



Laboratorio di Basi di Dati

Anno Accademico 2020-2021

Guida al pacchetto modGUI

Versione 2.1

Gruppo Interfaccia:

Davide Chen

Docente:

Giovanna Rosone

26 novembre 2020

Sommario

| | | |
|----------|--------------------|----------|
| 1 | Procedure | 4 |
| 1.1 | ApriPagina | 4 |
| 1.2 | ApriBody | 4 |
| 1.3 | ApriMenuNav | 5 |
| 1.4 | ApriMenuLat | 5 |
| 1.5 | ChiudiMenuNav | 6 |
| 1.6 | ChiudiMenuLat | 6 |
| 1.7 | CreaMenuPrincipale | 7 |
| 1.8 | CreaMenuBackOffice | 7 |
| 1.9 | Intestazione | 7 |
| 1.10 | ChiudiPagina | 8 |
| 1.11 | RitCarrello | 8 |
| 1.12 | Reindirizza | 8 |
| 1.13 | ApriForm | 9 |
| 1.14 | ApriFormHidden | 9 |
| 1.15 | ChiudiFormHidden | 10 |
| 1.16 | ChiudiForm | 10 |
| 1.17 | ApriDiv | 10 |
| 1.18 | ChiudiDiv | 10 |
| 1.19 | Paragrafo | 11 |
| 1.20 | Collegamento | 11 |
| 1.21 | CollegamentoNav | 12 |
| 1.22 | AreaDiTesto | 13 |
| 1.23 | Bottone | 13 |
| 1.24 | BottoneForm | 14 |
| 1.25 | BottoneInfo | 15 |
| 1.26 | BottoneEdit | 15 |
| 1.27 | BottoneConfirm | 15 |
| 1.28 | BottoneDelete | 15 |
| 1.29 | BottoneCart | 16 |
| 1.30 | EtichettaDiTesto | 16 |

| | | |
|------|-----------------------------------|----|
| 1.31 | CasellaDiTesto | 16 |
| 1.32 | CampoPassword | 17 |
| 1.33 | ApriBlocco..... | 18 |
| 1.34 | ChiudiBlocco | 18 |
| 1.35 | ApriBloccoSelect | 19 |
| 1.36 | ChiudiBloccoSelect | 19 |
| 1.37 | StringaDiTesto | 19 |
| 1.38 | ApriTabella..... | 20 |
| 1.39 | ChiudiTabella | 20 |
| 1.40 | ApriRigaTabella..... | 20 |
| 1.41 | ChiudiRigaTabella | 20 |
| 1.42 | ApriElementoTabella | 20 |
| 1.43 | ChiudiElementoTabella..... | 21 |
| 1.44 | ElementoTabella..... | 21 |
| 1.45 | IntestazioneTabella..... | 21 |
| 1.46 | CheckBox | 22 |
| 1.47 | RadioButton..... | 22 |
| 1.48 | ApriSelect..... | 23 |
| 1.49 | ApriInlineSelect..... | 23 |
| 1.50 | ChiudiInlineSelectConBottone..... | 24 |
| 1.51 | AggiungiOpzioneSelect | 24 |
| 1.52 | ChiudiSelect | 24 |
| 1.53 | SelettoreData..... | 25 |
| 1.54 | PassaParametro..... | 25 |
| 1.55 | PaginaFeedback..... | 26 |
| 1.56 | PaginaOspiti..... | 27 |
| 1.57 | PaginaPrincipaleBackoffice..... | 27 |

1 Procedure

1.1 ApriPagina

Questa procedura permette di aprire i tag e body di una pagina HTML e inserire automaticamente lo stile predefinito.

|*param* titolo: il titolo da assegnare alla pagina

|*param* idSessione: la sessione con cui vogliamo aprire la pagina

```
procedure ApriPagina(  
    titolo varchar2 default 'Senza titolo',  
    idSessione int default 0  
);
```

1.2 ApriBody

Questa procedura permette di aprire un tag body e inserire i bottoni di login/logout a seconda del parametro idSessione.

