

A.S. 2017-2018



Ministero dell'Istruzione dell'Università e della Ricerca

ITIA – ESAME DI STATO DI ISTRUZIONE SECONDARIA SUPERIORE

Indirizzo: ITIA - INFORMATICA E TELECOMUNICAZIONI

ARTICOLAZIONE INFORMATICA

Tema di: SISTEMI E RETI - *Tipologia C*

Il candidato (che potrà eventualmente avvalersi delle conoscenze e competenze maturate attraverso esperienze di alternanza scuola-lavoro, stage o formazione in azienda) svolga la prima parte della prova e due tra i quesiti proposti nella seconda parte.

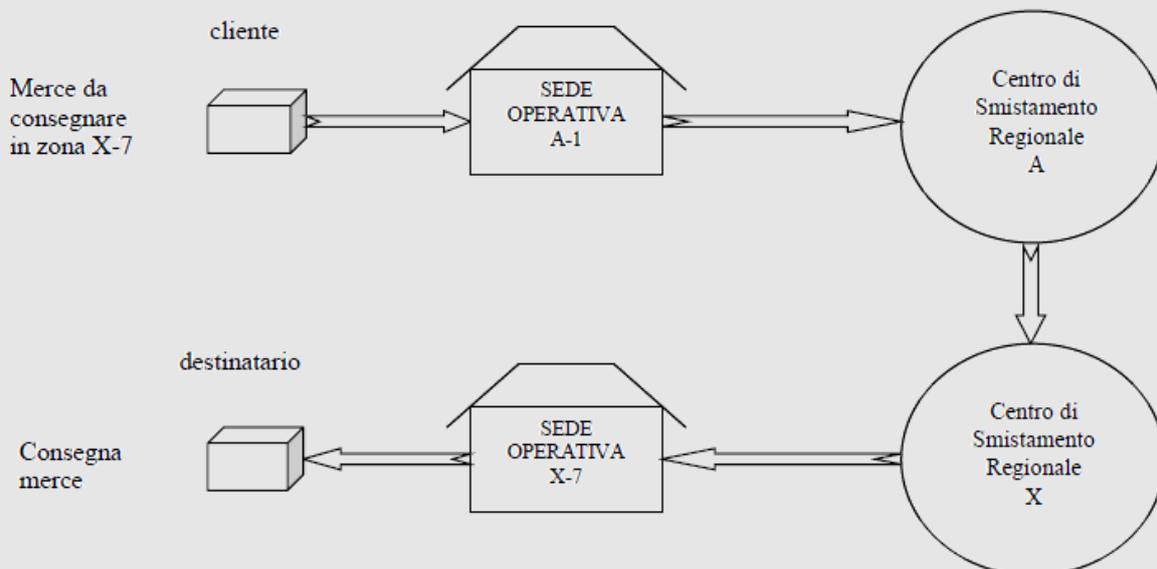
PRIMA PARTE

La società FastDelivery si occupa della spedizione di pacchi su tutto il territorio nazionale, per varie tipologie di clienti (privati cittadini, ditte, operatori di commercio elettronico, ...).

FastDelivery possiede proprie Sedi Operative (SO) in molte città italiane anche di piccole-medie dimensioni. In tali sedi, dotate di magazzino, gli addetti si occupano del ritiro dei pacchi da spedire che vengono loro affidati dai clienti, oltre che della consegna ai destinatari finali dei pacchi loro smistati dalle altre Sedi Operative di FastDelivery.

Ogni SO invia i pacchi da spedire al proprio Centro di Smistamento Regionale (CSR) di riferimento, e riceve dal CSR i pacchi da consegnare ai destinatari finali nella propria zona.

Il disegno illustra il percorso di un pacco tra due differenti regioni. Ad esempio, un cliente di Voghera (Lombardia) chiede di consegnare un pacco ad un destinatario di Barletta (Puglia): il pacco verrà preso in carico dalla SO di Voghera (A-1 nel disegno) che la inoltrerà al CSR di Milano (A nel disegno), che a sua volta lo inoltrerà al CSR di Bari (X nel disegno) che infine la inoltrerà alla SO di Barletta (X-7 nel disegno) per la consegna al domicilio del destinatario finale.



