UNIVERSIDADE DO PORTO
FACULDADE DE ENGENHARIA
Departamento de Engenharia Electrotécnica e de Computadores

# Implementação de Facilidades de Geração de Relatórios que Complementem o SQL*RW do SGBD ORACLE

**RELATÓRIO DE ESTÁGIO PRODEP**

**Maria Goretti de Oliveira Valente Soares**

**Porto, Junho de 1993**

Universidade do Porto
Faculdade de Engenharia
Biblioteca
Nº 621.3(043.3)/LEEC 1992/50Am
01 10 09

**PARECER**

**Acção de Formação PRODEP**

A aluna Maria Goretti de Oliveira Valente Soares esteve a realizar um estágio com vista à implementação de facilidades de geração de relatórios para complementar o SQL*RW do SGBD ORACLE.

A aluna cumpriu satisfatoriamente o plano de trabalho proposto. Em particular, estudou as facilidades de formatação de relatórios normalmente fornecidas por produtos comerciais, e concebeu programas que permitem gerar e visualizar alguns tipos de relatórios mais complexos não tratados pelas ferramentas comerciais disponíveis, recorrendo à linguagem de programação C e à biblioteca "curses" de gestão de ecrã e janelas.

22 de Julho de 1993

João Carlos Pascoal Faria

Eng. João Carlos Pascoal Faria
Assistente da FEUP

## PARECER

## Acção de Formação PRODEP

A estagiária Maria Goretti de Oliveira Valente Soares, cumpriu o trabalho proposto com vista à implementação de facilidades de geração de relatórios para complementar o SQL*Report Writer, do SGBD ORACLE.

Do seu trabalho resultou uma ferramenta, que permite colmatar algumas das lacunas existentes no SQL*Report Writer e que ajuda a ultrapassar dificuldades existentes no processo de geração de relatórios.

Porto, 22 de Julho de 1993

José Manuel Costa Correia

Eng. José Manuel da Costa Correia

Investigador

# RELATÓRIO DE ESTÁGIO

# "Implementação de Facilidades de Geração de Relatórios que Complementem o SQL*RW do SGBD Oracle"

**Maria Goretti de Oliveira Valente Soares**

**FEUP / INESC, Junho de 1993**

## AGRADECIMENTOS:

PRODEP.
INESC.
Prof. Vladimiro Miranda.
Eng. José Correia.
Eng. João Carlos Pascoal Faria.

# Indice

1. ESTUDO DO SGBD ORACLE (nomeadamente o PRO*C e o SQL*RW) E A LIGUAGEM SQL.

Durante esta fase, centrei o meu estudo sobre alguns dos programas que constituem o sistema de gestão de base de dados ORACLE e a linguagem SQL.

O SQL é uma linguagem de interrogação de bases de dados relacionais que permite cumprir os seguintes objectivos indispensáveis num gestor de base de dados: definição, manipulação e segurança dos dados. O SQL é constituido por um conjunto de palavras chaves que facilitam a realização dos objetivos anteriores. Nesta fase, os conhecimentos adquiridos sobre o SQL foram testados com a ajuda do SQL*PLUS.

O SQL*PLUS é um programa dirigido por comandos e tem como funções: a criação de tabelas, a inserção e alteração de informação nas mesmas, a pesquisa e visualização da informação armazenada e a manutenção da base de dados.

O SQL*PLUS aceita dois tipos de comandos: comandos SQL que permitem arranjar a informação contida na base de dados e comandos SQL*PLUS que permitem formatar resultados, editar, guardar ou alterar comandos SQL.

O SQL*RW é uma ferramenta para desenvolver e executar relatórios, especialmente concebido para utilizadores que conhecem a linguagem SQL. Consiste num conjunto de utilitários que permitem a manuipulação de relatórios através de janelas e menus.

Ainda, durante este ponto, estudei o PRO*C, linguagem na qual estão desenvolvidos os algoritmos requeridos.

O PRO*C é uma ferramenta que foi concebida para converter programas em C que incluam instruções SQL, num programa em C que possa ter acesso e manipular informação numa base de dados ORACLE. No papel de pré-compilador, o PRO*C converte as instruções EXEC SQL, encontradas no ficheiro fonte em chamadas ao ORACLE. O ficheiro de saída do pré-compilador pode ser compilado, "linkado" e executado como qualquer outro programa em C. A utilização destes utilitários permitiu-me conhecer as suas limitações quanto `a geração de relatórios.

