

Università degli studi di Padova

## FACOLTA' DI SCIENZE STATISTICHE

Tesi di Laurea Triennale in Statistica e Tecnologie Informatiche



## Sviluppo di un'applicazione di basi di dati online per la preventivazione

Developing an online database application for estimate management

**Relatore:**  
Prof.  
Melucci Massimo

**Laureando:**  
Artusi Ingres  
571826

Anno Accademico 2010-2011

# Indice

<b>Introduzione</b>	<b>1</b>
I.1 Obiettivo.....	1
<b>1. Analisi dei requisiti</b>	<b>2</b>
1.1 Analisi dei requisiti.....	2
<b>2. Progettazione</b>	<b>6</b>
2.1 Schema concettuale.....	6
2.2 Schema logico.....	8
2.3 Interfacce.....	10
2.4 Pagine e link.....	13
<b>3. Realizzazione</b>	<b>14</b>
3.1 Schema fisico.....	14
3.2 Flash.....	19
3.3 Applicazione.....	22
3.4 Documenti.....	43
3.5 Reportistica.....	46
<b>A. Appendice</b>	<b>48</b>
A.1 Il modello ER.....	48
A.2 I concetti base di php.....	49
2.1 Get e post.....	49
2.2 Connessione database.....	51
2.3 Sessioni.....	52
2.4 Upload.....	53
<b>Conclusioni</b>	<b>54</b>
<b>Bibliografia - Siti consultati - Ringraziamenti</b>	<b>55</b>



# Introduzione

L'azienda ospitante Tosato Metodi opera dal 1994 e realizza impianti elettrici, di sicurezza e di telecomunicazione.

Lo stage di 640 ore si è svolto sviluppando e realizzando il progetto di un software per la formulazione di preventivi, schede costi e ordini che prima erano gestiti "manualmente" con excel e word.

## I.1 OBIETTIVO

L'obiettivo dello stage è di realizzare un'applicazione che permetta di gestire tramite interfaccia web e database:

- Importazione dei file metel
- Gestione fornitori, clienti e cantieri
- Utilizzo di voci-computo
- Output dei documenti
- Reportistica

Per la realizzazione è stata necessaria la conoscenza dei linguaggi HTML, PHP e SQL.

# Capitolo 1

## Analisi dei requisiti

### 1.1 ANALISI REQUISITI

L'analisi dei requisiti è una delle prime fasi di vita di un software. Lo scopo è stabilire che cosa il sistema in questione deve fare.

E' una fase che richiede una interazione con gli utenti del sistema. L'analisi dei requisiti avviene come negoziazione fra gli analisti e i committenti. Gli analisti possono avere difficoltà a comprendere il linguaggio del committente, e viceversa.

Senza una precisa definizione dei requisiti è impossibile effettuare correttamente la progettazione e la realizzazione del sistema.

I termini utilizzati sono:

- Cliente (azienda che richiede uno o più preventivi)
- Cantiere (luogo dove viene svolto il lavoro)
- Fornitore (soggetto che vende prodotti alla Tosato Metodi)
- Prodotto (oggetto che può essere acquistato dalla Tosato Metodi e inserito nelle voci computo)
- Voce computo (insieme di prodotti)

- Metel (standard prodotti in formato txt)
- Documenti (comprende il preventivo, la scheda costi e l'ordine)

Tosato Metodi vuole archiviare le informazioni relative:

- clienti e relativi cantieri
- fornitori e relativi prodotti
- voci computo (per velocizzare la creazione dei documenti)
- preventivi

Le informazioni richieste per i clienti (circa 100 tuple all'anno) sono:

- Codice fiscale, Partita iva e Ragione sociale
- Nome e Cognome
- Indirizzo, Provincia, Città e CAP
- Telefono e Email

Per quanto riguarda i cantieri (circa 140 tuple all'anno):

- Indirizzo, Provincia, Città e CAP

Ogni cliente possiede uno o più cantieri. Ogni cantiere è di uno e un solo cliente.

Le informazioni richieste per i fornitori (3 tuple) sono:

- Codice fiscale, Partita iva e Ragione sociale
- Nome e Cognome
- Indirizzo, Provincia, Città e CAP
- Telefono e Email

e le informazioni necessarie per importare i prodotti dai file metel:

- Inizio e lunghezza codice Metel
- Inizio e lunghezza marca Metel
- Inizio e lunghezza prezzo
- Inizio e lunghezza descrizione
- Inizio e lunghezza unità di misura
- Numero decimali e se c'è il separatore

Per quanto riguarda i prodotti (circa 320000 tuple):

- Codice Metel e marca Metel
- Prezzo (con 5 cifre decimali), descrizione e unità di misura

Ogni fornitore vende uno o più prodotti. Ogni prodotto è venduto da uno e un solo fornitore (nell'importazione viene importato solo il prodotto dal fornitore con prezzo inferiore).

Le informazioni richieste per le voci computo (circa 50 tuple) sono:

- Nome
- Lista prodotti con quantità, prezzo e ore di manodopera

Le informazioni richieste per i preventivi (circa 150 tuple all'anno) sono:

- Nome e tipo
- Data
- Valore economico (utile del preventivo) e valore offerta (ricavo del preventivo)
- Altri costi associati (preparazione, collaudo, redazione documenti, carburante, pedaggio, alloggio, vitto, viaggio e trasferta)
- Percentuale utile e percentuale spese generali

Ogni preventivo appartiene a uno e un solo tipo. Ogni tipo è associato ad uno o più preventivi.



## Capitolo 2

# Progettazione

## Progettazione della base di dati

### 2.1 SCHEMA CONCETTUALE (Per una descrizione del modello ER - A.1)

Le entità in questo caso sono tipi, preventivi, clienti, cantieri, vociComputo, prodotti e fornitori mentre le associazioni sono afferenza (tra tipi e preventivi), emissione (tra preventivi e clienti), possessione (tra clienti e cantieri), righePreventivo (tra preventivi e vociComputo) e vendita (tra prodotti e fornitori).

Gli attributi sono elencati vicino all'entità corrispondente.

Nell'entità Clienti ho usato la chiave codiceCliente anche se la più naturale è partitaIva perché non sempre il committente dispone di questa informazione. Lo stesso vale per l'entità Fornitori.