N.B. Questa procedura è chiamata all'interno di *ApriPagina*, quindi non si presenterà la necessità di invocarla direttamente.

|*param* idSessione: la sessione con cui vogliamo aprire la pagina

```
procedure ApriBody(  
    idSessione int default 0  
);
```

1.3 ApriMenuNav

Procedura attraverso il quale viene creato il menu di navigazione classica orizzontale, personalizzabile con links e sub-menus, attraverso le chiamate alla procedura *CollegamentoNav* (da non confondere con la procedura *Collegamento*). Al termine ricordarsi di chiamare la procedura *ChiudiMenuNav*.

|*param* idSessione: la sessione con cui vogliamo aprire la pagina

```
procedure ApriMenuNav(  
    idSessione int  
);
```

1.4 ApriMenuLat

Procedura attraverso il quale viene creato il menu di navigazione laterale (che compare quando la dimensione del proprio browser è sotto un certo limite, nascondendo nel contempo il menu di navigazione orizzontale), personalizzabile con links e sub-menus, attraverso le chiamate alla procedura *Collegamento* (da non confondere con la procedura *CollegamentoNav*). Al termine ricordarsi di chiamare la procedura *ChiudiMenuLat*.

|*param* idSessione: la sessione con cui vogliamo aprire la pagina

|*param* ident: l'id **OBBLIGATORIO** di ogni menu e sub-menu laterale utile per l'apertura e la chiusura dei medesimi sulla pagina web

ATTENZIONE: dopo aver scelto l'id per il Menu Laterale PRINCIPALE, ricordarsi di copiarlo anche nella Costante mySidenav nel file Costanti.pks poiché viene usato anche in un'altra parte del codice.

N.B. Al fine di evitare problemi di omonimi, si consiglia di iniziare gli id con il nomeUtente o nomeRuolo

|*param* subMenu: settarlo a true quando quando il menu laterale che si sta andando a creare è un sub-menu di un altro menu laterale

|*param* parentName: se subMenu è settato a true, passare il nome del link chiamante (cioè il contenuto del parametro testo della procedura che chiama tale menu di navigazione)

```
procedure ApriMenuLat(  
    idSessione int,  
    ident varchar2 default '',  
    subMenu boolean default false,  
    parentName varchar2 default ''  
);
```

1.5 ChiudiMenuNav

Permette di chiudere il blocco Menu Navigazione Orizzontale dopo aver chiamato la procedura *ApriMenuNav*.

|*param* idSessione: la sessione con cui vogliamo aprire la pagina

```
procedure ChiudiMenuNav(  
    idSessione int  
);
```

1.6 ChiudiMenuLat

Permette di chiudere il blocco Menu Navigazione Laterale dopo aver chiamato la procedura *ApriMenuLat*.

```
procedure ChiudiMenuLat;
```

1.7 CreaMenuPrincipale

/***** DEPRECATO *****/

Procedura attraverso la quale viene creato il menu principale. Si raccomanda di invocare sempre questa procedura immediatamente dopo l'invocazione ad *ApriPagina*.

|*param* idSessione: la sessione con cui vogliamo aprire la pagina

```
procedure CreaMenuPrincipale(  
    idSessione int  
);
```

1.8 CreaMenuBackOffice

/***** DEPRECATO *****/

Procedura attraverso la quale viene creato il menu dedicato al personale di backoffice (Admin). Si raccomanda di invocare sempre questa procedura immediatamente dopo l'invocazione ad *ApriPagina* (solo se si sta costruendo una pagina dell'area di Backoffice, s'intende).

|*param* idSessione: Identifica la sessione corrente

1.9 Intestazione

In questa procedura si crea un tag di intestazione.

N.B. Dopo l'apertura di una pagina e l'aggiunta dell'apposito menù (se richiesto), si dovrebbe sempre aggiungere un'intestazione.

|*param* tipo: il tipo dell'intestazione (1 per h1 e così via)

|*param* testo: il testo da inserire nel tag

|*param* ident: id del tag

|*param* classe: classe del tag

```
procedure Intestazione(  
    tipo int,  
    testo varchar2,  
    ident varchar2 default '',  
    classe varchar2 default ''  
);
```

1.10 ChiudiPagina

Con questa procedura si chiudono i tag html e body di una pagina HTML.

```
procedure ChiudiPagina;
```