FastDelivery vuole automatizzare maggiormente il processo di raccolta, smistamento e consegna. In particolare, ai fini del tracciamento, vuole introdurre un sistema di identificazione dei pacchi che consenta all'azienda, ai clienti e ai destinatari di seguirne la posizione nel loro percorso fino a destinazione.

Quando un cliente (mittente) vuole spedire un pacco, effettua una richiesta on line: dalla SO più vicina, un trasportatore, recandosi presso il cliente per il ritiro, trasmette al sistema la presa in carico della spedizione. Un apposito algoritmo, già esistente, provvede a definire il percorso per la consegna di ciascun pacco. In ciascun passaggio del trasporto, il pacco viene prima preso in carico in ingresso e poi tracciato in uscita dai magazzinieri della SO o del CSR che gestisce il pacco stesso. Il trasportatore che porta a termine una consegna raccoglie la firma del destinatario e la trasmette al sistema di tracciamento.

Il candidato, fatte le opportune ipotesi aggiuntive:

1. Ipotizzi come potrà essere organizzata operativamente la nuova procedura di gestione informatizzata dei pacchi (acquisizione dei dati di mittente e destinatario, presa in carico dal mittente, metodi di identificazione e procedure operative di tracciamento in ciascuna SO e CSR fino alla consegna, tipologia delle informazioni raccolte, rilevamento dell'avvenuta consegna).
2. Illustri il progetto dell'infrastruttura informatica necessaria per realizzare la gestione automatizzata dei pacchi e consentirne la tracciabilità, dettagliando:
 - a. dispositivi utilizzati da trasportatori e magazzinieri per lo svolgimento delle proprie attività;
 - b. modalità di comunicazione tra i sistemi;
 - c. organizzazione dei server di raccolta dati ed offerta dei servizi informativi; si sviluppino e discutano due o più ipotesi alternative, di cui una totalmente interna all'azienda ed una che contempli anche il ricorso a servizi Cloud, scegliendone una motivatamente.
3. Approfondisca gli aspetti legati alla sicurezza delle strumentazioni, dei dati gestiti e del servizio offerto nel caso in esame, e discuta le misure che ritiene utili per garantire la continuità del servizio (aspetti di business continuity e fault tolerance).

SECONDA PARTE

- I. In relazione al tema proposto nella prima parte, si vuole permettere ad un utente di conoscere l'attuale stato della spedizione (l'elenco di SO o CSR da cui è già transitato il pacco e l'eventuale stato di avvenuta consegna). Il candidato progetti lo schema concettuale ed il modello logico della porzione di base di dati necessaria; progetti poi le pagine web che consentono di ottenere le informazioni richieste, scrivendo in un linguaggio a scelta il codice di una parte significativa.
- II. In relazione al tema proposto nella prima parte, la società FastDelivery è interessata anche a poter monitorare gli spostamenti dei propri automezzi sulla strada in tempo reale. Il candidato illustri quali potrebbero essere le soluzioni tecnologiche disponibili e le modalità e i protocolli utilizzati nella comunicazione tra automezzi e centrale operativa.
- III. Le sfide poste dalla necessità di assicurare in qualsiasi momento l'accessibilità dei dati agli utenti autorizzati hanno portato allo sviluppo di metodologie di gestione note come clusterizzazione delle risorse hardware e virtualizzazione delle risorse software. Il candidato illustri in cosa consistono queste metodologie ed analizzi vantaggi e svantaggi di ciascuna, anche con esemplificazioni applicative.
- IV. Le comunicazioni via email spesso necessitano dell'applicazione di specifiche precauzioni per la sicurezza. Si descrivano le possibili minacce alle comunicazioni via email e i principali protocolli e servizi per garantire la loro sicurezza.

Durata massima della prova: 6 ore.

È consentito soltanto l'uso di manuali tecnici e di calcolatrici tascabili non programmabili.