Sviluppo di un'applicazione di basi di dati online per la preventivazione

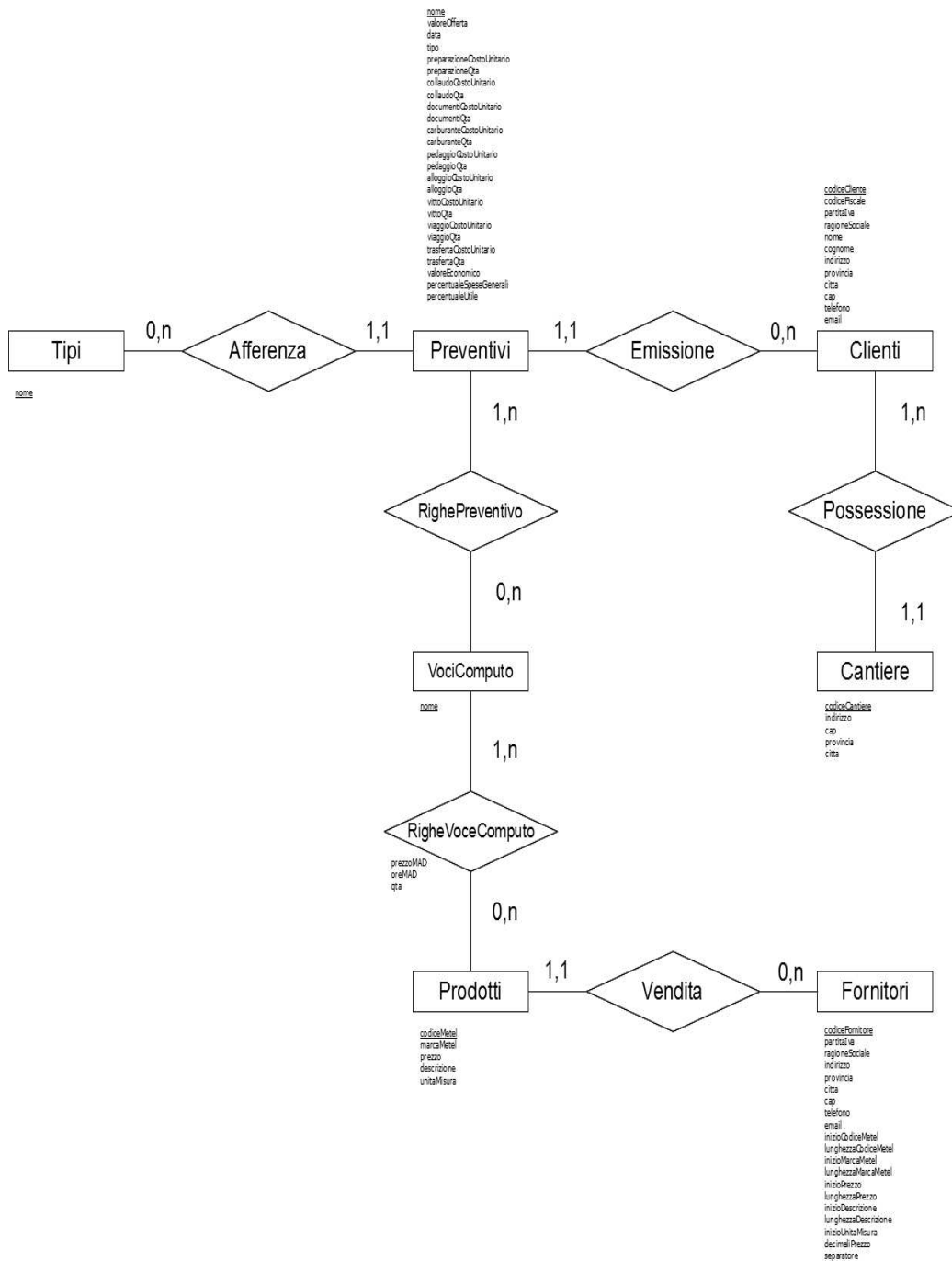


Figura 1

## 2.2 SCHEMA LOGICO

La seconda fase della progettazione logica corrisponde nella traduzione dallo schema ER allo schema logico equivalente, in grado cioè di rappresentare le medesime informazioni.

Con entità e associazioni molti a molti per ogni entità una relazione con lo stesso nome avente per attributi i medesimi attributi dell'entità e per chiave il suo identificatore. Per l'associazione, una relazione con lo stesso nome avente per attributi gli attributi dell'associazione e gli identificatori delle entità coinvolte. Con le associazioni uno a molti si ha per ogni entità una relazione con lo stesso nome avente per attributi i medesimi attributi dell'entità e per chiave il suo identificatore e nell'entità che partecipa all'associazione con cardinalità massima pari a uno ci sono anche gli identificatori e attributi dell'associazione stessa.

Dato lo schema ER (Figura 1), la traduzione è:

cantieri(codiceCantiere, indirizzo, cap, provincia, citta,  
*codiceCliente*)

clienti(codiceCliente, codiceFiscale, partitaIva, ragioneSociale,  
nome, cognome, indirizzo, provincia, citta, cap, telefono, email)

fornitori(codiceFornitore, partitaIva, ragioneSociale, indirizzo,  
provincia, citta, cap, telefono, email, inizioCodiceMetel,  
lunghezzaCodiceMetel, inizioMarcaMetel, lunghezzaMarcaMetel,

inizioPrezzo, lunghezzaPrezzo, inizioDescrizione,  
lunghezzaDescrizione, inizioUnitaMisura, lunghezzaUnitaMisura,  
decimaliPrezzo, separatore)

preventivi(nome, valoreOfferta, data, *tipo*, codiceCliente,  
preparazioneCostoUnitario, preparazioneQta,  
collaudoCostoUnitario, collaudoQta, documentiCostoUnitario,  
documentiQta, carburanteCostoUnitario, carburanteQta,  
pedaggioCostoUnitario, pedaggioQta, alloggioCostoUnitario,  
alloggioQta, vittoCostoUnitario, vittoQta, viaggioCostoUnitario,  
viaggioQta, trasfertaCostoUnitario, trasfertaQta, valoreEconomico,  
percentualeSpeseGenerali, percentualeUtile)

prodotti(codiceMetel, marcaMetel, prezzo, descrizione, unitaMisura,  
*codiceFornitore*)

righepreventivo(nomePreventivo, nomeVoceComputo)

righevocecomputo(nome, *codiceMetel*, prezzoMad, oreMad, qta)

tipi(nome)

vocicomputo(nome)

# Progettazione dell'applicazione

## 2.3 INTERFACCE

Il committente, unico utilizzatore del software, ha fornito le seguenti precise indicazioni per l'interfaccia:

- Titolo della pagina in alto-sinistra (font Verdana, colore #4F9671)
- Logo azienda in alto-destra
- Contenuto (font Verdana, colore #212121)
- Opzioni (font Verdana, colore #212121)

Il risultato è indicato nella Figura 2.

Nome Voce computo	Operazione
Sorveglianza 12RW	

Nome Voce Computo

Nome Voce Computo	Operazione
ELE9110	

Aggiungi Voce Computo  
Elimina Voce Computo  
Aggiungi Fornitore  
Elimina Fornitore  
Aggiorna prodotti Fornitore

Avanti >>>  
Indietro

Figura 2

Mentre per quanto riguarda i documenti:

Preventivo (formato word)

- Logo azienda in alto-sinistra
- Informazioni Tosato Metodi (font Verdana, colore #333333)
- Informazioni cliente a destra (font Verdana, colore #000000)
- Testo (font Verdana, colore #333333)