1.11 RitCarrello

Produce il tag html br (a capo).

```
procedure RitCarrello;
```

1.12 Reindirizza

Questa procedura permette di reindirizzare l'utente verso l'indirizzo (URL) specificato come parametro.

|*param* indirizzo: l'indirizzo di destinazione

```
procedure Reindirizza(  
    indirizzo varchar2  
);
```


1.13 ApriForm

La procedura *ApriForm* apre un tag form. L'invocazione di questa procedura dovrebbe sempre precedere l'aggiunta di controlli quali bottoni, caselle di testo, checkbox, etc.

|*param* azione: procedura che dev'essere legata alla form

|*param* nome: nome della form da passare al validatore

|*param* idSessione: sessione da passare come parametro hidden alla procedura "azione"

```
procedure ApriForm(  
    azione varchar2,  
    nome varchar2,  
    idSessione int default -1  
);
```

1.14 ApriFormHidden

Apre una form i cui campi di input sono SOLO di tipo HIDDEN.

Da usare solo per inserire bottoni usati per passare i parametri come idSessione, etc.

N.B. Dopo l'apertura di una pagina e l'aggiunta dell'apposito menù (se richiesto), si dovrebbe sempre aggiungere un'intestazione.

|*param* azione: procedura che dev'essere legata alla form

|*param* idSessione: sessione da passare come parametro hidden alla procedura "azione"

```
procedure ApriFormHidden (  
    azione varchar2,  
    idSessione int default -1,  
);
```

1.15 ChiudiFormHidden

Procedura con la quale si chiude una FormHidden. Dev'essere sempre chiamata dopo ApriFormHidden.

```
procedure ChiuiFormHidden;
```

1.16 ChiudiForm

Procedura con la quale si chiude un tag form. Dev'essere sempre chiamata dopo *ApriForm*.

```
procedure ChiudiForm;
```

1.17 ApriDiv

Chiamando questa procedura si apre un tag div di cui è possibile specificare anche id e classe.

|*param* ident: l'id del div

|*param* classe: la classe del div

```
procedure ApriDiv (  
    ident varchar2 default '',  
    classe varchar2 default '',  
);
```

1.18 ChiudiDiv

Questa procedura può essere chiamata solo dopo aver invocato *ApriDiv*. Essa permette di chiudere un tag div.

procedure ChiudiDiv;

1.19 Paragrafo

La procedura crea un tag p e lo riempie con il testo specificato dal parametro testo (la chiusura del tag è inclusa). Anche in questo caso è possibile specificare le informazioni di id e classe

|*param* testo: il testo che viene inserito all'interno del tag

|*param* ident: id del paragrafo

|*param* classe: classe del paragrafo

```
procedure Paragrafo(  
    testo varchar2,  
    ident varchar2 default '',  
    classe varchar2 default ''  
);
```

1.20 Collegamento

Questa procedura inserisce un collegamento (tag a) nella pagina HTML (include la chiusura del tag). Si devono specificare il testo del link e l'indirizzo a cui fa riferimento (href). SOLO nel caso di utilizzo nel Menu di Navigazione Laterale settare opportunamente anche il parametro hasSub.

N.B. Da non confondere con la procedura *CollegamentoNav*

|*param* testo: il testo del link mostrato sullo schermo

|*param* indirizzo: l'indirizzo di destinazione del collegamento, oppure quando il parametro hasSub è settato a true, passare l'id del sub-menu laterale da chiamare (parametro ident della procedura ApriMenuLat a cui si fa riferimento)

|*param* hasSub: solo nel caso di utilizzo nel Menu di Navigazione Laterale settarlo a true se il parametro indirizzo deve portare ad un sub-menu, invece che ad una procedura o altro tipo di destinazione

```
procedure Collegamento(  

```

```

        testo varchar2,
        indirizzo varchar2,
        hasSub boolean default false)
);

```

1.21 CollegamentoNav

Questa procedura inserisce un collegamento (tag a) nella pagina HTML (include la chiusura del tag). È usato esclusivamente nel Menu di Navigazione Orizzontale. Si devono specificare il testo del link e l'indirizzo a cui fa riferimento (href).