2. ESTUDAR OUTRO GERADORES DE RELATORIOS, PARA AJUDAR A CONCEBER FORMAS DE RESOLVER AS LIMITACOES DE SQL*RW.

Durante este ponto, adquiri algumas noções sobre o SAGA.

O SAGA é uma ferramenta criada no INESC para o desenvolvimento expedito e utilização final de aplicações interactivas sobre BD relacionais em ambiente UNIX, recorrendo a vistas, regras e janelas. O SAGA foi pensado para resolver principalmente os problemas que se levantam com aplicações interactivas, por serem as que consomem maior esforço de programação, enquanto que os problemas que se colocam nos processamentos não interactivos já estão bem tratados nas linguagens de programação ou interrogação.

As aplicações a construir com o SAGA são basicamente compostas por formul arios de ecrã, relatórios formatados, um sistema de menus e um sistema de ajuda.

O SAGA constitui uma ferramenta bastante poderosa para a geração de relatórios.

## 3. DEFINIR ESTRUTURAS DE DADOS PARA ESPECIFICAR O FORMATO DE RELATORIOS COMPLEXOS PARTINDO DE ELEMENTOS SIMPLES.

Nesta fase, investiguei o tipo de relatórios existentes e classifiquei-os segundo o seu grau de complexidade.

Um dos tipos de relatórios mais simples é um relatorio tabular em que o nome das colunas são preenchidos pelos nomes dos campos selecionados, podendo tambem ser dados pelo utilizador.

Num relatório podem tambem existir colunas que resultam de operações sobre campos existentes, como: SUM, AVG, COUNT, MIN, MAX. Estas operações podem ser aplicadas em qualquer ponto do relatório, por exemplo um subtotal, fazendo intervir queries que têm uma relação "parent/child", em que a informação retornada por cada uma das queries "child", fará parte de uma query "parent" e no layout final do relatório, formarão um único bloco de informação.

Existem, ainda, relatórios que permitem inserir texto (ex. uma carta, uma mensagem), junto com o relatório gerado.

As ferramentas existentes no ORACLE, não permitem gerar relatórios em forma de quadro, é esse o objetivo das funções que desenvolvi.

Para gerar um relatório, parto de um elmento simples constituído por um valor retornado por uma query e de uma janela que irá conter o resultado. Apartir daqui, o utilizador dirá qual a janela a ser criada, onde e qual o texto a ser inserido.

## 4. DESENVOLVER ALGORITMOS PARA GERAR OS RELATOROS ESPECIFICADOS CONFORME 3., DE UMA FORMA MODULAR E IMPLEMENTAR EM C E PRO*C.

Com o intuito de superar as lacunas existentes nestas ferramentas do ORACLE, para a formatação de relatórios, criei uma biblioteca de funções, tirando partido das informações recolhidas até esse momento.

Para isso, foi necessário um método que me permitisse retirar um valor da base de dados, condicionado a certos parâmetros. Depois do estudo de algumas técnicas avançadas de programação, decidi incluir no meu programa "statements" definidos dinamicamente (variam de execução para execução).

Este método involve 5 passos:

1. PREPARE <.....> FROM <.....>
2. DECLARE <cursor> FOR <.....>
3. OPEN <cursor>
4. FETCH <cursor> INTO :var
5. CLOSE ,cursor>

O "statement" em SQL é preparado, seguidamente é-lhe associado um cursor e realiza-se o seu "parse", desta forma são identificados os seus registos activos. Com o "fetch", os valores são copiados para as variáveis hospedeiras.

Os valores `a medida que são selecionados da BD, são copiados para uma estrutura, uma vez que serão necessários para cálculos posteriores ( em totais, ou outros valores que dependem dele explicitamente). Quando um conjunto de valores deixa de ser útil para os seguintes, é apagado, permitindo um acesso mais rápido, na medida em que a procura do valor na estrutura também é mais rápida ( já que intervêm menos parâmetros nessa procura). A parte de janelas é gerada com a ajuda do curses.