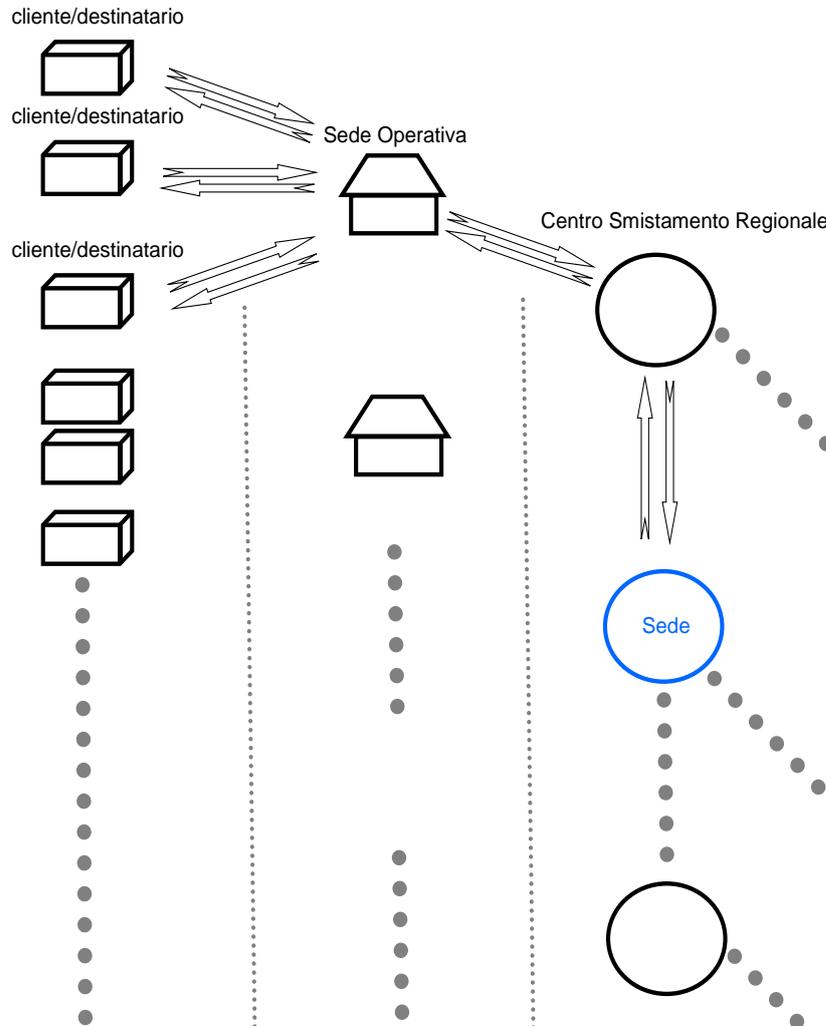
È consentito l'uso del dizionario bilingue (italiano-lingua del paese di provenienza) per i candidati di madrelingua non italiana.

Non è consentito lasciare l'Istituto prima che siano trascorse 3 ore dalla dettatura del tema.

SOLUZIONE

Il testo non necessita di particolari ipotesi aggiuntive o di precisazioni essenziali. Piuttosto, lo schema proposto andrebbe generalizzato, dato che la realtà descritta comprende vari clienti/destinatari associabili a vari **Sedi Operative (SO)**, che sono varie per ogni **Centro di Smistamento Regionale (CSR)**, i quali sono vari sul territorio nazionale:

Schema generalizzato



In particolare, si individua un Centro di Smistamento Regionale da identificare come **Sede** dell'Azienda FastDelivery, all'interno della quale collocare i server che dovranno sovrintendere alla gestione dati della nuova gestione informatizzata, con una gestione completamente via Web.

NB.

Si tralascia una soluzione alternativa basata su VPN e che aggregi tutti le SO e tutti i CSR (LAN to LAN) e consenta agli operatori di accedere in sicurezza (host to LAN).

PRIMA PARTE Quesito 1

I dati del **cliente/mittente** (anagrafica, pacco, anagrafica destinatario e altro) verranno acquisiti dall'**Azienda** tramite un **portale Web** pubblico a cui, previa registrazione e accesso, il cliente/mittente li compila e li invia. La parte server del portale risiede su una macchina collocata nella LAN del CSR denominato **Sede** (come in figura).