N.B. Da non confondere con la procedura *Collegamento*

ATTENZIONE: Se il parametro hasSub è settato a true, ricordarsi di chiamare la procedura ChiudiCollegamentoSubNav dopo aver chiamato tutte le sub-menus.

|*param* testo: il testo del link mostrato sullo schermo
 |*param* indirizzo: l'indirizzo di destinazione del collegamento,
 |*param* hasSub: settarlo a true è il link a un sub-menu
 |*param* tipoSub: se hasSub è settato a true, specificare il tipo del sub-menu

- a. 'primo' se di tipo "primario" (sub-menu)
- b. 'secondo' se di tipo secondario (sub-sub-menu)

```

procedure Collegamento(
    testo varchar2,
    indirizzo varchar2,
    hasSub boolean default false
    tipoSub varchar2 default ''
);

```

1.22 AreaDiTesto

La chiamata a questa procedura ha come effetto la creazione di una casella di testo espandibile in cui l'utente può scrivere del testo.

|*param* nome: nome da assegnare all'area di testo

|*param* etichetta: etichetta associata all'area di testo

|*param* valore: valore iniziale contenuto nell'area di testo (Stringa al suo interno)

|*param* center: se settato a true, la casella è posizionata al centro, altrimenti a sinistra

```
procedure AreaDiTesto(  
    nome varchar2,  
    etichetta varchar2,  
    valore varchar2 default '',  
    center boolean default true  
);
```

1.23 Bottone

La procedura consente di creare un bottone. Si consiglia fortemente di chiamare questa procedura dopo aver chiamato *ApriFormHidden*.

|*param* testo: testo da visualizzare nel bottone

|*param* nome: nome del parametro da passare alla procedura che verrà invocata in risposta al click su questo bottone

|*param* valore: valore del parametro da passare alla procedura che verrà invocata in risposta al click su questo bottone

|*param* ident: id del bottone

|*param* classe: classe del bottone

```
procedure Bottone(  
    testo varchar2 default '',
```

```

        nome varchar2 default '',
        valore varchar2 default '',
        ident varchar2 default '',
        classe varchar2 default ''
    );

```

1.24 BottoneForm

La procedura consente di creare un bottone. Si consiglia fortemente di chiamare questa procedura dopo aver chiamato *ApriForm*, altrimenti il bottone non avrà alcun effetto.

|*param* nome: nome del parametro da passare alla procedura che verrà invocata in risposta al click su questo bottone

|*param* valore: valore del parametro da passare alla procedura che verrà invocata in risposta al click su questo bottone

|*param* testo: testo da visualizzare sul bottone

|*param* valida: richiedo che il validatore controlli i contenuti della form

|*param* nomeForm: nome della form in cui è stato inserito il bottone
(obbligatorio se valida = true)

```

procedure BottoneForm(
    nome varchar2 default null,
    valore varchar2 default null,
    testo varchar2 default 'Submit',
    valida boolean default true,
    nomeForm varchar2 default ''
);

```

1.25 BottoneInfo

La procedura consente di creare il bottone Info. Si consiglia fortemente di chiamare questa procedura dopo aver chiamato *ApriFormHidden*, altrimenti il bottone non avrà alcun effetto.

procedure BottoneInfo;

1.26 BottoneEdit

La procedura consente di creare il bottone Edit. Si consiglia fortemente di chiamare questa procedura dopo aver chiamato *ApriFormHidden*, altrimenti il bottone non avrà alcun effetto.

procedure BottoneEdit;

1.27 BottoneConfirm

La procedura consente di creare il bottone Conferma. Si consiglia fortemente di chiamare questa procedura dopo aver chiamato *ApriFormHidden*, altrimenti il bottone non avrà alcun effetto.

procedure BottoneConfirm;

1.28 BottoneDelete

La procedura consente di creare il bottone Delete. Si consiglia fortemente di chiamare questa procedura dopo aver chiamato *ApriFormHidden*, altrimenti il bottone non avrà alcun effetto.

procedure BottoneDelete;

1.29 BottoneCart

La procedura consente di creare il bottone Carrello. Si consiglia fortemente di chiamare questa procedura dopo aver chiamato *ApriFormHidden*, altrimenti il bottone non avrà alcun effetto.

```
procedure BottoneCart;
```