O avanço no desenvolvimento dos algoritmos foi feito através do teste dos mesmos no projecto PEDAP e observando exemplos concretos do tipo quadro existentes no mesmo projecto.

## BIBLIOGRAFIA


- THE CURSES AND THE TERMOINFO PACKAGE.

- SEBENTA DE SQL.

- PRO*C USER'S GUIDE.
  Donna Neville.
  Peter Claire and Ken Jacobs.
  Oracle Corporation.

- LIVRO DE C.
  Kernighan and Ritchie.

- MANUAIS DE SQL*RW.
  MANUAIS DE SQL.
  Oracle Corporation.

ANEXO

# Listagem do programa principal

```
#include <string.h>

#define NMAX 100
#define HORIZ 0
#define VERT 1
#define NONE 2
#define OUTRO 3
#define ARGMAX 30
static int i,j,actual_x,actual_y,ind_jan_act,total,ultima,nur=0,total_col,u1,n,n1;
WINDOW *jan[NMAX];
char no_reg[3];
int regiao1[18],lp,comp_campo,len;
struct guard_jan{
                int coord_x;
                int coord_y;
                int altura;
                int comp;
                int ref[5];
               }lista_jan[NMAX];
struct guard_jan_sup{
                int coord_x;
                int coord_y;
                int altura;
                int comp;
                int ref[5];
                  }jan_sup[NMAX];
struct totais{
        char fonte[ARGMAX];
        long total;
      }g_tot[NMAX];
struct temp{
          char fontes[ARGMAX];
          long val;
         }vals_comps[NMAX];
struct verttot{
              char fonte[ARGMAX];
              int ano;
              long total;
             }g_totcol[NMAX];
```

```
EXEC SQL BEGIN DECLARE SECTION;
   char select[132];
   VARCHAR valor[10];
   short  ivalor;
   int regiao[18];
   VARCHAR uid[20];
   VARCHAR pwd[20];
EXEC SQL END DECLARE SECTION;
EXEC SQL INCLUDE SQLCA;

main()
{   int l,*arr_refalt,*arr_refcomp,*arr1,*arr2;

    strcpy(uid.arr,"PEDAP");
    uid.len=strlen(uid.arr);
    strcpy(pwd.arr,"ORAPEDAP");
    pwd.len=strlen(pwd.arr);
    EXEC SQL CONNECT :uid IDENTIFIED BY :pwd;

    comp_campo=9;
    lp=6;
    total=1;
    init(10);
          cel_sest("sub_est","invest",1986,VERT);
          celula_1("sub_feoga","invest",1986,VERT);
          cel_subtot("sub_tot","invest",1986,VERT);
          celula_1("auto_finan","invest",1986,VERT);
          cel_invtot("inv_tot","invest",1986,HORIZ);
          cel_sest("sub_est","invest",1987,VERT);
          celula_1("sub_feoga","invest",1987,VERT);
          cel_subtot("sub_tot","invest",1987,VERT);
          celula_1("auto_finan","invest",1987,VERT);
          cel_invtot("inv_tot","invest",1987,HORIZ);
          cel_sest("sub_est","invest",1988,VERT);
          celula_1("sub_feoga","invest",1988,VERT);
          cel_subtot("sub_tot","invest",1988,VERT);
          celula_1("auto_finan","invest",1988,VERT);
          cel_invtot("inv_tot","invest",1988,HORIZ);
          cel_sest("sub_est","invest",1989,VERT);
          celula_1("sub_feoga","invest",1989,VERT);
```

```
cel_subtot("sub_tot","invest",1989,VERT);
celula_1("auto_finan","invest",1989,VERT);
cel_invtot("inv_tot","invest",1989,HORIZ);
cel_tot("sub_est",VERT);
cel_tot("sub_feoga",VERT);
cel_tot("sub_tot",VERT);
cel_tot("auto_finan",VERT);
cel_tot("inv_tot",NONE);
jan_mlat(0,12,"sub.est.");
jan_mlat(1,12,"sub.feoga.");
jan_mlat(2,12,"sub_tot.");
jan_mlat(3,12,"auto.finan.");
jan_mlat(4,12,"inv.tot.");
jan_nmlat(5,9,12,1,"B.R.T.");
jan_msup(0,3,"1986");
jan_msup(1,3,"1987");
jan_msup(2,3,"1988");
jan_msup(3,3,"1989");
jan_vmsup(5,8,3,"ANOS",0);
arr_refalt[0]=9;
arr_refalt[1]=5;
arr_refcomp[0]=4;
arr1[0]=5;