1. Il sistema informatizzato prende in carico l'ordine, decide quale SO è coinvolto tramite algoritmo (già presente, come indicato nel testo), incarica un **trasportatore** che raggiunge il cliente/mittente.
2. Il trasportatore è dotato di uno smartphone e di una **App dedicata**, la consulta e individua e verifica l'ordine, ritira il pacco, **emette** una **targhetta adesiva** riepilogativa (codice a barre o QR), la applica sul pacco e registra quindi l'**acquisizione** dell'ordine.
3. Il trasportatore, giunto al SO, cede il pacco al **magazziniere**.
4. Il magazziniere è dotato di uno smartphone e di una **App dedicata**, **legge** la targhetta sul pacco segnalando al sistema informatizzato la prima tappa.
5. Il magazziniere **emette** una **targhetta adesiva** riepilogativa (codice a barre o QR), la applica sul pacco e registra quindi l'**acquisizione** dell'ordine.
6. Ora un trasportatore prende in carico il pacco **leggendo** la targhetta e ne **emette** un'altra che indica il trasporto **tra SO e CSR**.
7. La procedura prosegue fino al raggiungimento del destinatario: il trasportatore finale **chiuderà** il trasporto facendo firmare sull'App il destinatario (ovvero rientrerà alla SO per ritentare la consegna in seguito).

I dati gestiti dal sistema informatizzato descritto sono di tipo standard (caratteri, stringhe, interi, ecc...), tranne la firma del destinatario (una bitmap).

Eventualmente si potrebbe prevedere la firma anche del cliente/mittente per autenticarne l'ordine.

PRIMA PARTE Quesito 2

L'infrastruttura per il sistema informatizzato consta:

- **LAN** connessa alla rete pubblica per ogni SO.
- **LAN** connessa alla rete pubblica per ogni CSR
- **Server pubblico** (es., Web server) e **database server** collocati nel **CSR** denominato **Sede**
- **Smartphone** con connessione dati (dati mobili su Internet), ovvero connessione WiFi con a bordo una **App** client (es., client HTTP) dedicata in dotazione ad ogni **trasportatore** e **magazziniere**
- Stampante portatile per smartphone (es., in tecnologia Bluetooth) in dotazione ad ogni **trasportatore** e **magazziniere**

In alternativa allo smartphone si potrebbe prevedere un PDA proprietario, dotato di connessione pubblica, input a telecamera e stampante integrata.

Le modalità di comunicazione coinvolte:

| | |
|-----------------------------|----------------------------------------------------------|
| cliente/mittente – Azienda, | tramite Web, protocollo HTTP/HTTPS |
| trasportatore – Azienda, | tramite App, protocollo HTTP/HTTPS |
| magazziniere – Azienda, | tramite App, via WiFi e Web, protocolli WPA2, HTTP/HTTPS |

La parte server della gestione dei servizi (server Web) e dei dati (database server) può avvenire completamente all'interno dell'Azienda (nel **CSR** denominato **Sede**) oppure delegando entrambi in outsourcing via Cloud (di tipo hosting) ovvero via Cloud (di tipo housing).

Nel primo caso si hanno alcuni vantaggi, come la massima libertà, velocità e efficienza di apportare modifiche o cambiamenti all'infrastruttura e alle tecnologie coinvolte. Sarà però necessario possedere una sufficiente larghezza di banda sulla rete pubblica. Inoltre i costi di avviamento, manutenzione e gestione dell'infrastruttura interna potrebbero rivelarsi notevoli, poiché **NON** scalabili tempestivamente, per esempio, i costi iniziali dell'hardware, dei sistemi operativi, delle piattaforme software (**up front cost**).

La gestione e la manutenzione dei sistemi hardware e software, quindi, richiederanno competenze e forza lavoro tecnica relativamente costosa e non preventivabile con precisione.

In una seconda fase è sicuramente consigliabile una infrastruttura in Cloud, per la quale tutti i servizi e tutto l'hardware sono garantiti, così come i costi di gestione e di manutenzione, potendo quindi preventivare con precisione l'impegno economico chiavi in mano (**ROI**, return on investment). In questo caso la scalabilità è garantita.

Per quanto riguarda la sicurezza del sistema informatizzato, si può ricordare che, se il trasferimento dei dati su rete pubblica avviene sostanzialmente via Web (client e server Web), la misura di sicurezza più indicata per la sicurezza è l'adozione di **HTTPS** (HTTP trasportato da TLS/SSL) con **Certificati Digitali**.

In questo caso il protocollo garantisce autenticazione, segretezza e integrità, rispettivamente con i più importanti algoritmi disponibili di cifratura asimmetrica, simmetrica e di Hash: es., RSA, AES, SHA.

