id)
);

```

5.4 - Query significative (le due query richieste dalla traccia)

- Elenco dei ticket attualmente aperti riportando il nome del cliente che li ha aperti, la data di apertura, il tecnico che li sta seguendo

```
SELECT nome as "NOME", data_apertura as "DATA APERTURA", nominativo as "NOMINATIVO TECNICO"
FROM clienti,tecnicci,ticket,associazioni
WHERE stato = 1
AND tecnicci.id = associazioni.id_tecnico
AND associazioni.id_ticket = ticket.id
AND clienti.id = ticket.id_cliente;
```

- Tempo medio di chiusura dei ticket completati in un certo intervallo temporale fornito in ingresso

```
if(isset($_POST['calcoloMedia'])){
    try {
        $conn = $utils->dbConnect();
        $query = $conn->prepare("CREATE TABLE tmp AS
                                (SELECT DATEDIFF(data_chiusura, data_apertura) AS media
                                 FROM InfoService.ticket
                                 WHERE stato=0
                                 AND data_chiusura >= '2021-0".$_POST['mese']."-01 00:00:00'
                                 AND data_chiusura <= '2021-0".$_POST['mese']."-31 23:59:59');");
        $query->execute();
        $conn = null;

        $conn = $utils->dbConnect();
        $query = $conn->prepare(" SELECT ROUND(AVG( media ),1) FROM InfoService.tmp");
        $query->execute();
        $conn = null;

        $media = $query->fetchAll(PDO::FETCH_ASSOC);
    } catch (PDOException $e) {
        echo "Connection Failed: " . $e->getMessage();
    }
    echo "<p>Tempo medio di chiusura ticket nel mese n°".$_POST['mese']."' = ".$media[0]['mese']."'</p>";
}
```

La variabile in ingresso viene fornita tramite il metodo “post” dal form della pagina html

5.5 - Implementazione parte significativa del linguaggio lato back-end

Script per il login:

```
<?php
session_start();

include_once "Utils.php"; // il file in cui ho la classe php con le funzioni per la connessione al DB
$utils = new Utils();

if (isset($_POST['checker'])) {
    $username = $_POST['email'] ?? '';
    $password = $_POST['password'] ?? '';
    $match = false;

    if ($_POST['checker'] == "cliente") $table = "clienti"; // se l'utente accede come cliente
    if ($_POST['checker'] == "tecnico") $table = "tecnicici"; // se l'utente accede come tecnico

    if (empty($username) || empty($password)) {
        echo "Inserisci username e password";
    } else {

        try {
            $conn = $utils->dbConnect();
            $query = $conn->prepare("SELECT id, email, password FROM InfoService.$table.");
            $query->execute();
            $conn = null;

            $userlist = $query->fetchAll(PDO::FETCH_ASSOC);
        } catch (PDOException $e) {
            echo "Connection Failed: " . $e->getMessage();
        }

        foreach ($userlist as $row){ // controllo se username e password sono corretti
            if ($row['email'] == $username && $row['password'] == $password){
                $match = true;
                $_SESSION['user_id'] = $row['id']; // se sono corretti, assegno alla variabile superglobale $_SESSION l'id utente
            }
        }

        if ($match == false) {
            echo "Credenziali utente errate";
        } else {
            session_regenerate_id();
            $_SESSION['session_id'] = session_id();
            $_SESSION['session_user'] = $username;
            $_SESSION['type'] = $_POST['checker'];

            // se è un cliente lo reindirizzo alla dashboard dei clienti
            if($_SESSION['type'] == 'cliente') header('Location: dashboard-customer.php');

            if($_SESSION['type'] == 'tecnico') {
                try {
                    $conn = $utils->dbConnect();
                    $query = $conn->prepare("SELECT * FROM InfoService.tecnicici WHERE id = ".$_SESSION['user_id'].");
                    $query->execute();
                    $conn = null;

                    $user = $query->fetchAll(PDO::FETCH_ASSOC);
                } catch (PDOException $e) {
                    echo "Connection Failed: " . $e->getMessage();
                }

                if ($user[0]['nominativo'] == "direttore"){
                    $_SESSION['admin'] = true;
                    header('Location: dashboard-admin.php'); // se è un admin lo reindirizzo alla dashboard dell'admin
                }else{
                    header('Location: dashboard-employee.php'); // se è un tecnico lo reindirizzo alla dashboard dei tecnici
                }
            }
        }
    }
}
}
```

Form per il login:

```
● ● ●

<form action=<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?> method="post">

    <div id="login-form" class="login-form">

        <div class="title">
            <h1>Login</h1>
        </div>

        <label><input type="text" placeholder="email" name="email"></label>
        <label><input type="password" placeholder="password" name="password"></label>

        <div class="div-radio">
            <input type="radio" name="checker" value="cliente" checked>
            <label>Cliente</label><br>
            <input type="radio" name="checker" value="tecnico">
            <label>Personale</label><br>
        </div>

        <label><input class="sendButton" type="submit" name="submit" value="Accedi"></label>

    </div>
</form>
```

Script per visualizzazione dashboard del tecnico (elenco dei ticket a lui assegnati)

```
● ● ●

<?php

try {
    $conn = $utils->dbConnect();
    $query = $conn->prepare("SELECT *
                            FROM InfoService.ticket
                            WHERE id IN
                                (SELECT id_ticket
                                 FROM InfoService.associazioni
                                 WHERE id_tecnico = ".$_SESSION['user_id'].");
}

$query->execute();
$conn = null;
$ticketlist = $query->fetchAll(PDO::FETCH_ASSOC);
} catch (PDOException $e) {
    echo "Connection Failed: " . $e->getMessage();
}

echo "<h1 style='margin-bottom: 30px;'>Ticket che ti sono stati assegnati:</h1>";
echo "<div class='tab'>";
if(!empty($ticketlist)) {
    echo "<table>";
    echo "<tr>";
    echo "<th>ID</th>";
    echo "<th>DESCRIZIONE</th>";
    echo "<th>STATO</th>";
    echo "<th>DATA APERTURA</th>";
    echo "<th>DATA CHIUSURA</th>";
    echo "<th>CLIENTE</th>";
    echo "<th>VAI AI REPORT</th>";
    echo "</tr>";

    foreach ($ticketlist as $row) {
        echo "<tr>";
        echo "<td>" . $row['id'] . "</td>";
        echo "<td>" . $row['descrizione'] . "</td>";
        echo ($row['stato'] == 1) ? "<td>Aperto</td>" : "<td>Chiuso</td>";
        echo "<td>" . $row['data_apertura'] . "</td>";
        echo "<td>" . $row['data_chiusura'] . "</td>";
        echo "<td>" . $row['id_cliente'] . "</td>";
        echo "<td> <form action='dettaglioTicket.php' method='post'
                    <label><input type='hidden' name='id_ticket' value='".$row['id']."'>
                    <input class='sendButton reportButton' type='submit' name='dettaglioTicket' value='Dettaglio ticket'></label>
                </form></td>";
        echo "</tr>";
    }
    echo "</table>";
} else{
    echo "Ancora nessun ticket";
}
```

Script per l'apertura di un nuovo ticket da parte del cliente

```

● ● ●

session_start();

if($_SESSION['type'] != 'cliente'){
    session_destroy();
    header('Location: index.php');
}

include_once "Utils.php";
$utils = new Utils();

if (isset($_POST['newTicket'])){
    $conn = $utils->dbConnect();
    $query = "INSERT INTO InfoService.ticket (descrizione, stato, data_apertura, id_cliente) VALUES (?, 1, NOW(), ?);";
    $query = $conn->prepare($query);
    $query->bindParam(1, $_POST['newTicket']);
    $query->bindParam(2, $_SESSION['user_id']);
    $query->execute();

    if (!$query->rowCount() > 0) {
        echo "Abbiamo avuto un problema con l'inserimento dei dati. Verrai reindirizzato alla pagina di login...";
        header('Refresh: 2; url=index.php');
    }
}

```

Script per la creazione di un nuovo report da parte del tecnico

```

● ● ●

if($_SESSION['type'] == "tecnico") {
    echo "<div class='ticket-info'>";
    echo "<h1>Aggiungi un nuovo report:</h1>";
    echo "<form action='dettaglioTicket.php' method='post'>";
    echo "<label><textarea rows='10' placeholder='Report' name='newReport' cols='50'></textarea></label>";
    echo "<br><br><label>Tempo impiegato in minuti (min. 15 minuti)</label>";
    echo "<br><br><label><input type='number' name='minuti'></label>";
    echo "<br><br><label><input class='sendButton' style='width: 60%' type='submit' name='sendReport' value='Invia'></label>";
    echo "</form>";
    echo "</div>";
}

