

Progetto

Laboratorio di basi di dati:

Gruppo interfacce

Autore: Fabiana Chericoni

Email: f.chericoni1@studenti.unipi.it

Laboratorio di basi di dati, Anno Accademico: 2022/2023

II Semestre

May 12, 2023

1 Introduzione

L'obiettivo del progetto è la realizzazione di un sistema informatico che consiste in un sito web per la gestione di crociere da parte della compagnia *ISBD Crociere*. Il sistema permetterà di automatizzare il processo di raccolta delle prenotazioni di servizi offerti dalle crociere convenzionate. In particolare, sarà possibile svolgere una serie di operazioni che riguardano la manipolazione e l'interrogazione di tabelle presenti nello schema logico/concettuale, rappresentante lo scenario in questione, in relazione al ruolo e ai relativi permessi che l'utente possiede.

Ciascuna procedura PL/SQL riceve in input una serie di parametri e genera il codice HTML della pagina Web, quindi si chiede di capire il ruolo svolto dai parametri al fine di ottenere l'effetto desiderato. Si consiglia di leggere le specifiche (*GUI.pks*) e l'implementazione (*GUI.pkb*) all'interno del pacchetto GUI.

Per una maggiore comprensione il report si divide in tre parti:

1. CONVENZIONI generali da seguire in fase di progettazione delle procedure utilizzate per la realizzazione delle operazioni.
2. STANDARD da adottare per la progettazione dell'interfaccia delle pagine web create.
3. GUIDA AL PACCHETTO GUI, che consiste nella spiegazione delle procedure definite nel pacchetto **GUI**, create in accordo alle convenzioni e agli standard mostrati nei precedenti paragrafi.

1.1 Diagrammi di stato

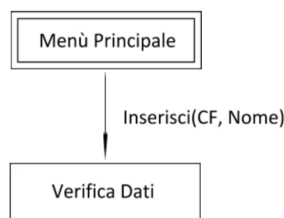
I diagrammi di stato sono un tipo di diagramma utilizzato per descrivere il comportamento di un sistema. In particolare, essi si presentano come un grafo orientato dove i

nodi indicano gli **stati** e gli archi le **transazioni** e mostrano come il sistema risponde agli eventi esterni e come cambia il suo stato interno in risposta a tali eventi. Gli stati, rappresentano le diverse condizioni interne del sistema e descrivono come esso si comporta in una determinata situazione. Le transizioni collegano gli stati e rappresentano il passaggio da uno stato all'altro in risposta agli eventi esterni, si indicano con una freccia etichettata, dallo stato di origine allo stato di destinazione: Lo stato iniziale di ogni

Rappresentazione degli stati nel diagramma



Rappresentazione delle transizioni nel diagramma



operazione è rappresentato dalla pagina *HomepageLogin* alla quale è possibile accedere inserendo le proprie credenziali, tramite il bottone di login dalla Homepage principale. Inoltre è possibile passare da un'operazione all'altra attraverso collegamenti dati da bottoni posizionati all'interno di un particolare stato. Dunque In ogni automa è presente lo stato "Stato Iniziale". Questo stato NON fa parte dell'automa dell'operazione in sé e pertanto non deve essere realizzato nel momento dell'implementazione dell'operazione. Tale stato rappresenta solo lo stato in cui si trova il sistema al momento del verificarsi dell'evento che dà inizio all'operazione, ovvero in *HomepageLogin*.

2 CONVENZIONI

Di seguito sono riportati le convenzioni da adottare nella fase di progettazione:

- Ogni pagina deve essere aperta con **APRIPAGINASTANDARD**, che permette l'apertura dei tag `<html >` e `<body >` di una pagina HTML e inserisce la Navbar. Quest'ultima può presentarsi in due modi, una in caso di utente loggato e l'altra per sessione da ospite.
- Ogni pagina deve essere chiusa da **CHIUDIPAGINASTANDARD** inserendo come parametro il nome del gruppo di lavoro da cui è stata progettata, in questo modo viene mostrato uno stile diverso (definito in *costanti.sql*) a seconda del gruppo di appartenenza in modo così da facilitare il riconoscimento.
- Ogni form deve essere aperto con **APRIFORM** e chiuso con **CHIUDIFORM**, inoltre deve contenere le procedure **APRIFIELD** e **CHIUDIFIELD** per aprire e chiudere un campo, **INPUT_FORM** per inserire un input, **APRISELECT-INPUT** e **CHIUDISELECTINPUT** per aprire e chiudere un input di tipo select le cui opzioni sono inserite tramite **OPTION_SELECT**. Infine si utilizza un bottone di submit e uno di reset con la procedura **BTNSUBMIT** e **BTNRESET** rispettivamente.
- Il passaggio dei parametri può essere anche parziale inserendo il nome del parametro a cui ci riferiamo, il simbolo `=>` e il valore da passare.
- Tutte le operazioni devono essere sempre raggiungibili, direttamente o indirettamente, partendo dal menu principale ed eventualmente, anche seguendo link (bottoni) all'interno delle pagine.
- Le date devono essere visualizzate nel formato *YYYY – MM – DD*.
- Nel caso in cui una ricerca non produca risultati, visualizzare la tabella con l'intestazione ed una riga in cui si annuncia la non presenza di dati o in alternativa aprire una pagina standard e inserire un messaggio nella pagina della form in cui si informa l'utente della mancanza di risultati (tramite l'attributo *LOAD = onload* nel body). Altrimenti è possibile aprire una pagina standard in cui si inserisce un messaggio di mancanza dei dati tramite la procedura **APRICARD**.
- Ogni interfaccia relativa alle operazioni, deve sempre poter avere un riferimento all'utente loggato (tramite il pacchetto *Authenticate* è sempre possibile chiamare la procedura *RECUPERA_SESSIONE* e *RECUPERA_UTENTE_DA_SESSIONE*).
- Nella creazione dei campi di un form, impostate il parametro *type* in modo che sia significativo e non usare sempre TEXT. In questo modo staticamente viene già

effettuato un primo controllo dei dati, ma comunque non è sufficiente a verificare tutti i vincoli imposti nel documento di analisi. Quindi prima di proseguire e sempre opportuno inserire uno stato di controllo dati.

- Non si ha la propagazione tra le varie procedure dei parametri: *IDSESSIONE*, *RUOLO* e *PERMESSI* per evitare che procedure intermedie eseguino il passaggio dei parametri senza utilizzarle al loro interno, ma quando necessarie vengono ricavate grazie alle procedure presenti nel pacchetto *Authenticate*. Dopo una qualsiasi operazione di inserimento, modifica o cancellazione (*INSERT, UPDATE, DELETE*), l'utente deve visualizzare una pagina contenente un messaggio che indica il successo o il fallimento dell'operazione.
- Una qualsiasi visualizzazione di dati ottenuti in seguito ad un'interrogazione della base di dati può essere mostrata tramite le procedure relative alla creazione di una tabella che utilizza ***APRITABELLA***(\dots) e ***CHIUDITABELLA*** con all'interno *GUI.APRIRIGATABELLA*;
GUI.CELLATABELLAHEADER(\dots);
 \dots
GUI.CELLATABELLAHEADER(\dots);
GUI.CHIUDIRIGATABELLA;
 \dots
GUI.APRIRIGATABELLA;
GUI.CELLATABELLA(\dots);
 \dots
GUI.CHIUDIRIGATABELLA;
 per l'intestazione delle colonne e l'inserimento di ciascuna riga che la costituisce.
- I messaggi di errore e di successo sono inseriti all'interno di una card che cambia il colore del proprio *header* a seconda dell'esito dell'operazione (Figura 1). Tali messaggi devono essere chiamati con la procedura ***GUI.ESITOPERAZIONE(ESITO, (MESSAGGIO)*** che riceve in input l'*ESITO* (può assumere solo due valori: *Successo* e *KO*, rispettivamente per messaggi di successo e messaggi di errore) e il *MESSAGGIO* che si vuole visualizzare all'interno della card creata.

ESEMPI: I messaggi di feedback devono essere mostrati all'utente per mostrare l'esito dell'operazione effettuata. Tale messaggio dovrà essere visualizzato in una nuova pagina creata utilizzando le procedure adeguate di creazione, dove infine potranno essere aggiunti dei bottoni che colleghino a procedure correlate oppure che permetta di tornare indietro.

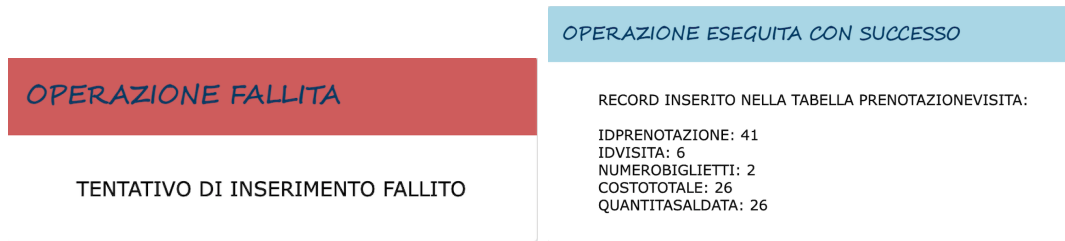


Figure 1: Messaggio in caso di fallimento (sx) e di successo (dx) di un'operazione.