Il primo canale (cliente/mittente e Azienda) garantirà solo l'autenticazione dell'Azienda presso il cliente/mittente, ma saranno comprese le cifrature simmetrica e d'integrità in ambo i versi.

Il canale tra App e Azienda, invece, potrebbe essere protetto completamente in entrambi i versi dotando anche il client App di Certificato Digitale.

Nel caso di uso WiFi dell'App, il protocollo **WPA2** Enterprise (o Personal con chiave preimpostata) sono sufficienti per proteggere la connessione wireless interna alla SO e al CSR.

A livello di intero sistema, invece la garanzia della **Continuità Operativa** (Business Continuity) e la gestione della **Tolleranza ai Guasti** (Fault Tolerance) dovrebbero essere affrontate sottoponendo l'Azienda alle certificazioni ISO del caso (**ISO 22301** del 2012 e **ISO 27001**).

Le politiche per ottenere tolleranza ai guasti si possono riassumere nella duplicazione dell'hardware a vari livelli.

A livello elaborativo si possono prevedere più macchine che ospitano le stesse funzionalità (es., più Domain Controller su una LAN) o su singole macchine (sistemi di dischi ridondanti tipo RAID).

A livello sistemistico, prevedere più linee pubbliche (es., linee di failover) o più linee private (es., topologie a maglia riducibili ad albero tramite STP).

Le politiche per ottenere la continuità operative si potrebbero tradurre invece nella capacità di prevenire disastri e di ristabilire l'infrastruttura in occasione di un disastro (disaster recovery), come ad esempio dotare le linee pubbliche di controllo dei flussi tramite apparati dedicati (es., Firewall) o a livello di singoli host, proteggere i flussi con antivirus e aggiornamenti repentini di sistema o del software server (es., gestione delle patch).

Le politiche per il ripristino da disastro si possono sintetizzare, in effetti, nell'organizzare delle tecnologie consolidate per il backup e restore dei sistemi (es., tramite la gestione strutturata dei repository, anche in luoghi esterni all'organizzazione).

Schema concettuale E/R

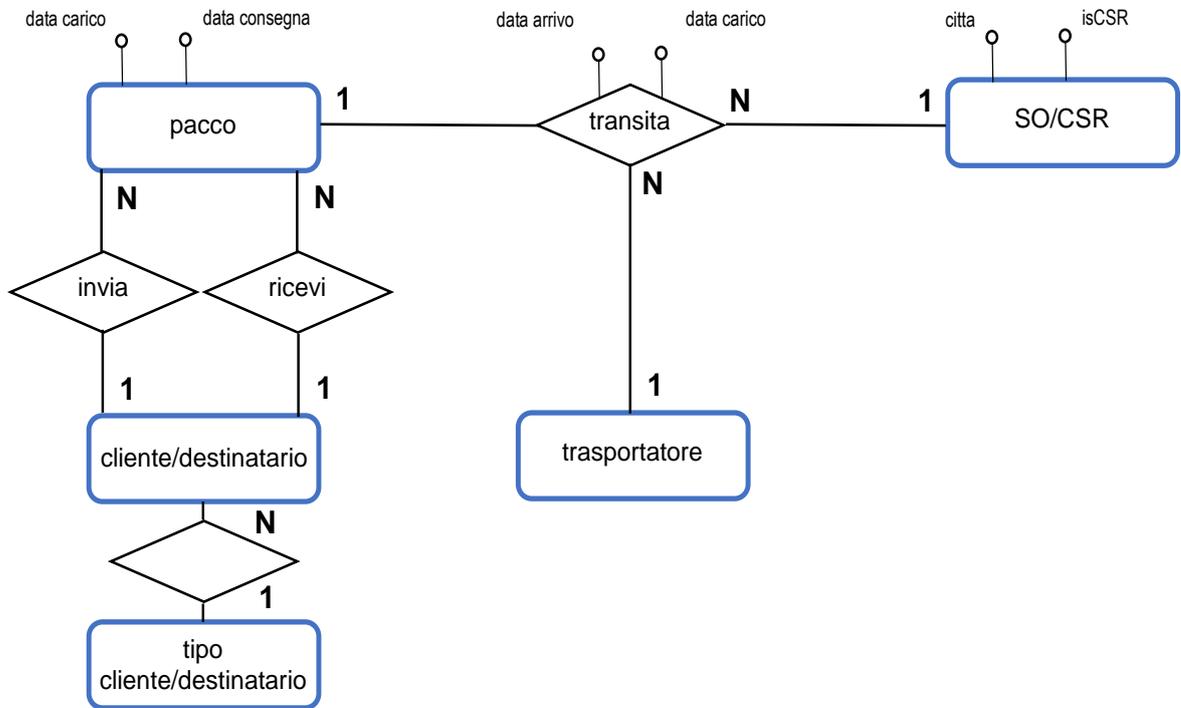
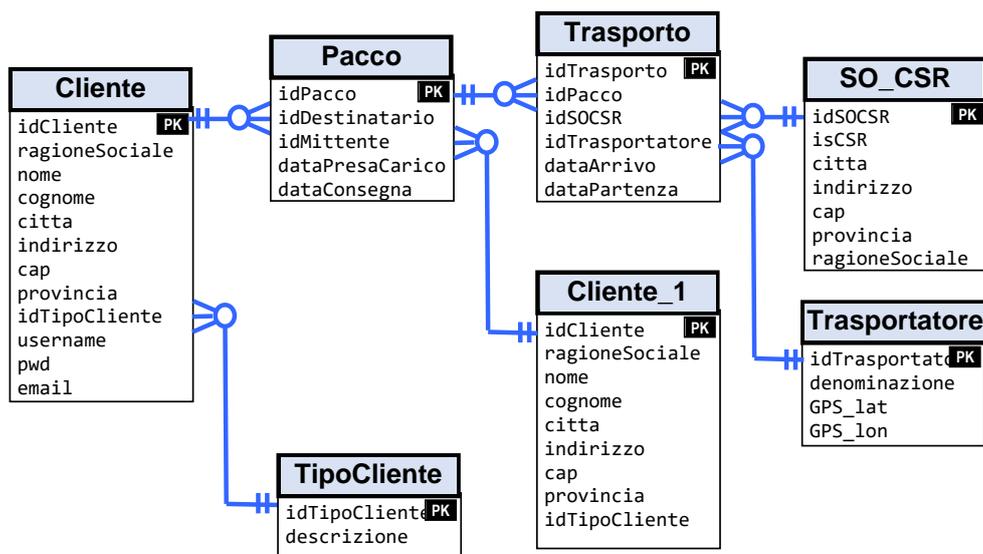


