

*Colucci Carmine Antonio – 961/087*  
*D'Apolito Filomeno – 961/081*  
*Ferrara Francesco Saverio – 961/079*

# **Consorzio di serre**

## **“Consorzio Natura Viva”**

(consegnato il 23/11/2006)

### **Breve descrizione del problema**

Quest'applicazione web permetterà a più serre di consorziarsi tra loro per poter offrire ai clienti un servizio rapido on-line di acquisto di prodotti biologici. L'utente potrà acquistare prodotti, e l'applicazione automaticamente sceglierà le serre più vicine per minimizzare i costi di spedizione. Questo rappresenta uno dei punti cruciali del problema, in quanto ogni serra dovrà tener traccia di tutte le distanze dalle altre. In base alle distanze tra le serre, alla disponibilità di prodotti e all'ordine del cliente, l'applicazione dovrà calcolare una scelta intelligente: il cliente stesso deve manualmente scegliere la serra a lui più vicina.

I tipi di utente che interagiranno con l'applicazione sono:

- CLIENTE: cliente generico, che consulta il sito e può essere registrato o meno;
- PROPRIETARIO: proprietario di una o più serre;
- AMMINISTRATORE: amministratore del consorzio.

Di seguito riportiamo gli USE-CASE previsti per questi tipi di utenti.

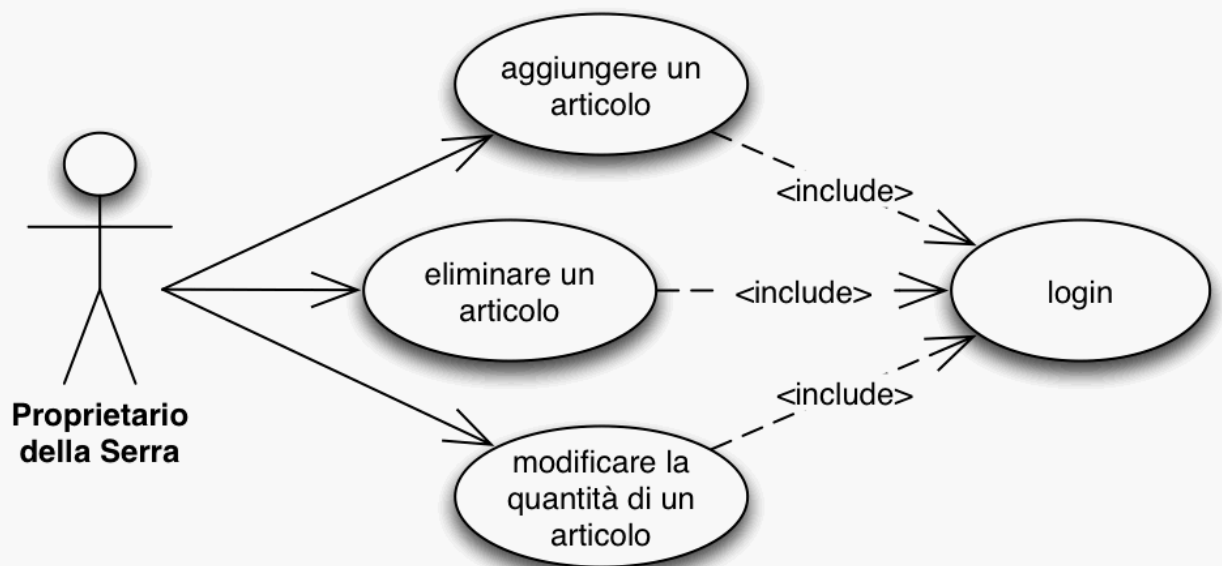
## USE-CASE per l'utente "Cliente"



### Descrizione:

Non ci aspettiamo competenza tecniche per l'utente "cliente", quindi la sua interfaccia deve essere molto semplice ed intuitiva, specialmente in fase di acquisto quando viene chiesto di scegliere la serra più vicina alla sua posizione geografica: una risposta errata comprometterebbe l'ottimizzazione dell'ordine.