3 STANDARD

3.1 Standard per l'inserimento

In Figura 2 si riporta lo standard da utilizzare per gli inserimenti.

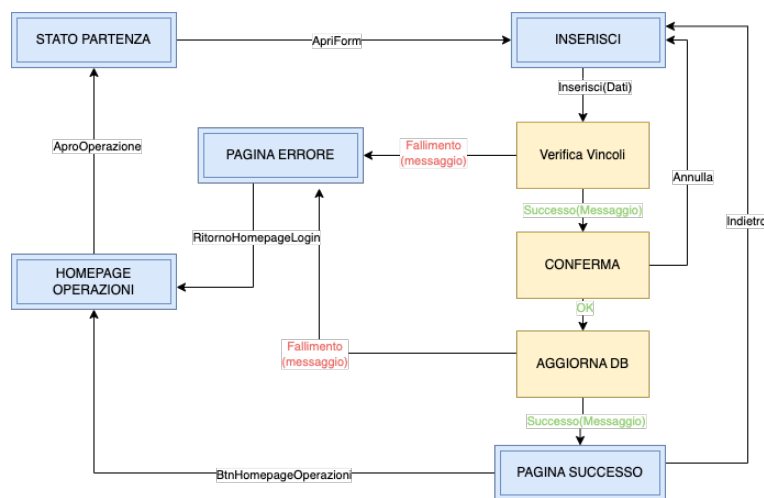


Figure 2: Diagramma a stati per inserimento

Quando l'utente seleziona un'operazione di inserimento dovrà essere presentata una pagina standard con un form dove poter inserire i dati, tramite input di tipo testo, select, checkbox, radio button e area di testo. Per passare un ulteriore parametro che non può essere visibile nel form si utilizza la procedura *INPUT_FORM*(...*TYPE* \Rightarrow *hidden*...). Una volta che l'utente avrà riempito gli input obbligatori (contenenti l'attributo *required*), tramite il bottone di submit è possibile essere reindirizzati ad un'altra procedura (specificata dal parametro *AZIONE* nella procedura *APRIFORM*) che provvederà ad effettuare l'inserimento nella basi di dati (direttamente o in seguito al reindirizzamento ad altre pagine intermedie necessarie per il recupero di altri parametri di input). Prima viene effettuata una *validazione* per i campi di testo del form, i quali possono assumere i seguenti tipi: *text*, *number*, *email*, *date*, *datetime-local*, *time*, *tel* a seconda del parametro

tipo inserito nella procedura *INPUT_FORM*. In base al risultato dell'inserimento dovrà essere aperta una nuova pagina di feedback la quale chiamerà la procedura *ESITOPERAZIONE*. Tale pagina mostrerà l'esito dell'operazione e da essa è possibile tornare alla homepage delle operazioni tramite il bottone presente nella navbar. Oppure, tramite un bottone all'interno della card, è possibile passare ad una nuova procedura collegata ad essa. Ad esempio l'operazione di inserimento dei clienti da parte dell'utente che ha effettuato la prenotazione. In caso di fallimento l'utente può riprovare a reimmettere i dati da inserire, premendo il pulsante "Indietro" oppure può decidere di tornare al Menù Principale mediante l'apposito bottone contenuto nella navbar della pagina .