1.30 EtichettaDiTesto

Questa procedura consente di inserire del testo preceduto da un'etichetta che descrive come dev'essere interpretato. Esempio di output della procedura: [etichetta: testo]

|*param* etichetta: l'etichetta da apporre al testo

|*param* testo: il testo da visualizzare

```
procedure EtichettaDiTesto(  
    etichetta varchar2,  
    testo varchar2,  
);
```

1.31 CasellaDiTesto

La procedura consente di creare una casella di testo semplice a cui è associata un'etichetta.

|*param* nome: nome della casella di testo

|*param* etichetta: etichetta visualizzata accanto alla casella

|*param* suggerimento: suggerisce all'utente la natura dell'informazione che dev'essere inserita nella casella

|*param* testo: testo da inserire all'interno della casella

|*param* lunghezzaMax: indica la lunghezza massima della stringa che l'utente può inserire

|*param* tipo: indica il tipo di stringa che l'utente deve inserire per superare la validazione del campo. I valori possibili sono:

- a. text: contiene solo lettere. I caratteri speciali sono esclusi.
- b. number: contiene solo numeri
- c. contiene un indirizzo e-mail
- d. alfa contiene stringhe alfanumeriche, esclusi caratteri speciali

|*param* require: se settato a true, indica che il campo è obbligatorio

|*param* center: se settato a true, la casella è posizionata al centro, altrimenti a sinistra

```
procedure CasellaDiTesto(
    nome varchar2,
    etichetta varchar2,
    suggerimento varchar2 default '',
    testo varchar2 default '',
    lunghezzaMax varchar2 default '',
    tipo varchar2 default '',
    require boolean default false,
    center boolean default true
);
```

1.32 CampoPassword

La procedura consente di crea una casella di testo adatta a contenere password.

|*param* nome: nome del campo password

|*param* etichetta: etichetta visualizzata accanto alla casella

|*param* suggerimento: suggerisce all'utente la natura dell'informazione

|*param* valore: valore iniziale del campo

|*param* require: se settato a true, il campo è obbligatorio

|*param* center: se settato a true, il campo è posizionata al centro

```

procedure CampoPassword(
    nome varchar2,
    etichetta varchar2 default 'Password',
    suggerimento varchar2 default '',
    valore varchar2 default '',
    require boolean default false,
    center boolean default true
);

```

1.33 ApriBlocco

Apri un blocco all'interno della form, che può contenere SOLO CheckBox, RadioButton e/o Bottoni. I campi specificati in precedenza sono aggiunti allineati in riga. Per inserirli in una riga successiva usare RitCarello. È OBBLIGATORIO chiamare questa procedura per inserire correttamente i campi sopra specificati

|*param* etichetta: etichetta visualizzata accanto al blocco

|*param* center: se settato a true, il blocco è posizionato al centro

```

Procedure ApriBlocco(
    etichetta varchar2 default '',
    center boolean default true
);

```

1.34 ChiudiBlocco

Chiude il blocco aperto con ApriBlocco.

```

procedure ChiudiBlocco;

```

1.35 ApriBloccoSelect

Apri un blocco all'interno della form, che deve contenere SOLO campi di tipo Select. Obbligatorio usare questa procedura per inserire campi di tipo Select nella form. Le Select sono inserite nel blocco sulla stessa riga. Per andare ad una riga successiva NON usare RitCarrello, ma Chiudere il Blocco Select corrente ed aprirne un altro con etichetta alla stringa vuota.

|*param* etichetta: etichetta visualizza accanto al blocco

|*param* center: se settato a true, il blocco è posizionato al centro

```
procedure ApriBloccoSelect(  
    etichetta varchar2 default '',  
    center boolean default true  
);
```

1.36 ChiudiBloccoSelect

Chiude il blocco aperto con ApriBloccoSelect.

```
procedure ChiudiBloccoSelect;
```

1.37 StringaDiTesto

Stampa sullo schermo una Stringa.

|*param* testo: il testo da stampare

```
procedure StringaDiTesto(  
    testo varchar2;  
);
```

1.38 ApriTabella

Permette di aprire una tabella HTML

procedure ApriTabella;

1.39 ChiudiTabella

Permette di chiudere una tabella HTML

procedure ChiudiTabella;