## USE-CASE per il "Proprietario della Serra"

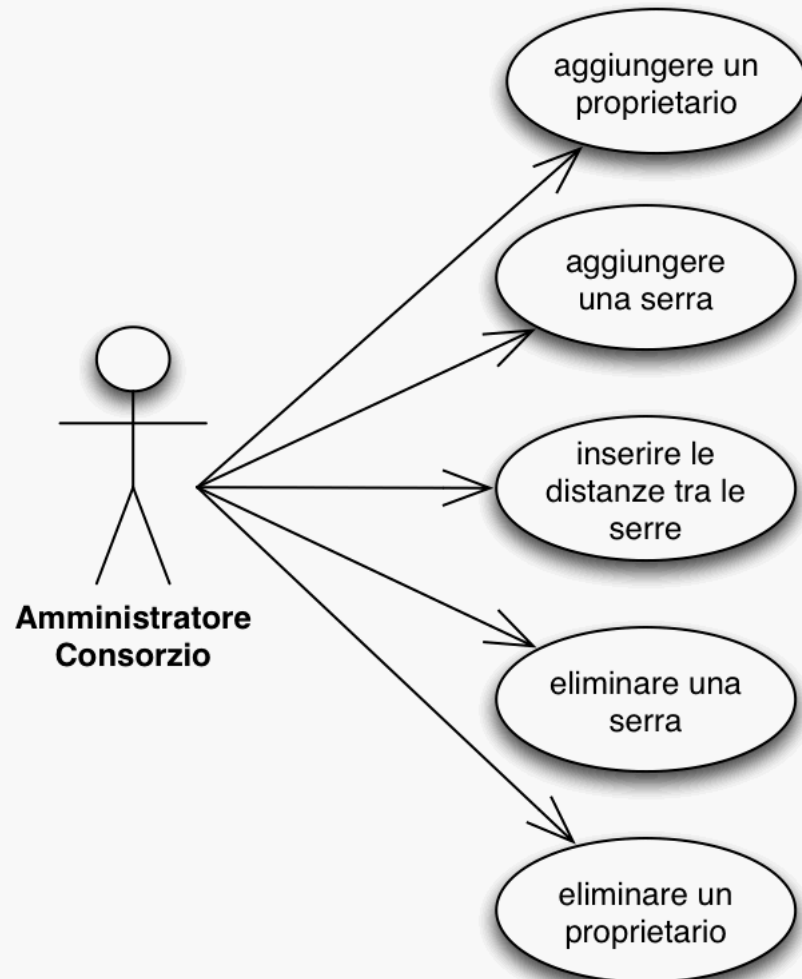


### Descrizione:

Non ci aspettiamo particolari competenze tecniche da parte dell'utente "Proprietario della Serra".

In particolare la sua interfaccia grafica deve semplificare i suoi compiti di gestione della giacenza degli articoli nella sua serra: un inserimento di dati errati porterebbe a conseguenze inaspettate come la vendita di articoli esauriti.

## USE-CASE per "Amministratore Consorzio"



Descrizione:

"Amministratore del Consorzio" è l'utente con maggiori competenze tecniche. La sua interfaccia prevede funzionalità per aggiungere o eliminare una serra dal consorzio.

## Tecnologie usate

Per quest'applicazione sono state usate due tecnologie. La parte relativa al cliente e proprietario è sviluppata con le servlets Java, mentre la parte relativa all'amministratore del consorzio è sviluppata con la tecnologia CGI.

La prima tecnologia è senz'altro portabile e sicura. Tuttavia esistono valide alternative per il tipo di applicazione da implementare quali JSP, PHP, e ASP. Dovendo scegliere preferiamo JSP o PHP che sono portabili, e permettono di immergere del codice direttamente nel codice HTML, rendendo lo sviluppo veloce e semplificato. Usando la tecnologia ASP, adotteremo una soluzione simile, ma il problema principale riguarda la non portabilità, per cui si dovrebbe usare un web server proprietario.