if (isset($_POST['newReport'])){

    $conn = $utils->dbConnect();
    $query = "INSERT INTO InfoService.interventi (data, report, tempo_impiegato, convalida, id_ticket, id_tecnico)
              VALUES (NOW(), ?, ?, ?, ?, ?)";
    $query = $conn->prepare($query);
    $query->bindParam(1, $_POST['newReport']);
    $query->bindParam(2, $_POST['minuti']);
    $query->bindParam(3, $_SESSION['id_ticket']);
    $query->bindParam(4, $_SESSION['user_id']);
    $query->execute();

    if (!$query->rowCount() > 0) {
        echo "Abbiamo avuto un problema con l'inserimento dei dati. Verrai reindirizzato alla dashboard...";
        header('Refresh: 2; url=dashboard-employee.php');
    }
}

```

Script per la convalida del report da parte del cliente e per chiudere il ticket se sono stati convalidati tutti i report relativi ad esso.

```

● ● ●

if (isset($_POST['convalidaReport'])){
    $chiudiTicket = null;
    try {
        $conn = $utils->dbConnect();
        // assegno '1' all'attributo 'convalida' del report (1 = convalidato / 0 = non convalidato)
        $query = "UPDATE InfoService.interventi SET convalida = 1, commento = ? WHERE id = ".$_POST['id_report'].";";
        $query = $conn->prepare($query);
        if(isset($_POST['commentoReport'])){
            $query->bindParam(1, $_POST['commentoReport']);
        }else{
            $query->bindParam(1, null);
        }

        $query->execute();
    } catch (PDOException $e) {
        echo "Connection Failed: " . $e->getMessage();
        echo "Abbiamo avuto un problema con la convalida del report. Verrai reindirizzato alla dashboard...";
        header('Refresh: 2; url=dashboard-customer.php');
    }
}

try {
    $conn = $utils->dbConnect();
    // seleziono tutte le convalide dei report associati a quel ticket
    $query = $conn->prepare("SELECT convalida FROM InfoService.interventi WHERE id_ticket = ".$_SESSION['id_ticket'].";");
    $query->execute();
    $conn = null;

    $convalidati = $query->fetchAll(PDO::FETCH_ASSOC);
    $chiudiTicket = true;
} catch (PDOException $e) {
    echo "Connection Failed: " . $e->getMessage();
}

foreach ($convalidati as $convalida){ // controllo se tutti i report del ticket sono stati convalidati
    if ($convalida == 0) $chiudiTicket = false;
}

if($chiudiTicket != null && $chiudiTicket == true){ // se si, chiudo il ticket
    try {
        $conn = $utils->dbConnect();
        $query = "UPDATE InfoService.ticket SET stato = 0, data_chiusura = NOW() WHERE id = ".$_SESSION['id_ticket'].";";
        $query = $conn->prepare($query);
        $query->execute();
    } catch (PDOException $e) {
        echo "Connection Failed: " . $e->getMessage();
        echo "Abbiamo avuto un problema con la convalida del report. Verrai reindirizzato alla dashboard...";
    }
}
}

```

6. Produzione scritta di Italiano

Sul tema del rapporto tra domanda e risposta, si presenti “la domanda di Matrimonio” di Svevo come una sequenza di query e di aperture di ticket (con reindirizzamento).

Se volessimo trasformare il racconto di Svevo in una sequenza di richieste e aperture di ticket, dovremmo innanzitutto chiarire quali sono gli attori in gioco:

- Zeno, il protagonista;
- Ada, la più bella delle quattro sorelle, promessa sposa di Guido, il rivale di Zeno;
- Alberta, la più piccola, se non si tiene conto della bambina
- Augusta, la più brutta delle quattro sorelle, ma anche la più dolce e, soprattutto, innamorata di Zeno.

Il primo avvenimento che può essere visto come una “richiesta di apertura ticket” è la domanda di Zeno ad Ada: il protagonista, approfittando del fatto che Anna, la più piccola delle sorelle, si era fatta male e che tutti i familiari, eccetto Ada, erano andati a soccorrerla, si avvicinò a lei e le dichiarò il suo amore. Lei rifiutò.

Si tratta quindi di una richiesta (apertura ticket) da parte di Zeno e un rifiuto (reindirizzamento) di Ada.

Zeno allora decise di fare un tentativo anche con Alberta, la più giovane tra le tre donne, ricordandosi ciò che gli diceva suo padre: “Scegli una donna giovine e ti sarà più facile di educarla a modo tuo”.

Cominciò a farle notare che essa si trovava “proprio nell’età di prendere marito”, ma rifiutò anche lei, spiegando che aveva interesse solo a continuare gli studi, con l’obiettivo di diventare una scrittrice.

Anche qui c’è una richiesta seguita da un rifiuto.

In un linguaggio di programmazione le **richieste** di Zeno possono essere viste come questa porzione di codice:

```
● ● ●
<div class="div-openTicket">
    <h1>Richiedi l'apertura di un nuovo ticket</h1>
    <form action="dashboard-customer.php" method="post">
        <label><textarea rows='3' placeholder='Descrizione del problema' name='newTicket' cols='50'></textarea></label>
        <label><input class="Button" type="submit" name="apriTicket" value="Invia"></label>
    </form>
</div>
```

```
● ● ●
$query = "INSERT INTO InfoService.ticket (descrizione, stato, data_apertura, id_cliente)VALUES (?,1,NOW(),?);";
```

Mentre i due rifiuti si possono tradurre in una richiesta non andata a buon fine, come in questo caso, dove oltre al “rifiuto” c’è il reindirizzamento dell’utente:

```
● ● ●

if($_SESSION['type'] != 'cliente'){
    session_destroy();
    header('Location: index.php');
}
```

Non potendo fare altro, dopo aver subito due rifiuti, Zeno si decise a proporsi ad Augusta: si mise faccia a faccia con lei e le chiese, in modo rude e diretto, se volesse che si sposassero. Le spiegò anche che poco prima aveva chiesto la mano sia ad Ada che ad Alberta, ma senza successo. Augusta, innamorata di Zeno, gli fece capire che aveva veramente intenzione di vivere per lui e di assisterlo, così i due si fidanzarono.

A differenza di prima, qui abbiamo un esito positivo alla richiesta e possiamo vederla come una vera e propria apertura di ticket, sempre con reindirizzamento, ma in questo caso verso la pagina contenente i dati del ticket.

Codice del form per inviare la richiesta di apertura del ticket:

```
● ● ●

<div class="div-openTicket">
    <h1>Richiedi l'apertura di un nuovo ticket</h1>
    <form action="dashboard-customer.php" method="post">
        <label><textarea rows='3' placeholder='Descrizione del problema' name='newTicket' cols='50'></textarea></label>
        <label><input class="Button" type="submit" name="apriTicket" value="Invia"></label>
    </form>
</div>
```

Codice contenente la query per l’apertura del ticket, con reindirizzamento sulla pagina dei ticket

```
● ● ●

if (isset($_POST['newTicket'])){
    $conn = $utils->dbConnect();
    $query = "INSERT INTO InfoService.ticket (descrizione, stato, data_apertura, id_cliente)VALUES (?,1,NOW(),?);";
    $query = $conn->prepare($query);
    $query->bindParam(1, $_POST['newTicket']);
    $query->bindParam(2, $_SESSION['user_id']);
    $query->execute();