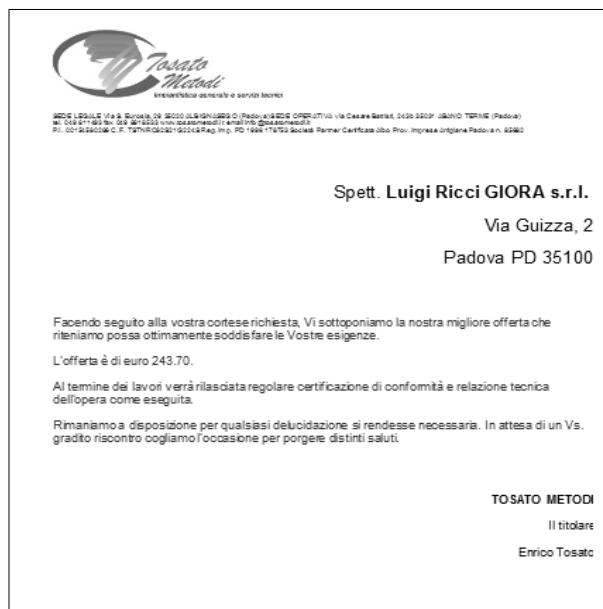


Figura 3

### Scheda costi (formato excel)

- Testo (font Arial, colore #000000)
- Layout fornito dal committente

A	B	C	D	E
<b>SCHEDA COSTI - Preventivo: 3WM</b>				
1	COMMITTENTE	Ingres Artusi	CANTIERE	Via Ful
2	Sorveglianza 339			
3	MATERIALI			
4	Codice Metal	Descrizione	Quantità	Prezzo
5	80609100047	PINZA SCOTCHLOCK X CONNETTORI TELEF	100	19.7382
6	H54221017	TELECAM ISP H 5422/H D.17MM	5	489.6144
7	2448.0			
8	ALTRI COSTI			
9		Costo unitario	Quantità	Importo
10	Preparazione	21	2	42
11	Collaudo	15	10	150
12	Redazione Documenti	10	1	10
13	Carburante	3	100	300
14	Pedaggio	7.9	4	31.6
15	Alloggio	0	0	0
16	Vitto	0	0	0
17	Viaggio	0	0	0
18	Trasferte	0	0	0
19	SINTESI			
20	Valore offerta			6907.93
21	Materiali			4421.89
22	Manodopera			260
23	Spese generali	15		1036.19
24	Altri costi			533.6
25	Totale costi			6251.68
26	Utile	9.5		656.25
27				

Figura 4

Ordine per ogni fornitore (formato pdf)

- Nome del fornitore (font Arial, colore #000000)
- Elenco prodotti (font Arial, colore #000000)
- Totale (font Arial, colore #000000)

Ordine ElektroVeneta	
Codice Metal:	49173
Descrizione:	DVR 4 INGRESSI STD ALONE
Prezzo:	132.52500
Quantità:	1
Codice Metal:	TBC15I
Descrizione:	MAK1202P TELECAMERA DA INT/EST 12 LED IR
Prezzo:	51.46000
Quantità:	1
<b>Totale</b>	<b>183.99</b>

Figura 5

## 2.4 PAGINE E LINK

La Figura 6 infine illustra tutti i collegamenti tra le varie pagine:

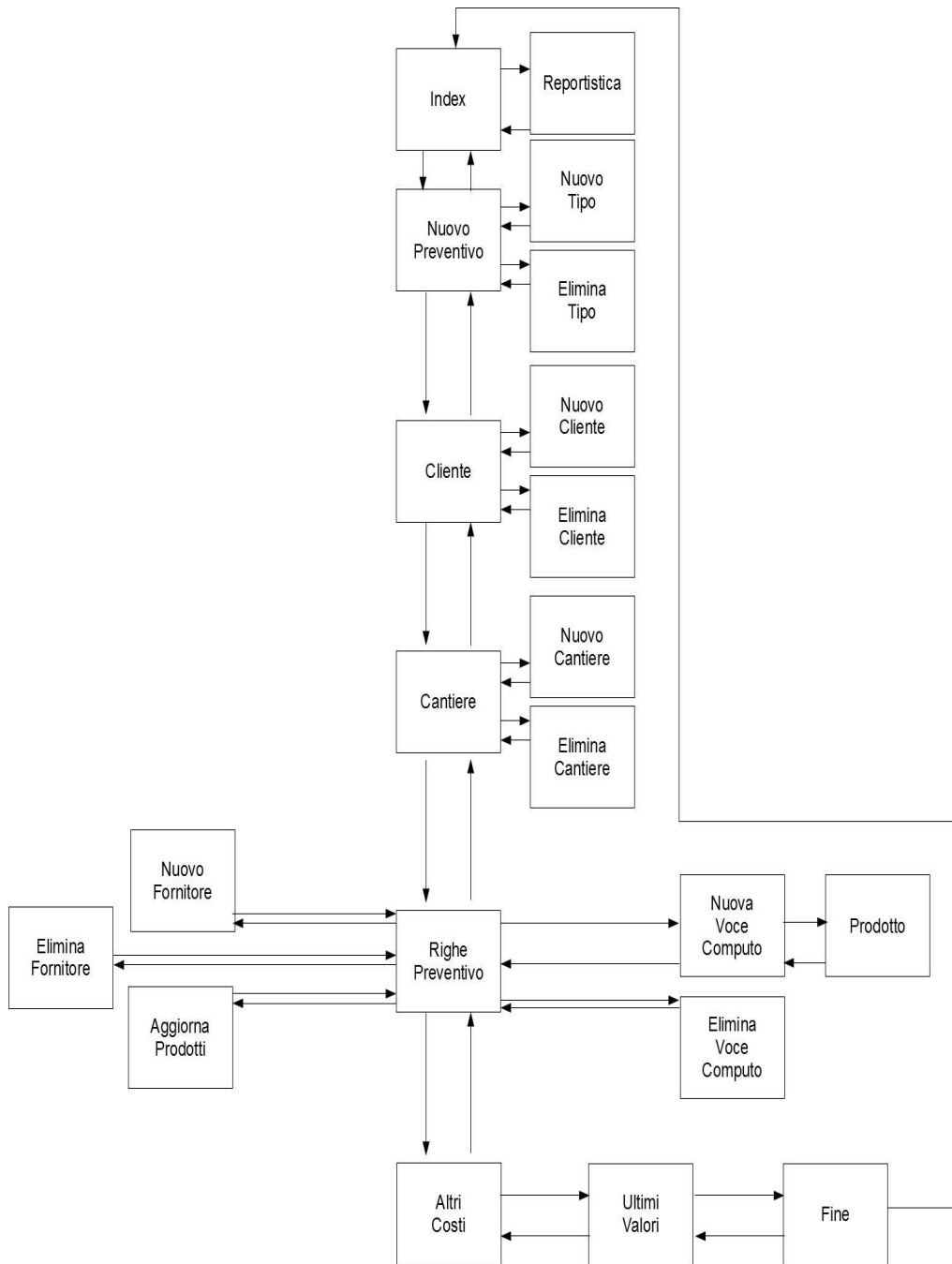


Figura 6



## Capitolo 3

# Realizzazione

## Realizzazione della base di dati

Nella fase finale ricevo in ingresso lo schema logico della base di dati e produco in uscita lo schema fisico della base di dati in forma di istruzioni create table in SQL.