Le servlets, invece, permettono un maggior controllo sull'applicazione rispetto a JSP o PHP, ma le stesse procedure sono scritte con un codice più lungo comportando uno sviluppo più lento. Questo maggior controllo non è richiesto per l'applicazione in questione, tuttavia scegliamo questa tecnologia poiché ci è stata imposta.

L'altra implementazione adotta la tecnologia CGI (Common Gateway Interface). Qui abbiamo una vasta gamma di scelta per quanto riguarda il linguaggio di programmazione da adottare: ad esempio C, C++, script Shell, Perl, ecc...

In questo caso la soluzione ottimale è il linguaggio di scripting Perl. Esso è molto usato in questo ambito, poiché ha molte funzionalità adatte a elaborare pagine web, ad esempio offre un ottimo supporto per la manipolazione di stringhe.

Purtroppo non conosciamo il linguaggio Perl, e sceglierlo ci porterà a un ritardo sulla consegna dell'applicazione. Optiamo per il linguaggio C, con il quale abbiamo già esperienza.

La maggior parte dell'informazione utile all'applicazione è

memorizzata in un database. Tra i vari DBMS, scegliamo mysql, poiché è una soluzione portatile e, specialmente, non proprietaria. Inoltre è possibile utilizzare MySQL con connettori JDBC oppure ODBC che ci permette di astrarre dal particolare DBMS rendendo l'applicazione indipendente da esso. Quindi in futuro sarà possibile cambiare la tecnologia o l'implementazione dell'applicazione, oppure cambiare il DBMS associato.

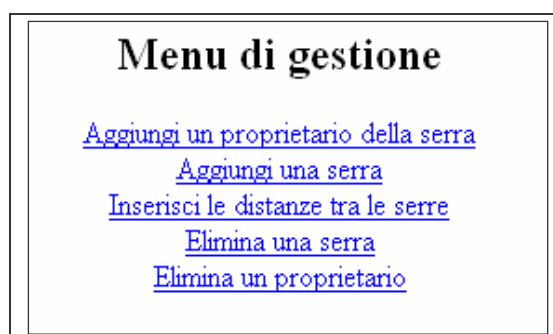
In allegato a questo documento vi è lo schema E/R del database usato.

## Analisi del problema

### Sezione Amministratore del consorzio (CGI)

L'amministratore dell'intero consorzio è quella persona che ha più competenze informatiche di tutti gli altri e, come abbiamo già visto, ha svariati compiti.

Le diverse funzioni che l'Amministratore può eseguire sono riassunte nel seguente menù:



#### 1. Aggiunta di un proprietario:

Per l'aggiunta di un nuovo proprietario, l'Amministratore deve memorizzare all'interno dell'archivio i seguenti dati (tutti obbligatori):

*Codice Fiscale, Nome, Cognome, Città, Provincia, CAP, Password.*

All'evento *submit* del pulsante "Inserisci", presente nel form, viene richiamato il modulo CGI *addproprietario* passando tutti i parametri precedentemente immessi tramite POST.

Il POST viene parserizzato e quindi inizializzato in un array di lunghezza `CONTENT_LENGTH+1` tramite una funzione chiamata *init\_param* (che ritornerà al programma chiamante questo array).

Nel caso in cui vi sia l'impossibilità di ricavare i parametri passati dal post sarà stampato un messaggio di errore, altrimenti viene effettuato un controllo (lato server tramite la funzione *find\_n\_param*) sul corretto tipo di dato e lunghezza che ogni

parametro può assumere.

Il successivo step è quello di inserire questi dati nel database tramite un'istruzione SQL di tipo INSERT. La chiave primaria per ogni Proprietario è il Codice Fiscale perciò vi è la possibilità di ricevere un errore qualora tentassimo di duplicarla.