Diagramma Logico



Versione in HTML/PHP

File **visualizzapacco.html**

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>Tracciamento pacco</title>

</head>

<body>
<p>Tracciamento Pacco</p>
<form id="formRichiesta" name="formRichiesta" method="post" action="visualizzapacco.php">
  <label>ID Pacco<input type="text" name="idpacco" /></label>
  <input type="submit" name="ok" />
</p>
</form>
</body>
</html>
```

File `visualizzapacco.php`

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Tracciamento pacco</title>
</head>

<body>
<?php
  $host = "localhost";
  $user = "root";
  $password = "";
  $db = "sistemi2018";

  $connect = new mysqli($host, $user, $password);
  if ($connect->connect_error)
  {
    die("Errore connessione: " . $connect->connect_error);
  }

  if($connect->select_db($db) == 0)
  {
    echo("Il database non esiste");
    exit;
  }
  $sql = "SELECT Pacco.idPacco, Pacco.dataConsegna, Trasporto.dataArrivo, Trasporto.idSO_Destinazione, ".
    "SO_CSR.indirizzo, SO_CSR.città, SO_CSR.RagioneSociale ".
    "FROM Trasportatore INNER JOIN (SO_CSR INNER JOIN (Pacco INNER JOIN Trasporto ON Pacco.idPacco = ".
    "Trasporto.idPacco) ".
    "ON SO_CSR.idSO=Trasporto.idSO_Destinazione) ON Trasportatore.idTrasportatore ".
    "=Trasporto.idTrasportatore ".
    "WHERE Pacco.idPacco='".$$_POST["idpacco"]."' ".
    "ORDER BY Trasporto.dataArrivo;";

  $result = $connect->query($sql);

  if ($result->num_rows > 0)
  {
    echo "<table>";
    echo "<tr><th>idPacco</th><th>Data Consegna</th><th>Data Arrivo</th><th>Destinatario</th></tr>";
    while($row = $result->fetch_assoc())
    { // approccio ai risultati come array associativo
      echo "<tr>";
      echo "<td>". $row["Pacco.idPacco"]. "</td><td>". $row["Pacco.dataConsegna"]. "</td>";
      echo "<td>". $row["Trasporto.dataArrivo"] . "</td><td>". $row["SO_CSR.RagioneSociale"]. "</td>";
      echo "</tr>";
    }
    echo "</table>";
  }
  else
  {
    echo "0 risultati";
  }

  $connect->close();
?>
</body>
</html>