### 3.1 SCHEMA FISICO

```
CREATE TABLE cantieri (  
    codiceCantiere int(10) auto_increment,  
    indirizzo varchar(40),  
    cap varchar(40) ,  
    provincia varchar(40) ,  
    citta varchar(40) ,  
    codiceCliente int(10) ,  
    PRIMARY KEY (codiceCantiere),  
    FOREIGN KEY (codiceCliente) REFERENCES clienti(codiceCliente)  
);  
  
CREATE TABLE clienti (  
    codiceCliente int(10) auto_increment,  
    codiceFiscale varchar(40) ,  
    partitaIva varchar(40) ,  
    ragioneSociale varchar(40) ,  
    nome varchar(40) ,  
    cognome varchar(40) ,  
    indirizzo varchar(40) ,
```

```
provincia varchar(40) ,
citta varchar(40) ,
cap varchar(40) ,
telefono varchar(40) ,
email varchar(40) ,
PRIMARY KEY (codiceCliente)
);
CREATE TABLE fornitori (
codiceFornitore int(10) auto_increment,
partitaIva varchar(40) ,
ragioneSociale varchar(40) ,
indirizzo varchar(40) ,
provincia varchar(40) ,
citta varchar(40) ,
cap varchar(40) ,
telefono varchar(40) ,
email varchar(40) ,
inizioCodiceMetel int(5) ,
lunghezzaCodiceMetel int(5) ,
inizioMarcaMetel int(5) ,
lunghezzaMarcaMetel int(5) ,
inizioPrezzo int(5) ,
lunghezzaPrezzo int(5) ,
inizioDescrizione int(5) ,
lunghezzaDescrizione int(5) ,
inizioUnitaMisura int(5) ,
lunghezzaUnitaMisura int(5) ,
decimaliPrezzo int(3) ,
separatore int(1) ,
PRIMARY KEY (codiceFornitore)
```

);

```
CREATE TABLE preventivi (  
  nome varchar(40) ,  
  valoreOfferta double(10,2) ,  
  data varchar(10) ,  
  tipo varchar(40) ,  
  codiceCliente int(10) ,  
  preparazioneCostoUnitario double(15,2) ,  
  preparazioneQta int(5) ,  
  collaudoCostoUnitario double(15,2) ,  
  collaudoQta int(5) ,  
  documentiCostoUnitario double(15,2) ,  
  documentiQta int(5) ,  
  carburanteCostoUnitario double(15,2) ,  
  carburanteQta int(5) ,  
  pedaggioCostoUnitario double(15,2) ,  
  pedaggioQta int(5) ,  
  alloggioCostoUnitario double(15,2) ,  
  alloggioQta int(5) ,  
  vittoCostoUnitario double(15,2) ,  
  vittoQta int(5) ,  
  viaggioCostoUnitario double(15,2) ,  
  viaggioQta int(5) ,  
  trasfertaCostoUnitario double(15,2) ,  
  trasfertaQta int(5) ,  
  valoreEconomico double(15,2) default NULL,  
  percentualeSpeseGenerali double(10,2) ,  
  percentualeUtile double(10,2) ,  
  PRIMARY KEY (nome),  
  FOREIGN KEY (tipo) REFERENCES tipi(nome),
```

```
FOREIGN KEY (codiceCliente) REFERENCES clienti(codiceCliente)
);
CREATE TABLE prodotti (
  codiceMetel varchar(16) ,
  marcaMetel varchar(10) ,
  prezzo double(15,5) default NULL,
  descrizione varchar(50) ,
  unitaMisura varchar(5) ,
  codiceFornitore int(10) ,
  PRIMARY KEY (codiceMetel),
  FOREIGN KEY (codiceFornitore) REFERENCES fornitori(codiceFornitore)
);
CREATE TABLE righepreventivo (
  nomePreventivo varchar(40) ,
  nomeVoceComputo varchar(40) ,
  PRIMARY KEY (nomePreventivo,nomeVoceComputo),
  FOREIGN KEY (nomePreventivo) REFERENCES preventivi(nome),
  FOREIGN KEY (nomeVoceComputo) REFERENCES righevocecomputo(nome)
);
CREATE TABLE righevocecomputo (
  nome varchar(40) ,
  codiceMetel varchar(16) ,
  prezzoMad double(15,2) ,
  oreMad int(10) ,
  qta int(10) ,
  PRIMARY KEY (nome,codiceMetel),
  FOREIGN KEY (codiceMetel) REFERENCES prodotti(codiceMetel)
);
CREATE TABLE tipi (
  nome varchar(40) ,
```

```
PRIMARY KEY (nome)  
);  
CREATE TABLE vocicomputo (  
nome varchar(40) ,  
PRIMARY KEY (nome)  
);
```

Non sono stati aggiunti constraint di chiave esterna perché, nel primo periodo di utilizzo, in caso di errori non previsti è possibile ricavare i valori dal database.

## Realizzazione dell'applicazione

### 3.2 FLASH

Index contiene un'animazione Flash (richiesta dal committente).

Flash è un software per uso prevalentemente grafico che consente di creare animazioni principalmente per il web.

Permette di creare animazioni interattive, grazie alla presenza di un linguaggio di scripting interno denominato ActionScript, è possibile applicare comportamenti agli oggetti o ai fotogrammi dell'animazione.

Per visualizzare i siti e le applicazioni creati con Flash, Adobe mette a disposizione gratuitamente il player.

Dopo una breve introduzione figura 7



Figura 7

Compare il menù figura 8



Figura 8

La linea Temporale di Flash contiene fotogrammi che riempiti formeranno un'animazione.

Utilizzando Inserisci fotogramma chiave crea un fotogramma chiave utilizzando il contenuto del fotogramma chiave precedente. Spostando un oggetto o cambiando le proprietà (in questo caso Alpha) crea l'animazione automaticamente. Ad esempio nel fotogramma chiave 110, ho i testi del menù con alpha 0 (Figura 9) e al fotogramma chiave 130, ho i testi del menù con alpha 100 (Figura 10)

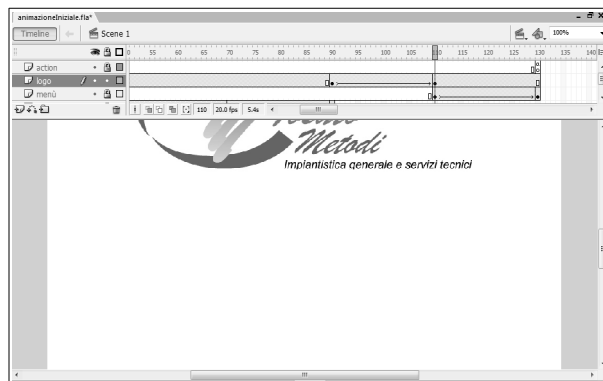


Figura 9



Figura 10

mi posiziono sul primo fotogramma chiave e seleziono motion, automaticamente crea l'animazione.

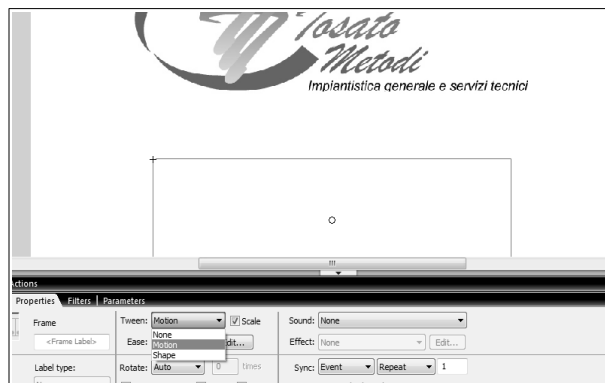


Figura 11

Nell'ultimo frame ho aggiunto lo script:

```
stop();
```

(altrimenti l'animazione continua a ciclo) e nei due bottoni (uno indirizzato a nuovoPreventivo.php l'altro a reportistica.php) lo script:



```
on(release)
{
    getURL("nuovoPreventivo.php");
}
```

Quando l'utente rilascia il bottone accede alla pagina nuovoPreventivo.php o reportistica.php.