## **2. Aggiunta di una serra:**

Per l'aggiunta di una nuova serra, l'Amministratore oltre a specificare i seguenti dati (tutti obbligatori):

*Nome, Indirizzo, Città, Provincia, CAP*

deve anche provvedere ad "assegnare" questa nuova serra ad un Proprietario precedentemente inserito.

All'evento *submit* del pulsante "Inserisci", presente nel form, viene richiamato lo stesso modulo CGI, questa volta con POST diverso da NULL contenente i parametri precedentemente immessi.

Il POST viene parserizzato e quindi inizializzato in un array di lunghezza `CONTENT_LENGTH+1` tramite una funzione chiamata *init\_param* (che ritornerà al programma chiamante questo array).

Nel caso in cui vi sia l'impossibilità di ricavare i parametri passati dal POST sarà stampato un messaggio di errore, altrimenti viene effettuato un controllo (lato server tramite la funzione *find\_n\_param*) sul corretto tipo di dato e lunghezza che ogni parametro può assumere.

Il successivo step è quello di inserire questi dati nel database tramite un'istruzione SQL di tipo INSERT. La chiave primaria per ogni Serra è il "Nome" per questo vi è la possibilità di ricevere un errore qualora tentiamo di duplicarla.

## **3. Inserimento distanze tra serre:**

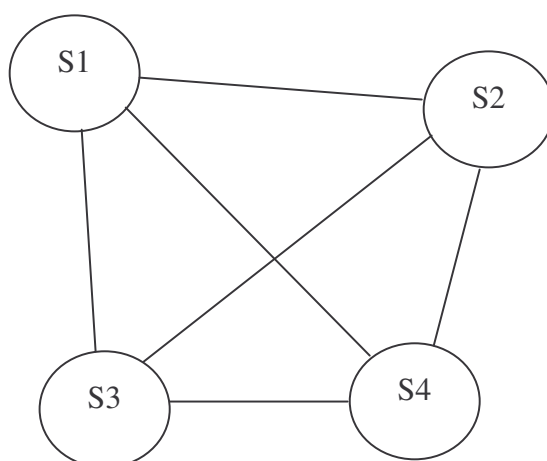
Per specificare le distanze tra le serre, l'Amministratore deve selezionare la serra di partenza e quella di arrivo.



Per default la distanza tra una stessa serra è settata a '0'.

Se la distanza tra due serre è già stata specificata e si tenta di reinserirla, il vecchio valore (in Km) verrà sovrascritto.

Per semplificarci il progetto, ogni volta che viene inserita una serra, supponiamo che l'amministratore inserisce tutte le distanze tra le altre serre, quindi il grafo risultante è sempre completo e non orientato.



#### **4. Eliminare una serra:**

Per specificare una serra da eliminare, l'Amministratore deve selezionarla dalla lista di tutte le serre facenti parte del consorzio.

All'evento *submit* del pulsante "Elimina", presente nel form, viene richiamato lo stesso modulo CGI, questa volta con POST diverso da NULL contenente il nome della serra da eliminare.

Prima di procedere definitivamente alla cancellazione della serra, viene controllato che ad essa non sono associati ancora ordini non evasi.

#### **5. Eliminare un Proprietario:**

Per specificare il proprietario da eliminare, l'Amministratore deve

selezionarlo dalla lista di tutti i proprietari del consorzio.

All'evento *submit* del pulsante "Elimina", presente nel form, viene richiamato lo stesso modulo CGI, questa volta con POST diverso da NULL contenente il nome del proprietario da eliminare.

Prima di procedere definitivamente alla cancellazione del proprietario, viene controllato che ad esso non siano associate una o più serre.

## Sezione Proprietario di serra (Servlets)

Il Proprietario, di una o più serre, è quella persona che non necessariamente ha competenze informatiche e deve avere un set di strumenti efficienti per gestire le proprie serre.