3.2 Standard per la cancellazione

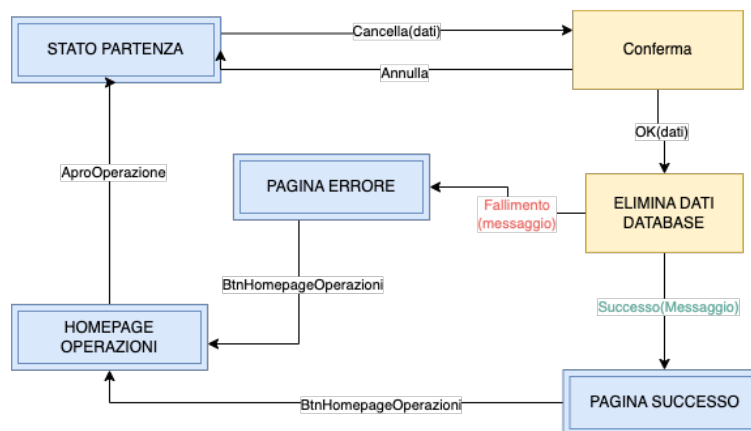


Figure 3: Diagramma a stati per la cancellazione

In Figura 3 è mostrato il diagramma degli stati minimali per l'operazione di cancellazione. Lo stato di partenza che avvia la cancellazione è sempre la pagina di homepage in cui sono presenti le operazioni, seguita però da una qualsiasi pagina di ricerca dati in cui, dopo aver visualizzato i risultati è possibile eliminare l'elemento selezionato se ne abbiamo il privilegio. Prima di poter effettuare tale operazione l'utente deve visualizzare un messaggio in cui si chiede la conferma. Se l'azione viene annullata si viene indirizzati nella pagina in cui si visualizzavano i dati precedentemente all'eliminazione. Mentre se si conferma si transita nello stato di verifica vincoli in cui, di nascosto all'utente, si verifica se le dipendenze permettono di eliminare i dati. Tramite la procedura *ESITOPERAZIONE* si viene reindirizzati in una pagina in cui viene notificato l'esito e dalla quale è possibile tornare alla pagina di *Homepage* delle operazioni tramite il bottone nella navbar o ad una qualsiasi altra pagina tramite il bottone inseribile nella card del messaggio tramite i parametri *BTN*, *LINKTO* e *TEXTBTN* nella procedura *ESITOPERAZIONE*.

3.3 Standard per la ricerca

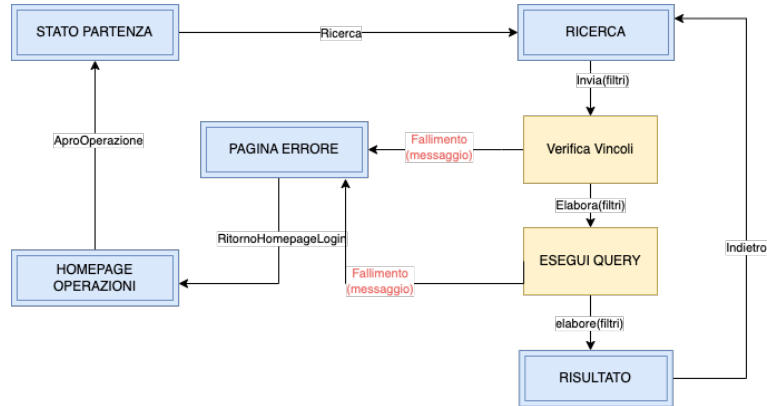


Figure 4: Diagramma a stati per la ricerca

Per visualizzare i dati correttamente quest'ultimi devono essere inseriti in una tabella nella quale, oltre a visualizzare le informazioni, si ha la possibilità di inserire un bottone che indirizza ad una nuova procedura passando i dati visualizzati all'interno della riga corrispondente. Nel testo del bottone viene specificata l'operazione a cui siamo reindirizzati (*CANCELLA* per procedere all'eliminazione del record corrispondente). Ad esempio nella procedura *VISUALIZZAVISITE*, tramite il parametro *ELIMINA* possiamo visualizzare il bottone *PRENOTA* o *CANCELLA* per avviare procedure diverse, ma che necessitano dei dati presenti all'interno della riga della tabella visualizzata. Quando il risultato della query corrisponde ad un unico record (ad esempio risultati di operazioni statistiche che estraggono il massimo/minimo) è possibile visualizzare i dati all'interno di una card con la procedura *APRICARD*, altrimenti sempre con una tabella avente un'unica riga oltre all'intestazione specifica. All'interno dello stato ricerca l'utente deve visualizzare un form nel quale deve impostare determinati filtri. Scelti i criteri che i dati da visualizzare devono rispettare l'utente transita nello stato successivo cliccando sul pulsante di invio: *VERIFICA VINCOLI*. Se i filtri sono consistenti si procede passando allo stato successivo, altrimenti viene ricaricata la pagina precedente passandogli un messaggio di errore. Esegui query esegue la query costruita con i filtri passati dallo stato precedente e tramite la procedura *ESITOPERAZIONE* viene visualizzato un messaggio di errore in caso di fallimento, altrimenti viene visualizzato il risultato. Da quest'ultimo stato è possibile tornare indietro per poter effettuare una nuova ricerca, mentre dallo stato in cui si notifica il fallimento dell'operazione è possibile tornare alla homepage delle operazioni tramite il bottone presente nella navbar.