### 3.3 APPLICAZIONE (Per una descrizione dei concetti base di PHP - A.2)

La pagina nuovoTipo.php contiene un form con il campo nome che indirizza alla stessa pagina (figura 12).



Figura 12

Se esiste `$_POST["nome"]` allora inserisce il tipo all'interno della tabella tipi:

```
if(isset($_POST["nome"]))
{
    if($_POST["nome"]=="")
    {
```

```
        echo("<p align='center'><font color='#CC0000' face='Verdana'
size='3'>Il campo nome è obbligatorio</font></p>");
    }
    else
    {
        include("connessione.php");
        $query="insert into tipi (nome) values ('$_POST[nome]')";
        $result=mysql_query($query, $conn);
        if($result)
        {
            echo("<p align='center'><font color='#00CC00'
face='Verdana' size='3'>Nuovo Tipo salvato</font></p>");
        }
        else
        {
            echo("<p align='center'><font color='#CC0000'
face='Verdana' size='3'>Impossibile salvare il nuovo Tipo</font></p>");
        }
    }
}
```

mysql\_query() restituisce vero se la query è stata eseguita, falso nel caso contrario. Questo permette di gestire gli errori mostrando del testo formattato utilizzando la funzione echo().



Figura 13



Figura 14

La pagina nuovoPreventivo.php contiene all'interno del form un menù a tendina con l'elenco dei tipi salvati, in questo caso utilizzo `mysql_fetch_array()`.

Il codice PHP relativo al menù a tendina è:

```
$query="select count(nome) as numero from tipi";  
$result=mysql_query($query, $conn);  
if($result)  
{
```

```

$row=mysql_fetch_array($result);
$numero=$row["numero"];
if($numero>0)
{
    $query="select nome from tipi order by nome";
    $result=mysql_query($query, $conn);
    if($result)
    {
        echo("<select size='1' name='tipo'>");
        while($row=mysql_fetch_array($result))
        {
            $nome=$row["nome"];
            echo("<option value='$nome'>$nome</option>");
        }
        echo("</select>");
    }
}
else
{
    echo("<font color='#CC0000' face='Verdana' size='2'>Non ci sono
tipi salvati</font>");
}
}
else
{
    echo("<font color='#CC0000' face='Verdana' size='2'>Errore</font>");
}
}

```

Quando l'utente clicca su AVANTI inserisce il preventivo e memorizza il nome del preventivo

```
$_SESSION["nomePreventivo"]=$_POST["nomePreventivo"];
```

e con la funzione `header()` va alla pagina `cliente.php`

```
header("Location: cliente.php");
```

La pagina `eliminaTipo.php` mostra una tabella con l'elenco dei tipi salvati (figura 15)



Nome	Operazione
Edilizio	X
Elettrico	X

[Indietro](#)

Figura 15

Ad esempio, cliccando sul pulsante di cancellazione corrispondente ad Edilizio va al link:

```
eliminaTipo.php?eliminare=Edilizio
```

Accede alla stessa pagina e se esiste `$_GET["eliminare"]`, elimina il tipo.

```
if(isset($_GET["eliminare"]))  
{  
    include("connessione.php");
```

```

$query="delete from tipi where nome='$_GET[eliminare]'";
$result=mysql_query($query, $conn);
if($result)
    {echo("<p align='center'><font color='#00CC00' face='Verdana'
size='3'>Tipo Eliminato</font></p>");
    }
else
    {
        echo("<p align='center'><font color='#CC0000' face='Verdana'
size='3'>Impossibile eliminare il Tipo</font></p>");
    }
}

```

Le pagine cliente.php ed eliminaCliente.php sono molto simili a nuovoPreventivo.php e eliminaTipo.php.



Figura 16 e 17

Nei campi vuoti è stato inserito “-”

```

if($indirizzo==""){ $indirizzo="-"; }
if($provincia==""){ $provincia="-"; }

```

```
if($citta==""){ $citta="-"; }  
if($cap==""){ $cap="-"; }
```

Nella pagina nuovoCliente.php il committente ha specificato che vuole identificare il cliente con codice fiscale o partita iva o cognome e nome.

Per fare questo lo script è:

```
if($_POST["codiceFiscale"]=="" && $_POST["partitaIva"]=="" &&  
$_POST["ragioneSociale"]=="" && ($_POST["nome"]=="" ||  
$_POST["cognome"]==""))  
{  
    echo("<p align='center'><font color='#CC0000' face='Verdana'  
size='3'> &nbsp;  Introduci almeno il Codice Fiscale o la Partita Iva o la Ragione  
Sociale o il Nome e il Cognome</font></p>");  
}
```

Quindi, ad esempio, se inserisco solo il nome viene visualizzato l'errore (figura 18)

The screenshot shows a web form titled "Nuovo Cliente" with a logo for "Rosato & Meroni" (Ingegneria generale e servizi tecnici). Below the title, there is a message: "Introduci almeno il Codice Fiscale o la Partita Iva o la Ragione Sociale o il Nome e il Cognome". The form contains several input fields: "Codice Fiscale", "Partita IVA", "Ragione Sociale", "Nome", "Cognome", "Indirizzo", "Provincia", "Città", "CAP", "Telefono", and "E-mail". A "SALVA NUOVO CLIENTE" button is at the bottom, along with a small "Indietro" link.

Figura 18

Nella pagina cantiere.php viene prelevato il codiceCliente (la chiave primaria della tabella clienti)

```
$codiceCliente=$_SESSION["codiceCliente"];
```

e successivamente vengono mostrati in un menù a tendina tutti i cantieri utilizzando la funzione `mysql_fetch_array()` associati al codiceCliente prelevato.

Il risultato è mostrato in figura 19



Figura 19

`nuovoCantiere.php` e `eliminaCantiere.php` sono molto simili a `nuovoCliente.php` e `eliminaCliente.php`



### Nuovo Cantiere



---

Indirizzo

Provincia

Città

CAP

[Indietro](#)

### Elimina Cantiere



---

Indirizzo	Provincia	Città	CAP	Operazione
Via Fulvio Testi, 2	-	Padova	-	<input checked="" type="checkbox"/>

[Indietro](#)

Figura 20 e 21

L'elenco dei cantieri o l'aggiunta di un nuovo cantiere è sempre associato ad un cliente.

Ad esempio la query che aggiunge il cantiere è:

```
$query="insert into cantieri (indirizzo, provincia, citta, cap, codicecliente) values
('$_POST[indirizzo]', '$_POST[provincia]', '$_POST[citta]', '$_POST[cap]',
$_SESSION[codiceCliente)";
```

La pagina righePreventivo.php contiene una casella di testo per cercare le voci computo (figura 22).



Figura 22

In questo caso, quando l'utente introduce del testo nella casella di testo e preme Cerca viene eseguita la seguente stringa sql contenente LIKE:

```
$stringaSql="select nome from vocicomputo where nome like '%  
$_POST[nomeVoceComputo]%'";
```

La condizione LIKE è un operatore logico che può essere utilizzato nella condizione WHERE, utile per effettuare le ricerche su dati parziali.

L'operatore LIKE utilizza due tipi di caratteri jolly il percento '%' e l'underscore '\_'. Il primo carattere jolly sostituisce un insieme di caratteri, il secondo si limita a sostituirne uno solo.

Il risultato è un elenco di voci computo che contengono all'interno la stringa inserita nella casella di testo (figura 23)



Figura 23

Cliccando sul pulsante aggiungi relativo alla voce computo, la voce computo viene introdotta nel preventivo (figura 24).