```

Versione in ASPX

File TracciaPacco.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="TracciaPacco.aspx.cs"
Inherits="WebApplicationSistemi2018.TracciaPacco" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>Inserisci il codice spedizione ;
<asp:TextBox ID="tbCodPacco" runat="server"></asp:TextBox>
<asp:Button ID="btInvia" runat="server" Text="Cerca" OnClick="btInvia_Click" /><br /><br />
<asp:Repeater ID="repTracciaPacco" runat="server">
<ItemTemplate>
<%= Eval("RagioneSociale") %> - <%= Eval("DataArrivo") %> - <%= Eval("Città") %><br />
</ItemTemplate>
</asp:Repeater>

<br />
<asp:Literal ID="litData" runat="server"></asp:Literal>

</div>
</form>
</body>
</html>
```

File **TracciaPacco.aspx.cs**

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.OleDb;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplicationSistemi2018
{
    public partial class TracciaPacco : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        private void CaricaTracciabilita()
        {

            string connectionString = System.Configuration.ConfigurationManager.
                ConnectionStrings["ConnectionStringAccess"].ConnectionString;

            // Provide the query string with a parameter placeholder (@Par).
            string queryString = "SELECT Pacco.idPacco, Pacco.dataConsegna, Trasporto.dataArrivo,
Trasporto.idSO_Destinazione, SO_CSR.indirizzo, SO_CSR.città, SO_CSR.RagioneSociale FROM Trasportatore INNER
JOIN (SO_CSR INNER JOIN(Pacco INNER JOIN Trasporto ON Pacco.idPacco = Trasporto.idPacco) ON SO_CSR.idSO =
Trasporto.idSO_Destinazione) ON Trasportatore.idTrasportatore = Trasporto.idTrasportatore
WHERE(((Pacco.idPacco) = " +tbCodPacco.Text + ")) ORDER BY Trasporto.dataArrivo;";

            // Create and open the connection in a using block. This ensures that all resources will be
            // closed when the code exits.
            using (OleDbConnection connection = new OleDbConnection(connectionString))
            {
                // Create the Command and Parameter objects.
                OleDbCommand command = new OleDbCommand(queryString, connection);

                try // Open the connection in a try/catch block. Create and execute the DataReader, writing
                    the result
                {
                    connection.Open();
                    OleDbDataAdapter da = new OleDbDataAdapter(command);
                    DataSet ds = new DataSet();
                    da.Fill(ds);
                    repTracciaPacco.DataSource = ds.Tables[0];
                    repTracciaPacco.DataBind();
                }
                catch (Exception ex)
                {
                    litData.Text = ex.Message;
                }
            }

            protected void btInvia_Click(object sender, EventArgs e)
            {
                CaricaTracciabilita();
            }
        }
    }
}

```

Tramite la soluzione proposta, la geolocalizzazione degli automezzi potrebbe essere ottenuta aggiungendo alla App in uso al trasportatore un codice client che attraverso un timer (con frequenza impostabile) richiami la posizione attraverso le **API di GoogleMaps** e, ottenuta la posizione, la invii al server Web che provvederà a salvarla sul database. Lo smartphone dovrà essere dotato di localizzatore GPS.

Esempio di codice:

```
<!DOCTYPE html>
<html>
<body>

<p>Cliccare per acquisire le coordinate. Il GPS deve essere in funzione</p>
<button onClick="PrendiPosizione()">Clicca</button>
<p id="demo"></p>
<script>
var x = document.getElementById("demo");

function PrendiPosizione()
{
    if (navigator.geolocation)
    {
        navigator.geolocation.getCurrentPosition(mostraPosizione);
    }
}

function mostraPosizione(position)
{
    x.innerHTML = "Latitudine: " + position.coords.latitude + "<br>
                Longitudine: " + position.coords.longitude;
}
</script>

</body>
</html>
```

Per **virtualizzazione** si intendono quelle tecniche informatiche per cui si riuniscono in un solo hardware (**host**) un insieme di sistemi operativi (**guest**) e, tramite uno strato software specializzato (**VMM Virtual Machine Monitor o Hypervisor**) l'host simula gli strati hardware dei vari guest attraverso il proprio.