jan_valtsup(arr_refalt,2,arr_refcomp,1,"TOTAL",0);
jan_supaltlat(arr_refalt,2,arr1,1,"FONTES",0);
arr1[0]=11;
arr2[0]=10;
jan_vlatssup(arr1,1,arr2,1,"",0);
endwin();
}
```

```
celula_1(campo_tab,nome_tab,ref_ano,ind)
char campo_tab[ARGMAX],nome_tab[ARGMAX];
int ref_ano,ind;
{
    int comp;
    char str[5],cad[3];

    sprintf(cad,"%d",regiao[nur]);
    sprintf(str,"%d",ref_ano);
    strcpy(select,"SELECT ");
    strcat(select,campo_tab);
    strcat(select," FROM ");
    strcat(select,nome_tab);
    strcat(select," WHERE ANO=");
    strcat(select,str);
    strcat(select," AND ");
    strcat(select,"c_prog=");
    strcat(select,cad);
    strcat(select," AND ");
    strcat(select,"c_subprog=");
    strcat(select,no_reg);
    comp=strlen(select);
    if (comp>131)
        {
        endwin();
        printf("SELECT muito longo\n");
        exit(1);
        }
    EXEC SQL PREPARE S1 FROM :select;
    EXEC SQL DECLARE C1 CURSOR FOR S1;
    EXEC SQL OPEN C1;

    EXEC SQL WHENEVER NOT FOUND GOTO fim;
    {
    EXEC SQL FETCH C1 INTO :valor:ivalor;
    if (ivalor==-1){  if (ind==OUTRO)
                return;
            else{
                des_jan("\0",comp_campo,ind);
                return;
                }
        }
```

```
    else {
        len=valor.len;
        valor.arr[len]='\0';
        copia_temp(campo_tab,valor.arr);
        }
    }
fim:
  if (ind==OUTRO)
    return;
  if (total==1)
    soma_tot(campo_tab,valor.arr);
  if (total_col==1)
    soma_totcol(campo_tab,ref_ano,valor.arr);
  des_jan(valor.arr,comp_campo,ind);
}



des_jan(ovalor,comp,sind)
char *ovalor;
int sind,comp;
{
    jan[ind_jan_act]=newwin(3,comp,actual_x,actual_y);
    if(!strcmp(ovalor,"\0")){
                box(jan[ind_jan_act],0,0);
                wrefresh(jan[ind_jan_act]);
               }
    else{
        wmove(jan[ind_jan_act],1,comp-len-1);
        waddstr(jan[ind_jan_act],ovalor);
        box(jan[ind_jan_act],0,0);
        wrefresh(jan[ind_jan_act]);
       }
     if (actual_x==i)
        creat1(3,comp);
     if (actual_y==j)
        creat(3,comp);
```

```
switch(sind){
            case VERT: actual_x+=2;
                    if (actual_y==j)
                      ultima=actual_x+2;
                    break;
            case HORIZ: actual_y+=comp_campo-1;
                     actual_x=u1;
                     break;
            case NONE: u1=ultima;
                     actual_x=ultima;
                     actual_y=j;
                     break;
             default: break;
           }
   ind_jan_act++;
   if (ind_jan_act>NMAX){   endwin();
                    printf("Neste momento  so sao permitidas %d janelas\n",NMAX);
                    exit(1);
                 }
}

init(um)
int um;
{
  int l;

   i=1+um;
   j=25;
   actual_x=i;
   actual_y=j;
   ultima=u1=i;
   ind_jan_act=n=n1=0;
   initscr();
   for(l=0;l<=NMAX;l++)
     {
      g_tot[l].fonte[0]='\0';
      g_tot[l].total=vals_comps[l].val=0;
      vals_comps[l].fontes[0]='\0';
      g_totcol[l].fonte[0]='\0';
      g_tot[l].total=0;
      }
```

```
for(l=0;l<=17;l++)
   regiao[l]='\0';
 if (lp==0){
          endwin();
          puts("Deve ser dado o numero da regiao");
          exit(0);
         }
 if (comp_campo==0){
               endwin();
               puts("Deve ser dado o comprimento dos campos segundo a construcao das tabelas");
              exit(0);
              }
 sprintf(no_reg,"%d",lp);
 subprogramas(no_reg);