Egli accede al menù di gestione serra effettuando il login dal sito fornendo il suo username (codice fiscale) e una password.



The screenshot shows the 'Natura Viva' website interface. On the left, there is a large green banner with the text 'Natura Viva' in a white serif font and 'consorzio di serre' in a smaller black sans-serif font below it. To the right of the banner is a login form with the following elements: a label 'User:' above a text input field, a label 'Password:' above another text input field, and a button labeled 'Accedi' below the password field. Below the login form, there is a horizontal navigation bar with three buttons: 'Aggiungi un articolo', 'Elimina un articolo', and 'Modifica un articolo', all in blue text on a light green background.

Poiché http è un protocollo senza stato, ossia nessuna informazione riguardo al client viene immagazzinata dal server che risponde ad ogni richiesta sulla base di quella corrente, è stata implementata una tecnica di Session Tracking (variabili di sessione) che tiene traccia del Proprietario finchè il browser rimane aperto. Username e password vengono salvate nelle variabili di sessione in modo da non reinsere queste informazioni tutte le volte che si inoltra una richiesta.

L'ID della sessione (JSESSIONID) è memorizzato all'interno di un cookie. Se lo "user agent" non accetta i cookies oppure sono stati disabilitati dall'utente, il sito non attua la tecnica di Session Tracking e quindi non

ha il funzionamento che ci si aspetta, avvertendo l'utente con un opportuno messaggio.

Il menù relativo al Proprietario ha varie scelte che corrispondono alle operazioni che egli può compiere:

*Aggiungi un articolo, Elimina un articolo, Modifica un articolo*

## **1. Aggiunta di un articolo:**

Per l'aggiunta di un articolo alla serra, il Proprietario deve memorizzare all'interno dell'archivio i seguenti dati (tutti obbligatori):

*Nome, Ambiente, Ombra, Umidità, Cura, Prezzo*

All'evento *submit* del pulsante "Invia", presente nel form, viene richiamata la stessa servlet ("AddProdotto"), questa volta con POST diverso da NULL passando tutti i parametri precedentemente immessi.

I parametri vengono recuperati dal POST mediante il metodo `getParameter` dell'interfaccia `HttpServletRequest`.

Nel caso in cui vi sia l'impossibilità di ricavare i parametri passati dal post sarà stampato un messaggio di errore.

Dopo il recupero, un controllo sulla lunghezza di ogni parametro ne permetterà di verificarne la validità.

Per motivi di sicurezza, un ulteriore controllo viene fatto sul carattere " ' " che può provocare problemi di SQL INJECTION. In pratica viene sostituito il carattere " ` " con i due caratteri " ` ` ".

Il successivo step è quello di inserire questi dati nel database tramite un'istruzione SQL di tipo INSERT. La chiave primaria per ogni Prodotto è il *Nome* per cui vi è la possibilità di ricevere un errore qualora tentassimo di duplicarlo.

## 2. Eliminazione di un articolo:

Per eliminare un articolo dal Consorzio, il Proprietario deve selezionare il nome del Prodotto dalla lista dei prodotti.

All'evento *submit* del pulsante "Cancella", presente nel form, viene richiamata la stessa servlet ("DelProdotto"), questa volta con POST diverso da NULL passando il nome del prodotto da eliminare.

Il parametro viene recuperato dal POST mediante il metodo `getParameter` dell'interfaccia `HttpServletRequest`.

Nel caso in cui vi sia l'impossibilità di ricavare i parametri passati dal post sarà stampato un messaggio di errore.

Dopo il recupero, un controllo sulla lunghezza del parametro ne permetterà di verificarne la validità.

Per motivi di sicurezza, un ulteriore controllo viene fatto sul carattere " ' " che può provocare problemi di SQL INJECTION. In pratica viene sostituito il carattere " ` " con i due caratteri " ` ` ".