1.40 ApriRigaTabella

Inserisce una riga nella tabella. Può essere chiamata solo dopo *ApriTabella*.

procedure ApriRigaTabella;

1.41 ChiudiRigaTabella

Permette di chiudere una riga di una tabella HTML.

procedure ChiudiRigaTabella;

1.42 ApriElementoTabella

Delimita l'inizio di un elemento all'interno di una riga di una tabella HTML. La procedura *ApriRigaTabella* deve procedere questa procedura

procedure ApriElementoTabella;

1.43 ChiudiElementoTabella

Delimita la fine di un elemento all'interno di una riga di una tabella HTML.

```
procedure ChiudiElementoTabella;
```

1.44 ElementoTabella

Crea un elemento testuale all'interno di una riga di una tabella HTML.

|*param* testo: il testo da associare all'elemento

```
procedure ElementoTabella(  
    testo: varchar2  
);
```

1.45 IntestazioneTabella

Crea un elemento intestazione (tag th) all'interno di una riga di una tabella HTML. Questa procedura si occupa di aprire e chiudere il tag.

|*param* testo: il testo dell'intestazione

```
procedure IntestazioneTabella(  
    testo varchar2  
);
```

1.46 CheckBox

Crea una checkbox. Questa procedura si occupa di aprire e chiudere il tag. Si consiglia di chiamare questa procedura all'interno di una form, altrimenti la checkbox non avrà effetto.

|*param* etichetta: etichetta da affiancare alla checkbox

|*param* nome: nome del parametro da passare alla procedura che verrà invocata in caso di un evento submit

|*param* valore: valore del parametro da passare alla procedura che verrà invocata in caso di un evento submit

|*param* selezionato: indica se la checkbox dev'essere già selezionata o meno

```
procedure CheckBox(  
    etichetta varchar2,  
    nome varchar2,  
    valore varchar2,  
    selezionato boolean default false  
);
```

1.47 RadioButton

Crea un radiobutton. Questa procedura si occupa di aprire e chiudere il tag.

Si consiglia di chiamare questa procedura all'interno di una form, altrimenti il radiobutton non avrà effetto.

|*param* etichetta: etichetta da affiancare al radiobutton

|*param* nome: nome del radiobutton

|*param* valore: valore del parametro passato dal radio button

|*param* selezionato: indica se il radiobutton dev'essere già selezionato o meno

```
procedure RadioButton(  

```

```

        etichetta varchar2,
        nome varchar2,
        valore varchar2,
        selezionato boolean default false
    );

```

1.48 ApriSelect

Questa procedura consente di inserire un tipico menù a tendina. Tale menù dovrebbe essere sempre incluso all'interno di una Form.

|*param* nome: il nome del parametro da passare ad un'eventuale azione che verrà invocata mediante la pressione di un bottone. Il suo valore corrisponderà all'opzione selezionata dal menù

|*param* center: se settato a true, il campo è posizionato al centro, altrimenti a sinistra

```

procedure ApriSelect(
    nome varchar2,
    center boolean default true
);

```

1.49 ApriInlineSelect

Questa procedura consente di inserire un tipico menù a tendina all'interno di una tabella. Tale menù dovrebbe essere sempre incluso tra le procedure *ApriMenuHidden* e *ChiudiMenuHidden*.

|*param* nome: il nome del parametro da passare ad un'eventuale azione che verrà invocata mediante la pressione di un bottone. Il suo valore corrisponderà all'opzione selezionata dal menù

```

procedure ApriInlineSelect(
    nome varchar2,

```

);

1.50 ChiudiInlineSelectConBottone

Questa procedura è da usarsi solo ed obbligatoriamente dopo una chiamata ad *ApriInlineSelect*, permettendo così di concludere il menu a tendina con un bottone di conferma.

```
procedure ChiudiInlineSelectConBottone;
```

1.51 AggiungiOpzioneSelect

Questa procedura consente di aggiungere un'opzione ad un menù a tendina precedentemente creato con *ApriSelect*.