Figura 24

EliminaVoceComputo.php e nuovoFornitore.php seguono la stessa logica delle pagine viste precedentemente.

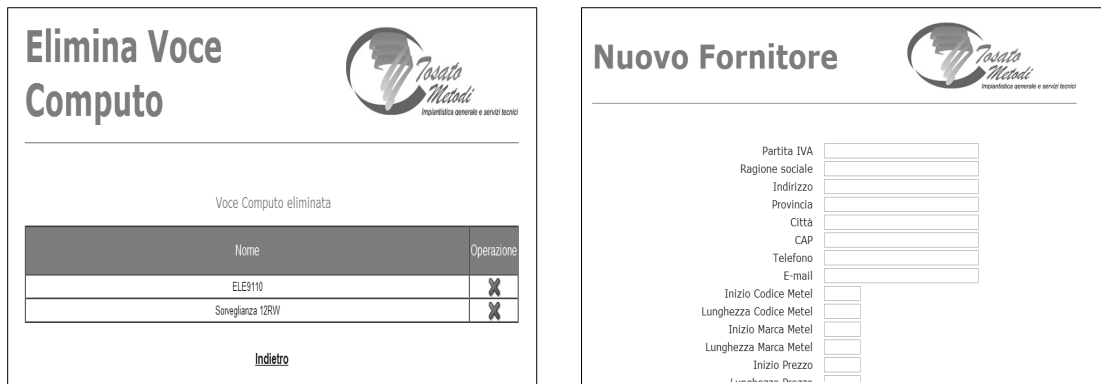


Figura 25 e 26

Dopo l'inserimento di un nuovo fornitore si viene automaticamente indirizzati alla pagina relativa all'aggiornamento dei prodotti (tramite funzione header()).

Ora, è necessario caricare il file metel corrispondente al fornitore.

In questo caso il form ha la particolarità che l'attributo enctype deve essere posto uguale a multipart/form-data perché indica il passaggio di un file da parte del form.

E' stato fissato il tempo massimo di esecuzione a 100000 (altrimenti non sarebbe stato in grado di caricare tutti i prodotti).

```
set_time_limit(100000);
```

Il file viene salvato attraverso la funzione copy() nella cartella archiviMetel

```
copy($_FILES["file"]["tmp_name"], "archiviMetel/" .
$_GET['codiceFornitore'] . ".txt");
```

ne viene verificata l'esistenza del file e in caso positivo procede con:

```
$codiceMetel=substr($lines[$i], $codiceMetelInizio-1, $codiceMetelLunghezza);
    $query="select prezzo from prodotti where
codiceMetel='$codiceMetel'";
    $result=mysql_query($query, $conn);
    if($result)
    {
        $row=mysql_fetch_array($result);
        $prezzoD=$row["prezzo"];
        $prezzo=str_replace(",",".",substr($lines[$i],
$prezzoInizio-1, $prezzoLunghezza));

        if($separatore==0)
        {
            $lunghezza=strlen($prezzo);
            $sottoStringa2=substr($prezzo,-
1*$decimaliPrezzo,$decimaliPrezzo);
            $sottoStringa1=substr($prezzo,-
1*$lunghezza,$lunghezza-$decimaliPrezzo);
            $prezzo=$sottoStringa1.".".
$sottoStringa2;
        }

        if(empty($prezzoD))
        {
```

```

                                $descrizione=substr($lines[$i],
$descrizioneInizio-1, $descrizioneLunghezza);
                                $descrizione=str_replace("'", "",
$descrizione);
                                $descrizione=str_replace("''", "'",
$descrizione);

                                $q1="insert into prodotti (codiceMetel,
marcaMetel, prezzo, descrizione, unitaMisura, codiceFornitore) values
('".substr($lines[$i], $codiceMetelInizio-1,
$codiceMetelLunghezza)."', '".substr($lines[$i], $marcaMetelInizio-1,
$marcaMetelLunghezza)."', $prezzo, '$descrizione', '".substr($lines[$i],
$unitaMisuraInizio-1, $unitaMisuraLunghezza)."', $codiceFornitore)";

                                $r1=mysql_query($q1, $conn);
                                if(!($r1)){echo("<p align='center'><font
color='#CC0000' face='Verdana' size='3'>Errore Inserimento</font></p>");}
                                }
                                else
                                {
                                        if($prezzo<$prezzoD)
                                        {
                                                $q2="update prodotti set
prezzo=$prezzo, codiceFornitore=$codiceFornitore where
codiceMetel='$codiceMetel'";

                                                $r2=mysql_query($q2, $conn);
                                                if(!($r2)){echo("<p align='center'><font
color='#CC0000' face='Verdana' size='3'>Errore Aggiornamento</font></p>");}
                                                }
                                        }
                                }
}

```

Quindi per ogni prodotto (identificato dal codiceMetel) viene ricavato il prezzo. Se non esiste (quindi non esiste il prodotto) il prodotto viene aggiunto, se esiste controlla se il prezzo è minore e in caso affermativo aggiorna il prodotto con il prezzo del nuovo fornitore.

La funzione `substr()` restituisce una porzione della stringa, in base al secondo parametro che indica l'inizio della porzione da estrarre, e il terzo parametro, che indica quanti caratteri devono essere estratti. Se il terzo parametro non viene indicato, viene restituita tutta la parte finale della stringa a partire dal carattere indicato.

La funzione `str_replace()` è utilizzata in questo caso per eliminare “” perché nell'inserimento del record “” avrebbe generato errore.

Al termine

```
unlink("archiviMetel/".$codiceFornitore.".txt");
```

Elimina il file importato.

La pagina `eliminaFornitore.php` è molto simile alle pagine viste in precedenza

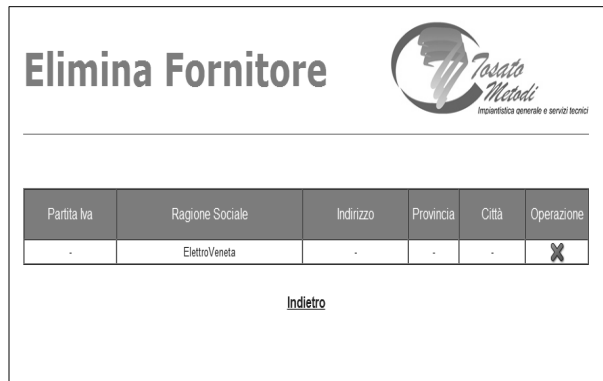


Figura 27

tuttavia quando viene eliminato un fornitore non basta eliminare il fornitore, occorre aggiornare anche la tabella prodotti:

```
$query2="update prodotti set codiceFornitore=0, prezzo=9999999999.99999 where codiceFornitore=$_GET[eliminare]";
```

All'interno del database, nella tabella fornitori, con codiceFornitore 0 c'è un'azienda fittizia con ragione sociale “altri fornitori” che viene esclusa dalle query quando vengono mostrati i fornitori.

Questo mi è utile successivamente quando dovrò mostrare l'elenco dei prodotti da ordinare da “altri fornitori”.

Infine il prezzo viene settato a 9999999999.99999 (il numero più elevato che ci possa essere in quel campo). In questo modo quando viene caricato un nuovo fornitore, come visto prima, il software esegue il confronto relativo al prezzo e aggiorna il codiceFornitore.