Il successivo step è quello di eliminare il prodotto.

Esso sarà effettivamente eliminato solo se non è presente in tutte le altre serre facenti parte del consorzio e se non è presente in nessun ordine non evaso.

## 3. Modifica della quantità di un articolo:

Per modificare la quantità di un articolo, il Proprietario deve selezionare la serra su cui effettuare la modifica, il nome del Prodotto di cui modificarne la quantità e la nuova quantità.

All'evento *submit* del pulsante "Modifica", presente nel form, viene richiamata la stessa servlet ("UpdateProdotto"), questa volta con POST diverso da NULL contenente: l'id della serra, il nome dell'articolo e la nuova quantità.

Questi parametri vengono recuperati dal POST mediante il metodo

getParameter dell'interfaccia HttpServletRequest.

Nel caso in cui vi sia l'impossibilità di ricavare i parametri passati dal post sarà stampato un messaggio di errore.

Dopo il recupero, un controllo sulla lunghezza dei parametri ne permetterà di verificarne la validità.

Per motivi di sicurezza, un ulteriore controllo viene fatto sul carattere " ' " che può provocare problemi di SQL INJECTION. In pratica viene sostituito il carattere " ` " con i due caratteri " ` ` ".

Il successivo step è quello di aggiornare la quantità del prodotto.

Se il prodotto è già presente nella serra, un UPDATE SQL ne permetterà l'aggiornamento della quantità altrimenti sarà eseguita un INSERT SQL che aggiungerà il prodotto alla serra con la relativa giacenza.

## Sezione Cliente (Servlets)

Il Cliente è quella persona che non necessariamente ha competenze informatiche. Esso potrà essere un cliente generico, che consulta il sito il quale potrà essere registrato o meno.

Dopo aver effettuato l'operazione di login (fornendo codice fiscale e password), sarà visualizzato il menù di gestione dove saranno possibili le seguenti opzioni:

<a href="#">Vetrina</a>	<a href="#">Carrello</a>	<a href="#">Registrati</a>
-------------------------	--------------------------	----------------------------

Poiché http è un protocollo senza stato, ossia nessuna informazione riguardo al client viene immagazzinata dal server che risponde ad ogni richiesta sulla base di quella corrente, è stata implementata una tecnica di Session Tracking (variabili di sessione) che tiene traccia del Proprietario finché il browser rimane aperto. Username e password vengono salvate nelle variabili di sessione in modo da non reinserire queste informazioni tutte le volte che si inoltra una richiesta.

L'ID della sessione (JSESSIONID) è memorizzato all'interno di un cookie. Se lo "user agent" non accetta i cookies oppure sono stati disabilitati dall'utente, il sito non attua la tecnica di Session Tracking e quindi non ha il funzionamento che ci si aspetta, avvertendo quindi l'utente con un opportuno messaggio.

E' importante sottolineare il fatto che anche se un utente non è loggato o registrato al sito avrà la possibilità di riempire comunque il proprio carrello, nel momento in cui deciderà di effettuare l'acquisto dovrà necessariamente o loggarsi (ed eventualmente registrarsi).

Per default la tipologia di utente e quindi il relativo menù di gestione è quello relativo al cliente.

Le informazioni riguardanti il carrello sono salvate in dei cookies, ragion per cui se l'user agent viene chiuso, lo stato del carrello non viene perso e ripristinato nel momento in cui un qualsiasi altro utente accederà in un secondo momento al sito.

## 1. Vetrina

Questa opzione permetterà all'utente di visionare tutti i prodotti offerti dal consorzio. Per ogni articolo è disponibile anche la foto per dettaglio più completo. Alla pressione del pulsante "Dettagli" saranno disponibili informazioni più approfondite sul prodotto quali: *Nome, Ambiente, Ombra, Umidità, Cura, Prezzo* e la quantità totale del prodotto in tutto il consorzio.