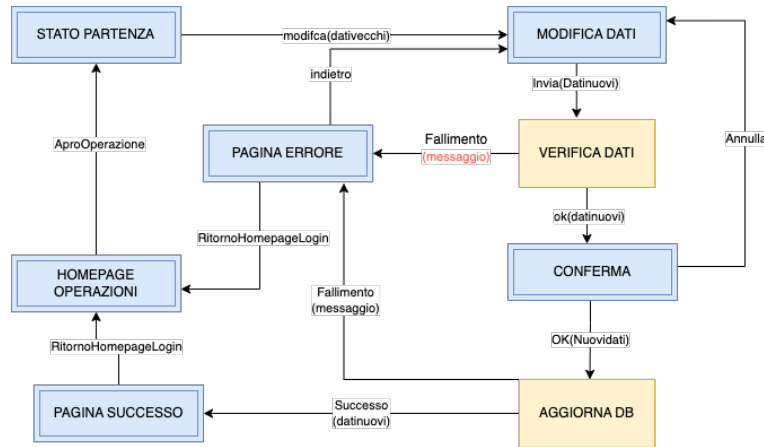


Figure 5: Diagramma a stati per la modifica.

3.4 Standard per la modifica

In figura 5 lo stato di partenza, può essere il risultato di una ricerca, dove è possibile inserire una colonna all'interno della tabella di visualizzazione contenente bottoni di modifica. Questi permettono di avviare l'evento modifica che consente di transitare nell'interfaccia grafica rappresentata dallo stato esterno MODIFICA. A questo punto l'utente può inserire il nome della colonna da modificare, il nuovo valore e la condizione che il record deve soddisfare affinché avvenga la sua modifica. Lo stato VERIFICA VINCOLI agisce di nascosto all'utilizzatore ed ha il compito di verificare che i nuovi dati immessi non violino i vincoli stabiliti nel documento di analisi. Da questo stato, se uno o più dati violano qualche vincolo, si transita nello stato di fallimento con la notifica corrispondente, in caso contrario si chiede l'ultima conferma per effettuare l'operazione e in caso affermativo si transita nello stato AGGIORNA DB, che analogamente ai casi precedenti può generare un messaggio di successo o di fallimento, tramite la procedura ESITOPERAZIONE i cui parametri sono diversi a seconda che sia chiamata all'interno di una EXCEPTION della procedura corrispondente. Infine è possibile il reindirizzamento alla homepage delle operazioni tramite il bottone nella navbar.

4 Guida al pacchetto GUI

Di seguito sono mostrate le procedure presenti nel pacchetto GUI:

4.1 APRIPAGINASTANDARD e CHIUDIPAGINASTANDARD

La procedura APRIPAGINASTANDARD permette l'apertura del tag `<html>` e `<body>` e del container della pagina. Imposta la codifica dei caratteri della pagina, un metadato per permettere il responsive design e in fine, carica i font esterni, lo stile e gli script definiti ed utilizzati all'interno delle pagine. Inoltre inserisce la navbar tramite la procedura APRINAV, presentata in seguito.

\param TITOLO: il titolo da assegnare alla pagina.
\param LOAD: parametro passato alla procedura apricontainer.
\param MESSAGGIO: messaggio che viene visualizzato al caricamento della pagina (se LOAD=ONLOAD) o alla chiusura (se LOAD=onUnLoad).

```
PROCEDURE APRIPAGINASTANDARD(TITOLO VARCHAR2 DEFAULT  
    'ISBD Crociere',LOAD VARCHAR2 DEFAULT '', MESSAGGIO  
    VARCHAR2 DEFAULT '');
```

CHIUDIPAGINASTANDARD rappresenta la corrispondente procedura di chiusura e richiama la chiusura del container prima di aggiungere il footer della pagina, il cui colore è scelto in base al gruppo che ha progettato la pagina.

\param FOOTER: con 'verde', 'arancione', 'blu' e 'interfaccia' si cambia il colore al footer della pagina che mostra il gruppo che ha progettato la pagina stessa.

```
PROCEDURE CHIUDIPAGINASTANDARD(footer varchar2 default  
    'Interfacce');
```

4.2 APRICONTAINER e CHIUDICONTAINER

La procedura APRICONTAINER permette il posizionamento del footer in fondo alla pagina. Viene aggiunto un container con altezza e larghezza 100% che racchiude tutto il corpo della pagina.