La soluzione usata è una delle tante possibili ad esempio avrei potuto creare un constraint di chiave esterna con la clausola ON DELETE SET NULL e successivamente, nell'importazione del file metel, aggiornare i prodotti con codiceFornitore NULL.

La pagina nuovaVoceComputo.php inserisce la voce computo nella tabella voci computo e passa alla pagina prodotto.php il nome appena inserito.



Figura 28

```
if(isset($_POST["nomeVoceComputo"]))
{
    $query="insert into vocicomputo (nome) values
('$_POST[nomeVoceComputo]')";
    $result=mysql_query($query, $conn);
    if($result)
    {
```

```
header( "Location: prodotto.php?
nome=$_POST[nomeVoceComputo]" );
}
else
{
echo("<p align='center'><font color='#CC0000' face='Verdana'
size='2'>Errore</font></p>");
}
}
```



The screenshot shows a web page titled "Prodotto". At the top right, there is a logo for "Tosato Metodi" with the tagline "Ingegnistica generale e servizi tecnici". Below the logo, the text "Sorveglianza 339" is displayed. The main content area contains a search form with three input fields labeled "Codice", "Marca", and "Descrizione", followed by a "Cerca" button. At the bottom center of the form area, there is a link labeled "Indietro".

Figura 29

Dalla pagina prodotto.php, l'utente può aggiungere i prodotti associati a questa voce computo.

Può cercare per codice, marca o descrizione.

Per far questo ho creato una stringa che in base ai parametri inseriti crea la query.

Il `$c` è un metodo per evitare che nella stringa query ci siano “and” di seguito o “and” immediatamente dopo il where. Inoltre se l'utente inserisce il codice metel e altri parametri, viene considerato solo il codice metel.

```
$stringaSqlA="select count(codiceMetel) as numero from prodotti where ";
$stringaSql="select codiceMetel, marcaMetel, descrizione, unitaMisura, prezzo from
prodotti where ";
    if($_POST["codiceMetel"]!="")
    {
        $stringaSql=$stringaSql."
codiceMetel='$_POST[codiceMetel]';
        $stringaSqlA=$stringaSqlA."
codiceMetel='$_POST[codiceMetel]';
    }
    else
    {
        if($_POST["marcaMetel"]!="")
        {
            $stringaSql=$stringaSql."
marcaMetel='$_POST[marcaMetel]';
            $stringaSqlA=$stringaSqlA."
marcaMetel='$_POST[marcaMetel]';
            $c=1;
        }
        if($_POST["descrizione"]!="")
        {
            if($c==1)
            {
```

```

$stringaSql=$stringaSql." and ";
$stringaSqlA=$stringaSqlA." and ";
}
$stringaSql=$stringaSql." descrizione like '%
$_POST[descrizione]%'";
$stringaSqlA=$stringaSqlA." descrizione like '%
$_POST[descrizione]%'";
}
}

```

Ad esempio se inserisco 3mi come marca e ss in descrizione ottengo

Codice	Marca	Descrizione	Um	Prezzo	Operazione
XE003411453	3MI	TEMFLEX NASTRO ROSSO	PZ	0.34994	

Figura 30

Ora è possibile specificare la Qtà di prodotto necessario, le ore di manodopera e il prezzo di manodopera all'ora. Questi valori sono utili per generare la scheda costi e il preventivo.

Dopo aver introdotto tutte le voci computo, la pagina successiva è altriCosti.php

Nome costo	Costo unitario	Quantità
Preparazione	0,00	0 Ore
Collaudo	0,00	0 Ore
Redazione documenti	0,00	0 Ore
Carburante	0,00	0 Km
Pedaggio	0,00	0
Alloggio	0,00	0
Vitto	0,00	0
Viacchini	0,00	0 Ore

Figura 31

Il committente ha specificato che “altri costi” contiene tutti i costi utilizzati dall'azienda e che non è necessario gestire l'inserimento o l'eliminazione di costi, quindi non ho creato una tabella nel database altriCosti, le informazioni sono memorizzate nella tabella preventivi.

Infine, le ultime informazioni richieste sono la percentuale utile e la percentuale spese generali nella pagine ultimiValori.php

Percentuale utile

Percentuale per calcolo spese generali

Figura 32

Questi valori mi permettono di calcolare il valore dell'offerta al cliente (valoreOfferta) con la seguente formula:

$$Stot = StMad + StMat + Sta;$$

$$SvaloreOfferta = (-100 * Stot) / (SpersentualeUtile - 100 + SpersentualeSpeseGenerali);$$

Dove \$tMad è il totale dei costi della manodopera, \$tMat è il totale dei costi materiali e \$ta è il totale degli altri costi.

### 3.4 DOCUMENTI

Nella pagina fine ho tutti i documenti generati:

- scheda costi
- preventivo
- ordini divisi per fornitori



Figura 33

La scheda costi deve essere in formato excel.

Ho creato quindi la pagina schedaCosto.php e all'inizio della pagina ho specificato

```
header ("Content-Type: application/vnd.ms-excel");  
header ("Content-Disposition: inline; filename=SchedaCosti.xls");
```

In questo modo quando l'utente clicca su Scheda Costi, la pagina PHP viene trasformata in un file excel.

Per quanto riguarda invece Preventivo, ho creato la pagina preventivo.php e all'inizio ho specificato

```
header("Content-type: application/vnd.ms-word");  
header("Content-Disposition: attachment; Filename=Preventivo.doc");
```

Per quanto riguarda l'ordine ho utilizzato la classe FPDF che permette di generare file pdf con codice PHP:

```
define('FPDF_FONTPATH','pdf/font/');  
include('pdf/fpdf.php');  
$p = new fpdf();  
$p->Open();  
$p->AddPage();  
$p->SetTextColor(0);  
$p->SetFont('Arial', "", 30);
```

Con la prima istruzione assegno alla costante FPDF\_FONTPATH il valore “*pdf/font/*”, con le altre funzioni, ho creato l'oggetto dalla classe FPDF (istanza della classe) e ho impostato il font e il colore.

Per scrivere ho utilizzato Write()

```
$p->Write(7,"Ordine ".$nomeFornitore."\n\n");
```

Al termine, per generare il file in pdf

```
$p->output();
```

Dopo aver generato i documenti, è comunque possibile tornare indietro e modificare le percentuali, aggiungere o togliere voci computo, ecc.

Se sono stati eliminati dei fornitori e ci sono dei prodotti inseriti nel preventivo con codiceFornitore=0 non è possibile generare il preventivo e la scheda costi e compare l'ordine da effettuare, se l'utente torna indietro e aggiunge un fornitore che offre i prodotti inseriti nel preventivo, è possibile generare il preventivo e la scheda costi.





Figura 34

### 3.5 REPORTISTICA

E' possibile poi generare una reportistica dei preventivi effettuati:



Figura 35

Per formattare correttamente la data (giorno-mese-anno a differenza di anno-mese-giorno) ho utilizzato la funzione explode()


```
list($anno1, $mese1, $giorno1)=explode("-", $data1);
```

```
list($anno2, $mese2, $giorno2)=explode("-", $data2);
```

La data inserita di default è la data del primo preventivo inserito e dell'ultimo preventivo inserito.

Cliccando su elabora compare l'elenco dei preventivi con le informazioni principali e il valore massimo minimo e medio del valore economico.