Nella modalità dettagli sarà possibile aggiungere qualsiasi prodotto

nel proprio carrello specificandone la quantità.

Nome: **Arancio**

Ambiente: **Mediterraneo**

Ombra: **A volte sì e a volte no**

Umidità: **dal 30% al 31%**

Cura: **Potatura nel mese di novembre**

Prezzo: **30.00 euro**

Sono disponibili 994 prodotti

Quantità di articoli desiderata:

La medesima vetrina è reperibile nella home page del sito in questione.

## 2. Carrello

Scegliendo questa opzione il cliente potrà tener traccia in ogni momento dello stato del proprio carrello.

Nella modalità dettagli, sarà possibile aggiungere qualsiasi prodotto nel proprio carrello specificandone la quantità.

Il carrello viene prelevato dalla sessione e sincronizzato tramite cookies. Ovviamente se i cookies non sono presenti il carrello sarà vuoto.

I cookies hanno una validità di 7 giorni dopodichè scadono e vengono eliminati dal browser, provocando uno svuotamento del carrello

*c.setMaxAge(604800); // Il cookie scade dopo una settimana!*

## Il Tuo Carrello

Il tuo carrello e' vuoto!

**Consorzio di Serre "Natura Viva"**

Colucci Carmine Antonio (961-87)  
D'Apolito Filomeno (961-81)  
Ferrara Francesco Saverio (961-79)

## Il Tuo Carrello

Nel tuo carrello ci sono i seguenti articoli:

Nome: Arancio(quantita'=1)

Il cliente può decidere di acquistare i prodotti del carrello in qualsiasi momento, ma l'acquisto è vincolato dalla registrazione o dal login al sito.

Dopo aver deciso di acquistare, il cliente dovrà decidere da quale serra acquistare i prodotti scelti.

## Conferma dell'ordine

Seleziona la serra a te piu' vicina:

Se la serra scelta non permette di soddisfare pienamente la



richiesta del cliente, soddisferà in parte l'ordine e la nostra applicazione provvederà a calcolare la prossima serra più vicina a quella scelta per continuare con l'ordine.

Se l'ordine va a buon fine viene effettuato l'acquisto altrimenti un opportuno messaggio di errore segnalerà quale prodotto non è stato possibile reperire (la quantità totale del prodotto nel consorzio è inferiore a quella richiesta dal cliente).

### **3. Registrazione al sito**

Per potersi registrare il cliente dovrà inserire le seguenti informazioni (tutte obbligatorie):

*Codice Fiscale, Nome, Cognome, Indirizzo, Città, Provincia, CAP, Password.*

All'evento *submit* del pulsante "Conferma", presente nel form di registrazione al sito, viene richiamata la stessa servlet *NewCliente* passando tutti i parametri precedentemente immessi tramite POST. Questi parametri vengono recuperati dal POST mediante il metodo *getParameter* dell'interfaccia *HttpServletRequest*.

Nel caso in cui vi sia l'impossibilità di ricavare i parametri passati dal post sarà stampato un messaggio di errore.