\param LOAD: permette di inserire un alert al caricamento della pagina (onload) oppure in uscita (onUnLoad).

```
PROCEDURE APRICONTAINER(LOAD VARCHAR2 DEFAULT '');
```

CHIUDICONTAINER rappresenta la corrispondente procedura di chiusura la quale chiude il div del container contenente il corpo della pagina. Non viene chiamata direttamente ma si apre e chiude all'interno della procedura APRIPAGINASTANDARD e CHIUDIPAGINASTANDARD rispettivamente. Dopo tale procedura si inserisce il footer della pagina con la procedura APRICHIUDIFOOTER.

```
PROCEDURE CHIUDICONTAINER (footer varchar2 default  
    'Interfacce');
```

4.3 APRICHIUDIFOOTER

APRICHIUDIFOOTER permette di inserire il footer dopo il container della pagina.

\param CREATORE: permette di cambiare il colore e il nome presente all'interno del footer in modo da riconoscere più facilmente chi ha progettato la pagina.

```
PROCEDURE APRICHIUDIFOOTER(creatore varchar2 default  
    'Interfacce')
```

4.4 APRIHOMEPAGE

APRIHOMEPAGE è la prima procedura che costituisce una intera pagina dunque è costituita dalla apertura e dalla chiusura di una pagina standard e al suo interno inserisce il form per la ricerca delle crociere disponibili.

```
PROCEDURE APRIHOMEPAGE;
```

4.5 APRIPAGINALOGIN

APRIPAINALOGIN costituisce pagina intera con apertura e chiusura di una pagina standard e al suo interno inserisce il form per l'inserimento delle credenziali tramite la procedura FORMLOGIN.

PROCEDURE APRIPAGINALOGIN;

4.6 HOMEPAGELOGIN

HOMEPAGELOGIN costituisce la pagina di homepage dopo aver effettuato il login, rappresenta lo stato iniziale della operazioni di ciascun gruppo. Contiene l'apertura e la chiusura di una pagina standard e al suo interno inserisce un form per la visualizzazione delle operazioni di ciascun gruppo (FORMGRUPPI).

PROCEDURE HOMEPAGELOGIN;

4.7 APRINAV

APRINAV è una procedura che permette l'inserimento di una navbar all'interno di ogni pagina che viene visualizzata dall'utente. Al suo interno si richiama la funzione RECUPERA_SESSIONE dal pacchetto AUTHENTICATE per il recupero della sessione, questo permette di distinguere due tipologie diverse a seconda che l'utente sia identificato o meno

PROCEDURE APRINAV;

4.8 APRIFORM e CHIUDIFORM

APRIFORM e CHIUDIFORM: permettono di aprire e chiudere una div con un form all'interno del quale è possibile specificarne le caratteristiche. CHIUDIFORM chiude il form. A ciascun APRIFORM deve corrispondere la chiamata alla procedura CHIUDIFORM

\param CLASSE: indica la classe di appartenenza del form.
\param METODO: get o post indicano le due metodologie possibili con cui il form invia i dati di input.
\param AZIONE: indica il link a cui si ha il reindirizzamento dopo aver premuto il bottone di invio dei dati (submit).
\param JS: se diverso da NULL permette di specificare un'azione da effettuare nel momento in cui viene premuto il bottone submit del form (come la conferma di invio dei dati tramite la funzione .confirm di javascript).

```
PROCEDURE APRIFORM(CLASSE VARCHAR2 DEFAULT 'FORM CARD',
    METODO VARCHAR2 DEFAULT '', AZIONE VARCHAR2 DEFAULT '',
    JS VARCHAR2 DEFAULT NULL);
```

```
PROCEDURE CHIUDIFORM;
```

4.9 CARDHEADER

La procedura CARDHEADER permette di definire il titolo del form e di scegliere uno dei due loghi disponibili, uno indica il simbolo del login inserito nei form di login e l'altro indica il logo della compagnia.

\param TITOLO: indica il titolo che viene visualizzato nel form

\param LOGO: assume valore 0 o 1 permettendo di inserire due loghi diversi a seconda dello scopo del form.

```
PROCEDURE CARDHEADER( TITOLO VARCHAR2 DEFAULT 'ISBD
    CROCIERE', LOGO INT DEFAULT 0);
```

4.10 APRIFIELD e CHIUDIFIELD

APRIFIELD e CHIUDIFIELD: permettono di aprire e chiudere una div contenente un campo da inserire all'interno del form con la label corrispondente.

\param CLASSE: indica la classe del div.