## Reportistica



DA:    - A:

Tipo di contratto:  Elettrico  **Elabora**

Codice fiscale	Ragione Sociale	Partita Iva	Nome	Cognome	Preventivo	Data	Valore Economico
-	ArtusiWeb	-	Ingres	Artusi	P1	04-08-2011	893.30
-	ArtusiWeb	-	Ingres	Artusi	ADDF	21-08-2011	656.25

Valore Economico= max: 893.30 - min: 656.25 - media: 774.78

[Indietro](#)

Figura 36

# Appendice

## A.1 IL MODELLO ER

Il modello Entity-Relationship (anche detto modello ER) è un modello per la rappresentazione concettuale dei dati.

I costrutti principali del modello sono: entità, associazioni e attributi.

Le entità rappresentano le classi di oggetti (fatti, cose, persone,...) che hanno proprietà comuni ed esistenza autonoma. Un'occorrenza è un oggetto della classe.

In uno schema ogni entità ha un nome che la identifica univocamente e viene rappresentata graficamente tramite un rettangolo con il nome dell'entità all'interno.

Le associazioni rappresentano un legame tra due o più entità. Il numero di entità legate è indicato dal grado dell'associazione. Di norma viene rappresentata graficamente da un rombo contenente il nome dell'associazione. Il nome può essere un sostantivo in modo da non dare una direzione di lettura. Infine, un'entità è descritta usando una serie di attributi. Tutti gli oggetti della stessa classe entità hanno gli stessi attributi. Per ciascuna classe entità, dobbiamo definire anche una chiave. La chiave è un insieme di attributi che

identificano univocamente una tupla all'interno del database. L'attributo si identifica con un cerchio dove viene specificato il nome dell'attributo o anche semplicemente, nel caso di diagrammi complessi, indicandone solo il nome. La chiave primaria viene sottolineata.

## **A.2 I CONCETTI BASE DI PHP**

### **A.2.1 Get e post**

La principale vantaggio del web dinamico è la possibilità di variare i contenuti delle pagine in base alle richieste degli utenti. Questa possibilità si concretizza attraverso i meccanismi che permettono agli utenti, oltre che di richiedere una pagina ad un web server, anche di specificare determinati parametri che saranno utilizzati dallo script PHP.

Esistono due sistemi per passare dati ad uno script: il metodo GET e il metodo POST.

Il metodo GET consiste nel concatenare i dati all'indirizzo della pagina richiesta, dopo il nome della pagina c'è un punto interrogativo e dalle coppie nome e valore separati dal segno '&'. Ad esempio, avendo la pagina prodotto.php passandole la categoria e il numero del prodotto stesso, dovremo richiamare la pagina in questo modo:

```
<a href="prodotto.php?categoria=a7&numero=2">
```

Quando la pagina `prodotto.php` viene richiamata in questo modo, essa avrà a disposizione, al suo interno, le variabili `$_GET['categoria']` (con valore 'a7') e `$_GET['numero']` (con valore '2').

Il metodo POST viene utilizzato con i moduli: quando una pagina HTML contiene un tag `<form>`, uno dei suoi attributi è `method`, che può valere GET o POST. Se il metodo è GET, i dati vengono passati come prima. Se il metodo è POST, i dati vengono invece inviati in modo invisibile per l'utente.

I dati che vengono passati attraverso il metodo POST sono memorizzati nell'array `$_POST`.

```
<form action="prodotto.php" method="post">
  <input type="text" name="categoria">
  <input type="text" name="numero">
  <input type="submit" name="submit" value="invia">
</form>
```

Questo modulo contiene semplicemente due caselle di testo (categoria e numero), poi c'è il tasto che invia i dati, attraverso il metodo POST, alla pagina `prodotto.php`.

Nella pagina `prodotto.php` c'è a disposizione la variabile `$_POST['categoria']` e `$_POST['numero']`, contenente il valore che l'utente ha digitato nel campo di testo.

### **A.2.2 Connessione database**

Perché un'applicazione realizzata in PHP possa utilizzare le informazioni contenute all'interno di un database deve connettersi ad esso.

Per aprire una connessione da un'applicazione in PHP a un database MySQL, si utilizza la funzione `mysql_connect()` che in caso di errore restituisce `FALSE`;

Questa funzione richiede il passaggio di tre parametri:

- `hostname`: è il nome dell'host, può essere espresso sotto forma di indirizzo IP. Nel caso di un'installazione locale l'hostname è "localhost".
- `username`: è il nome dell'utente abilitato alla connessione e alla manipolazione di uno o più database; MySQL prevede un utente iniziale `root`.
- `password`: per questioni di sicurezza è possibile associare una password ad ogni nuovo utente MySQL creato.

Il file `connessione.php` contiene

```
$conn=mysql_connect("localhost","root","");  
$db=mysql_select_db("softwarePreventivazione", $conn);
```

e nelle pagine in cui viene utilizzato il database:

```
include("connessione.php");
```

`mysql_query` e `mysql_fetch_array()`

In PHP l'inserimento o la cancellazione dei dati all'interno di una tabella è possibile passando come parametro alla funzione `mysql_query()` un'istruzione SQL di inserimento o cancellazione.

La funzione `mysql_fetch_array()` invece archivia i risultati di un'interrogazione all'interno di un vettore.

`mysql_fetch_array()` ha come parametro una query di selezione passata alla funzione `mysql_query()`; una query di selezione è formata da un `SELECT` a cui devono seguire i nomi dei campi o in alternativa (\*) che indica tutti i campi presenti nelle tabelle coinvolte.

### **A.2.3 Sessioni**

Con le sessioni, il server apre una sessione con la funzione `session_start()` e gli assegna un nome univoco.

Le variabili di sessione sono memorizzate dentro un array globale di nome `$_SESSION[]`.

Deposita un cookie nella macchina dell'utente. Ogni volta che ne ha bisogno dal cookie estrae i dati per risalire alla sessione e al file salvato.

Ogni volta che interrompiamo il collegamento, il server cancella tutto.

#### **A.2.4 Upload**

Con PHP, c'è la possibilità di gestire gli Uploads da un computer-client verso un computer-server.

Upload indica in breve l'operazione di trasferimento di uno o più files da un client verso un server, dove il trasferimento viene effettuato tramite un form con metodo POST.



## Conclusioni

Lo stage è stato svolto presso Tosato Metodi da Maggio ad Agosto 2011.

Il progetto ha richiesto la progettazione e la realizzazione di un'applicazione di basi di dati online per la preventivazione.

Il progetto e' stato portato a termine con successo raggiungendo l'obiettivo iniziale nei tempi previsti. Lo sviluppo e la progettazione dell'applicazione sono stati condotti secondo la metodologia di progettazione di basi di dati che viene descritta in letteratura.

## Bibliografia

- Paolo Atezeni, Stefano Certi, Stefano Paraboschi, Riccardo Torlone, Basi di dati – modelli e linguaggi di interrogazione, McGraw-Hill

## Siti consultati

- [www.html.it](http://www.html.it)
- [it.wikipedia.org](http://it.wikipedia.org)

## Ringraziamenti

Ringrazio Enrico Tosato per l'opportunità che mi ha dato di svolgere lo stage presso la sua azienda, il Prof. Melucci Massimo che mi ha seguito nella stesura della tesi e infine un grande grazie ai miei genitori e ai miei compagni di studio.