    header('Location: ticket.php');
}
```

7. Produzione scritta di storia:

Sul tema dell'assistenza, parlare delle forme di welfare organizzate nei regimi dittatoriali.

La politica sociale ai tempi del fascismo:

Il regime fascista, nel ventennio che va dal 1920 al 1940 propose grandi progetti di riforma sociale.

Nel 1927 fu emanata la carta del lavoro: essa assegnava allo Stato importanti compiti riguardanti la previdenza, il collocamento e l'assistenza sociale.

Procediamo con ordine: parlando appunto di assicurazioni sociali, nel 1915 solo il 5% dei lavoratori italiani ne godeva, contro l'11% della Francia, il 36% dell'Inghilterra e il 43% della Germania. Gli sconvolgimenti della prima guerra mondiale attivarono però un processo riformatore il cui punto più alto fu raggiunto nel **1919** dal governo **Nitti**, con il **decreto legge** che rese obbligatorie le pensioni e le assicurazioni d'invalidità per tutti i lavoratori dipendenti. L'Italia fu trainata in questo da una tendenza internazionale: anche Inghilterra e Francia cominciavano a costruire i propri sistemi di welfare recuperando terreno sulla Germania, la nazione già avanti sotto questo punto di vista.

Nel 1922, il Consiglio dei ministri bloccò il progetto di accentrare le assicurazioni sociali nelle mani dello Stato, tutto ciò era finalizzato a conservare il favore di industriali e agrari dell'epoca, ma dal 1925 in poi il regime, preoccupato dalla svalutazione della lira, dovette in parte modificare le sue politiche economiche. Si cominciò a parlare di **collaborazione delle classi sociali per il fine comune della prosperità nazionale**.

Nel 1923 la riforma del 1919 sulle pensioni venne finalmente convertita in legge ma "depotenziata", escludendo dalle categorie tutelate mezzadri, coloni e impiegati pubblici. (È quindi falsa la storia secondo cui fu Mussolini a creare le pensioni, in realtà si limitò semplicemente a concludere un processo già avviato).

L'assistenza sociale come gestione del consenso:

Quasi tutte le politiche sociali fasciste (ma in generale di ogni regime dittoriale) sono finalizzate alla propaganda e a **garantire consenso al regime da parte di vari ceti sociali**, in base alle necessità del momento.

Fino al 1925 il regime si impegnò sistematicamente a depotenziare o smantellare gli enti assistenziali. Attraverso questa strategia i politici fascisti miravano a impossessarsi degli apparati di Stato e quindi indebolire il radicamento sociale delle opposizioni.

Durante il ventennio fascista vennero interrotte le inchieste condotte da commissioni di deputati e senatori sulle condizioni sociali del Paese.

Si mirava a deresponsabilizzare gli imprenditori per le pessime condizioni sanitarie delle fabbriche. Condizioni che peggiorarono notevolmente dopo lo scoppio della grande crisi economica del 1929, quando i sindacati di regime accettarono ritmi e ambienti di lavoro peggiori a patto di limitare i licenziamenti.

Una spinta verso il cambiamento:

Una parziale spinta in senso opposto provenne dalla pressione internazionale: tutti i maggiori Paesi stavano sviluppando sistemi di welfare sempre più avanzati.

Inoltre dopo la prima guerra mondiale era nata l'Organizzazione internazionale del lavoro, con lo scopo di promuovere e migliorare la tutela dei lavoratori nei vari Stati.

Il regime fascista accettò di introdurre nel 1929 l'assicurazione obbligatoria contro alcune delle malattie professionali più diffuse, anche se l'attuazione della legge venne di fatto ritardata al 1934.

Insomma, è chiaro che l'obiettivo del fascismo, così come di ogni altra dittatura, fosse la gestione del consenso e non la creazione di un sistema di assistenza sociale ben funzionante. L'assenza di elezioni libere non c'era bisogno di guadagnarsi supporto a breve termine e la crescita della burocrazia statale servì al tempo stesso a gestire la domanda politica.

8. Produzione scritta di inglese:

How can I prevent hacking of personal data?

The first thing to think about when we choose a method for protecting our data is the installation of antivirus. But this is not enough to protect data. For example we need to improve passwords, make them more secure and much more. It's very important to take care of the security of the data stored on our devices and on social networks.

1. Secure use of public Wi-Fi

Every day we check our bank accounts or make purchases online using public Wi-Fi. Posting some photo or information to Facebook, Twitter or Instagram in this way is very dangerous. It is important to connect only to HTTPS protocols (where "S" means "SECURE") and in no case to HTTP. Because when we communicate with someone on the web using HTTPS protocol our data are encrypted. In this way, data can only be read by us and our interlocutor. To be sure that we're using HTTPS protocol, we need to pay attention to the "lock" at the beginning of the link line, it indicates a secure connection.

2. Password protection:

The password must be at least 8 characters long, including uppercase and lowercase letters, numbers, and symbols. It seems difficult, especially to remember it, but is one of the most important thing that we have to do for protecting ourselves. Of course, you can add something else or change the characters, but try to make the password long and memorable. You can also use a password manager, google chrome have it. Obviously it's not recommended to use the same password for different accounts and sites.

3. Ignoring suspicious emails and messages:

Never open suspicious links in your emails or messages! The best would be to delete this type of messages without opening them. If you got an unusual message from your friend or relative, the best thing to do would be to contact them and ask if they sent you something before following the link.

4. Use anti-virus and anti-malware software:

It's recommend to install antivirus software and a firewall. It is important not to forget to update them. There are many type of those software on the web, but keep in mind the risks: downloads from suspicious websites are note recommended, we must download files and software only on the official ones.

A good antivirus will help identify dangers and protect our devices.

So we have to remind: taking care of the security of personal data is not just important, but necessary. The best protection in this case is to prevent attacks in advance. And the above methods can help us so much in this process.