\param TESTO: Indica il testo da inserire nella label.

\param ALLINEAMENTO: indica l'allineamento del campo.

```
PROCEDURE APRIFIELD (CLASSE VARCHAR2 DEFAULT
    'FIELD', TESTO VARCHAR2 DEFAULT '', ALLINEAMENTO
    VARCHAR2 DEFAULT '');
```

```
PROCEDURE CHIUDIFIELD;
```

4.11 APRISELECTINPUT, OPTION_SELECT, CHIUDISELECT-INPUT

APRISELECTINPUT permette di aprire una select all'interno del form. Ogni selezione da parte dell'utente la impostiamo obbligatoria (inserendo l'attributo required al suo interno). CHIUDISELECT permette la chiusura della select, mentre OPTION_SELECT permette di inserire le opzioni da visualizzare al suo interno

.
\param NOME: indica il nome della select.
\param ID: Indica il codice univoco per l'identificazione della select.
\param ONCHANGE: indica un evento attivato quando viene selezionato un altro elemento da una lista o quando si modifica un campo di testo, per cui oltre all'attivazione, occorre anche operare un'azione.
\param VALORE: indica il valore del campo selezionato.
\param TESTO: indica il testo che l'utente visualizza all'interno della select.

```
PROCEDURE APRISELECTINPUT(NOME VARCHAR2 DEFAULT 'NAME',  
    ID VARCHAR2 DEFAULT '', ONCHANGE VARCHAR2 DEFAULT '');
```

```
PROCEDURE CHIUDISELECT;
```

```
PROCEDURE OPTION_SELECT(VALORE VARCHAR2 DEFAULT '', TESTO  
    VARCHAR2 DEFAULT '');
```

4.12 APRICARD, CORPOCARD e CHIUDICARD

APRICARD, CORPOCARD e CHIUDICARD scritte in sequenza permettono di aprire/chudere una card e inserire all'interno il testo che si vuole visualizzare. Vengono utilizzati per la visualizzazione di messaggi importanti come conclusione di un operazione.

\param INTESAZIONE: definisce l'intestazione della card.
\param CLASSEHEADER: definisce la classe dell'header, cambiando il colore.
\param FOOTER: Indica il testo da inserire nel footer della card.

```

PROCEDURE APRICARD(INTESTAZIONE VARCHAR2 DEFAULT
    '',CLASSEHEADER varchar2 DEFAULT 'card1-header');

PROCEDURE CHIUDICARD(FOOTER VARCHAR2 DEFAULT '');

PROCEDURE ESITOPERAZIONE(ESITO varchar2, MESSAGGIO
    varchar2 DEFAULT '',BTN VARCHAR2 DEFAULT 'NO',LINKTO
    VARCHAR2 DEFAULT '',TESTOBTN VARCHAR2 DEFAULT '');

```

4.13 PROCEDURE PER CREARE BOTTONI

La procedura BTN crea una bottone standard (quelli presenti nella nav bar) se *conferma* = 0, altrimenti crea un bottone standard con la richiesta di conferma dopo il click del bottone stesso.

La procedura BTNG crea un bottone standard con uno stile diverso dal precedente (quelli presenti nel form frm nella pagina di login).

La procedura BTNRESET crea un bottone di tipo reset nel form.

La procedura BTNSUBMIT crea un bottone di tipo submit nel form responsabile per l'invio dei dati in input. Con il parametro alert è possibile inserire un messaggio di alert che si mostra in seguito al click.

\param ID: Specifica un id univoco per un elemento

\param NOME: Specifica un nome per il pulsante.

\param TXT: Specifica il testo da visualizzare sopra il pulsante

\param VALORE: Indica il valore iniziale del pulsante.

\param DISABLED: Indica che il pulsante deve essere disabilitato

\param ALERT: indica la presenza o meno di un messaggio di alert in seguito al click del bottone.

\param CONFERMA: indica la presenza o meno di un messaggio di conferma al passaggio della pagina a cui il bottone indirizza

\param ONSUBMIT: specifica l'azione che il bottone dovr eseguire dopo il click.