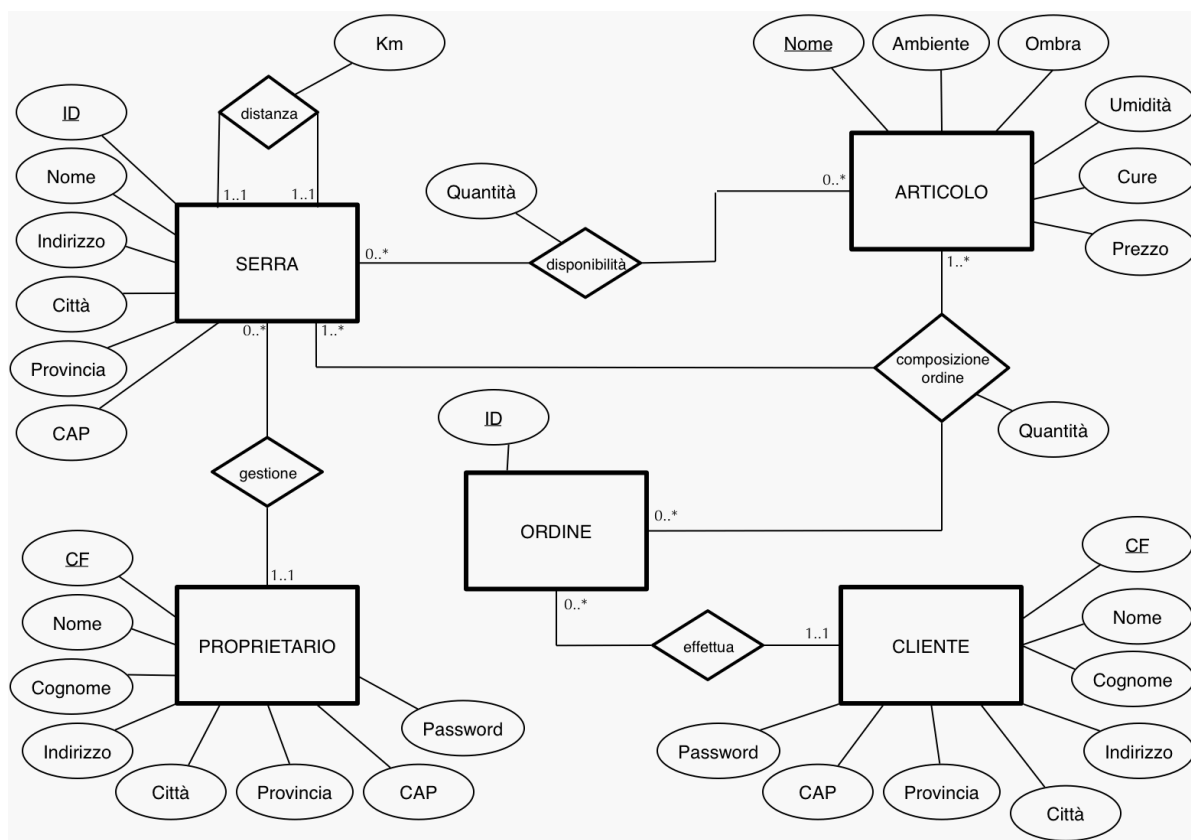
Dopo il recupero, un controllo sulla lunghezza dei parametri ne permetterà di verificarne la validità.

Per motivi di sicurezza, un ulteriore controllo viene fatto sul carattere " ' " che può provocare problemi di SQL INJECTION. In pratica viene sostituito il carattere " ` " con i due caratteri " ` ` ".

Il successivo step è quello di registrare il cliente al sito tramite una INSERT SQL.

La chiave primaria per ogni Cliente è il Codice Fiscale perciò vi è la possibilità di ricevere un errore qualora tentassimo di duplicarla.

## Allegato: Diagramma E/R



### Traduzione in tabelle:

articolo(Nome, Ambiente, Ombra, Umidità, Cura, Prezzo)

cliente(CF, password, Nome, Cognome, Indirizzo, Città, Provincia, CAP)

proprietario(CF, password, Nome, Cognome, Indirizzo, Città, Provincia, CAP)

ordine(ID, ~~CF\_Cliente~~)

composizione\_ordine(Quantità, ~~Nome\_Articolo~~, ID\_Ordine, ID\_Serra)

serra(ID, Nome, Indirizzo, Città, Provincia, CAP, ~~CF\_Proprietario~~)

disponibilità(Quantità, ~~Nome\_Articolo~~, ID\_Serra)

distanza(Km, ~~ID\_P~~, ~~ID\_A~~)

Notazioni: CHIAVE PRIMARIA

~~Chiave Esterna~~