La centralizzazione dell'hardware per vari sistemi di rete offre una gestione semplificata del sistema complessivo: un solo sistema hardware è più facilmente gestibile di molti sistemi hardware ed è tendenzialmente meno costoso. Per esempio, la memoria principale (RAM) viene gestita in maniera più efficiente e senza le duplicazioni che la gestione di più sistemi fisici comporta. La gestione di pochi ma ampi ed efficienti hard disk è sicuramente migliore della gestione di tanti hard disk, magari di diverso formato, taglio e velocità. Il costo dell'alimentazione di una sola macchina, benchè più potente, è inferiore alla somma dei costi delle alimentazioni di più macchine, ecc...

Inoltre le funzionalità avanzate di una singola macchina ad alte prestazioni vengono distribuite a tutte le macchine virtuali, come ad esempio la tolleranza ai guasti degli hard disk (sistemi RAID), le funzionalità avanzate dei più recenti microprocessori, le prestazioni delle schede di rete più sofisticate, ecc...

Anche la manutenzione di un solo hardware è nettamente più semplice della manutenzione di molti hardware, magari diversi tra loro e di case costruttrici differenti: la gestione dei driver di periferica, da sempre, è un cruccio per i sistemisti. I servizi generali, come per esempio l'effettuazione periodica della copia dei dati (backup) può essere gestita in modo centralizzato su tutti i sistemi virtualizzati senza dover prevedere, per ogni singolo sistema, una politica dedicata.

Una macchina virtuale (che ospita un sistema operativo e tutte le sue configurazioni e applicazioni), essendo del tutto scorrelata dall'hardware su cui è installata, può essere completamente salvata su un disco di copia (snapshot), sia i dati sia soprattutto il sistema operativo con tutte le sue configurazioni. Questa opzione è fondamentale: effettuando il backup periodico di una macchina virtuale, essa potrà essere recuperata completamente in seguito ad un guasto o a una errata configurazione, oppure spostata interamente su un nuovo hardware virtualizzato senza alcuna necessità di reinstallazione e riconfigurazione del sistema operativo e dei servizi.

Infine i sistemi virtualizzati si rendono particolarmente efficienti per quanto riguarda la loro gestione remota: tramite un software di sistema l'accesso al sistema virtualizzato da remoto consente la gestione di tutti i server in esso contenuti.

Per **clusterizzazione** si intende un insieme di computer connessi tra loro tramite una rete telematica locale o remota allo scopo di distribuire una elaborazione molto complessa tra i vari computer componenti il cluster. Il clustering aumenta la potenza di calcolo del sistema e permette una maggiore affidabilità, una garanzia superiore nella sicurezza dei dati e della continuità operativa (**business continuity**) in occasione di mal funzionamenti del sistema.

Virtualizzazione e clusterizzazione possono e devono essere gestite anche contemporaneamente.

SECONDA PARTE Quesito 4

La **Posta Elettronica** è uno dei canali tipici attraverso i quali vengono veicolate le minacce informatiche, soprattutto tramite l'apertura incauta dell'allegato.

Inoltre il flusso dei dati che consente la spedizione e la ricezione di eMail non nasce nativamente protetto, nei sui due lati client, ne' nella tratta tra i server di posta.

Oggi le tratte servite da **SMTP** (client di spedizione- server e server-server) sono sempre protette dal protocollo **TLS**, che garantisce autenticazione, segretezza e integrità.

La tratta di ricezione server-client invece puo' essere protetta con speciali protocolli.

La tecnica della **Posta Elettronica Certificata** invece dovrebbe garantire anche il riscontro di spedizione e ricezione, benchè a livello puramente tecnico questa garanzia e' valida solo tra i due server coinvolti e non *end to end*.

Testi di:

Sistemi e reti

prof. Paolo Ollari
prof. Ramon Ugolotti

Informatica

prof. Corrado Pagani
prof. Alberto Paganuzzi

ITIS "L. Da Vinci", Parma