Esempio: CONFERMA => 1, ONSUBMIT => 'onclick="if(confirm('vuoi eliminare la riga?')){location.href=''|Costanti.macchina2_|_|Costanti.radice_||...)}';

```

PROCEDURE BTN(CLASSE VARCHAR2 DEFAULT 'BTN',id varchar2
    default '',nome varchar2 default 'btn',VALORE VARCHAR2

```

```

        DEFAULT '', TXT varchar2 default 'button standard',
        ONSUBMIT VARCHAR2 DEFAULT '',CONFERMA INT DEFAULT 0);

PROCEDURE BTNG(id varchar2 default '', nome varchar2
        default 'btn', TXT varchar2 default 'button standard');

PROCEDURE BTNRESET(id varchar2 default '',nome varchar2
        default 'reset', TXT varchar2 default 'Cancella dati
        immessi');

PROCEDURE BTNSUBMIT(nome varchar2 default 'submit', id
        varchar2 default '',TXT varchar2 default 'Invia',ALERT
        INTEGER DEFAULT 0,MESSAGGIO VARCHAR2 DEFAULT 'CIAO');

```

4.14 ESITOPERAZIONE

La procedura ESITOPERAZIONE permette di visualizzare una card di colore blu se l'operazione è stata eseguita con successo, con un eventuale messaggio e un bottone al suo interno che porta ad una pagina voluta. Altrimenti mostra una card di colore rosso, con un messaggio di errore al suo interno in caso di fallimento dell'operazione

\param ESITO: indica l'esito dell'operazione 'SUCCESSO' o 'KO'.
 \param MESSAGGIO: indica il messaggio che verr visualizzato all'interno.
 \param BTN: 'SI' indica la presenza di un bottone, 'NO' in caso di assenza del bottone nella card del successo.
 \param LINKTO: nel caso in cui BTN='SI' permette di inserire il link per il bottone.
 \param TESTOBTN: nel caso in cui BTN='SI' permette di inserire il testo da visualizzare sul bottone.

```

PROCEDURE ESITOPERAZIONE(ESITO varchar2, MESSAGGIO
        varchar2 DEFAULT '',BTN VARCHAR2 DEFAULT 'NO',LINKTO
        VARCHAR2 DEFAULT '',TESTOBTN VARCHAR2 DEFAULT '');

```

4.15 PROCEDURE PER CREARE TABELLE

Infine sono riportate le procedure che devono essere chiamate per la realizzazione di una tabella: APRITABELLA permette di inserire il titolo, attributi particolari che la tabella dovrà avere e la tableId che permette di cambiare colore a seconda del gruppo di realizzazione. La chiusura viene effettuata con la procedura CHIUDITABELLA. Per la creazione dell'intestazione è presente la procedura CELLATABELLAHEADER. Per la creazione di una nuova riga si inserisce in sequenza APRIRIGATABELLA, CELLATABELLA (tante volte quante le colonne) e infine CHIUDIRIGATABELLA. All'interno di una cella è possibile inserire un bottone, ad esempio per l'operazione di eliminazione o di modifica della riga in riferimento, tramite la procedura CELLATABELLABTN, che grazie al parametro ONSUBMIT, se conferma=1, è possibile richiedere la conferma per effettuare l'operazione.

```
PROCEDURE APRITABELLA(titolo VARCHAR2 default '',
    dimTitolo int default 1, attributi varchar2 default
    '', tableId VARCHAR2 default 't00');

PROCEDURE CHIUDITABELLA;

PROCEDURE APRIRIGATABELLA(attributi VARCHAR2 default '');

PROCEDURE CHIUDIRIGATABELLA;

PROCEDURE CELLATABELLA(val varchar2 default '',
    attributi varchar2 default '');

PROCEDURE CELLATABELLAHEADER(val varchar2 default '',
    scopo varchar default 'col', attributi varchar2
    default '');

PROCEDURE CELLATABELLABTN(TXT VARCHAR2 DEFAULT '',
    ATTRIBUTI VARCHAR2 DEFAULT '', LINKTO VARCHAR2 DEFAULT
    '', ONSUBMIT VARCHAR2 DEFAULT '', CONFERMA INTEGER
    DEFAULT 0);

PROCEDURE CELLATABELLALINK(TXT VARCHAR2 DEFAULT '',
    ATTRIBUTI VARCHAR2 DEFAULT '', LINKTO VARCHAR2 DEFAULT
    '');
```


All'interno del pacchetto GUI sono definite altre procedure il cui significato è dato dai tag html che racchiudono e che data la loro semplicità non sono riportanti nella seguente guida al pacchetto. Anche le procedure più complesse che rappresentano le pagine progettate dal gruppo interfaccia non sono qui riportate in quanto si mostrano come esempi di utilizzo delle procedure sopra citate, dunque corrispondono a chiamate in sequenza delle procedure mostrate con parametri specifici, adeguati al contesto della progettazione della pagina.