|*param* valore: il valore (inteso come quello che verrà passato ad un'eventuale azione) dell'opzione che si desidera includere tra quelle del menù a tendina

|*param* etichetta: il nome dell'opzione così come compare nel menù a tendina

|*param* selezionato: determina se l'opzione dev'essere quella selezionata per default:

```
procedure AggiungiOpzioneSelect(  
    valore varchar2,  
    etichetta varchar2,  
    selezionato boolean default false  
);
```

1.52 ChiudiSelect

Questa procedura è da usarsi solo ed obbligatoriamente dopo una chiamata ad *ApriSelect*.


```
procedure ChiudiSelect;
```

1.53 SelettoreData

Questa procedura permette di includere un controllo per la selezione di una data all'interno della pagina. Si richiede che tale controllo sia inserito all'interno di una Form.

|*param* idCampoGiorno: il nome del parametro – il cui valore sarà quello selezionato dal menù a tendina contenente i giorni – che verrà passato ad un'eventuale azione che verrà invocata

|*param* idCampoMese: il nome del parametro – il cui valore sarà quello selezionato dal menù a tendina contenente i mesi – che verrà passato ad un'eventuale azione che verrà invocata

|*param* idCampoAnno: il nome del parametro – il cui valore sarà quello selezionato dal menù a tendina contenente gli anni – che verrà passato ad un'eventuale azione che verrà invocata

|*param* dataDefault: la data predefinita visualizzata nel controllo

|*param* etichetta: etichetta del selettore

|*param* center: se settato a true è posizionato al centro

```
procedure SelettoreData(  
    idCampoGiorno varchar2 default 'giorno',  
    idCampoMese varchar2 default 'mese',  
    idCampoAnno varchar2 default 'anno',  
    dataDefault date,  
    etichetta varchar2,  
    center boolean default true  
);
```

1.54 PassaParametro

Questa procedura consente di passare un parametro in modalità *hidden* (cioè nascosto) ad una certa azione quando verrà invocata,

ossia quando l'utente premerà il bottone che la lancerà. Ciò potrà accadere soltanto se la chiamata di questa procedura avverrà tra un'invocazione ad *ApriForm* ed una *ChiudiForm*.

|*param* nome: il nome del parametro da passare

|*param* valore: il valore del parametro da passare

```
procedure PassaParametro(  
    nome varchar2,  
    valore varchar2 default null  
);
```

1.55 PaginaFeedback

Questa procedura permette di creare una pagina che informa l'utente dell'esito di un'operazione da lui richiesta.

|*param* tipoMessaggio: tipo del messaggio che si vuole mostrare.

Può assumere i valori

- a. 'SUCCESSO'
- b. 'ERRORE'

|*param* descrizioneMessaggio: messaggio di feedback da mostrare

|*param* indirizzoRitorno: il nome della procedura da invocare nel caso in cui l'utente desideri tornare all'ultima schermata di ritorno visualizzata o ad un'altra pagina come la 'Home'

|*param* idSessione: l'id che racchiude informazioni sulla sessione in corso. Se questo parametro non viene specificato, la procedura impostata tramite il parametro 'indirizzoRitorno' non riceverà il parametro 'idSessione'

|*param* parametriProcedura: una tabella contenente i parametri da passare alla procedura specificata in 'indirizzoRitorno' (escluso 'idSessione'). Si veda il file 'modGUI.pks' per ulteriori informazioni sul tipo 'modGUI.stringDict'

```
procedure PaginaFeedback(  
    tipoMessaggio varchar2,
```

```

        descrizioneMessaggio varchar2,
        indirizzoRitorno varchar2,
        idSessione int default -1,
        parametriProcedura modGUI.stringDict default
modGUI.emptyDict
    );

```

1.56 PaginaOspiti

Realizza la pagina da mostrare agli utenti che non hanno effettuato ancora il login o la registrazione. Si tratta del punto d'ingresso dell'applicazione, nonché della pagina di destinazione dopo un'operazione di logout.

```

procedure PaginaOspiti;

```

1.57 PaginaPrincipaleBackoffice

Realizza la pagina principale dell'area backoffice (Admin) da visualizzare in seguito ad un click sulla voce 'Area Backoffice' del menù principale.

|*param* idSessione: identifica la sessione corrente

```

procedure PaginaPrincipaleBackoffice(
    idSessione int
);

```