}


creat(alt,compri)
int alt,compri;
{

    lista_jan[n].coord_x=actual_x;
    lista_jan[n].coord_y=actual_y;
    lista_jan[n].altura=alt;
    lista_jan[n].comp=compri;
    n++;
}

creat1(alt,compri)
int alt,compri;
{
    jan_sup[n1].coord_x=actual_x;
    jan_sup[n1].coord_y=actual_y;
    jan_sup[n1].altura=alt;
    jan_sup[n1].comp=compri;
    n1++;
}
```

```
soma_tot(o_fonte,val1)
char *o_fonte;
char *val1;
{
  int l;

  if (!atol(val1))
    return;
  for (l=0;g_tot[l].fonte[0]!='\0';l++)
      {
        if (!strcmp(g_tot[l].fonte,o_fonte)){
                                    g_tot[l].total+=atol(val1);
                                    return;
                                   }
      }
   strcpy(g_tot[l].fonte,o_fonte);
   g_tot[l].total=atol(val1);
}

copia_temp(ctab,longo)
char *ctab,*longo;
{
    int l;

    for(l=0;vals_comps[l].fontes[0];l++);
    strcpy(vals_comps[l].fontes,ctab);
    vals_comps[l].val=atol(longo);
}

cel_subtot(ncam,ntab,nano,ind)
char ncam[ARGMAX],ntab[ARGMAX];
int nano,ind;
{
    int l;
    long t,num1,num2,mvalor1;
    char cad[NMAX];

    t=encontra("sub_feoga");
    if (t==EOF)
      celula_1("sub_feoga",ntab,nano,OUTRO);
```

```
celula_1("perc_subs",ntab,nano,OUTRO);
  num1=encontra("sub_feoga");
  num2=encontra("perc_subs");
  switch(num1){
          case EOF: if (ind==OUTRO)
                      return;
                    else{ des_jan("'\0'",comp_campo,ind);
                      return;
                     }
                    break;
          default: if (num2==EOF){
                      if(ind==OUTRO)
                        return;
                       else{  des_jan("'\0'",comp_campo,ind);
                           return;
                         }
                       }
                   else{
                      mvalor1=num1/num2;
                      sprintf(cad,"%ld",mvalor1);
                      copia_temp(ncam,cad);
                      }
                   break;
        }
  if (ind==OUTRO)
    return;
  if (total==1)
    soma_tot(ncam,cad);
  if (total_col==1)
    soma_totcol(ncam,nano,cad);
  if (ind==HORIZ){
           for(l=0;vals_comps[l].fontes[0]!='\0';l++)
            { vals_comps[l].fontes[0]='\0';
              vals_comps[l].val=0;
            }
          }
  len=strlen(cad);
  des_jan(cad,comp_campo,ind);
}
```

```
long encontra(campo)
char *campo;
{
   int l;

   for(l=0;vals_comps[l].fontes[0];l++)
    {
       if(!strcmp(vals_comps[l].fontes,campo))
            return(vals_comps[l].val);
    }
   return (EOF);
}


cel_sest(est1,est2,nano,ind)
char est1[ARGMAX],est2[ARGMAX];
int nano,ind;
{
   long t1,num1,num2,mvalor1;
   char cad[NMAX];
   int l;

   t1=encontra("sub_tot");
   if (t1==EOF)
      cel_subtot("sub_tot",est2,nano,OUTRO);
   num1=encontra("sub_tot");
   num2=encontra("sub_feoga");
   switch(num1){
          case EOF:if (num2==EOF){
                        if(ind==OUTRO)
                          return;
                        else{ des_jan("\0",comp_campo,ind);
                            return;
                            }
                        }
                else{
                    sprintf(cad,"%ld",num2);
                    copia_temp(est1,cad);
                    }
                break;
```

```
        default: if (num2==EOF){
                    sprintf(cad,"%ld",num1);
                    copia_temp(est1,cad);
                   }
                 else{  mvalor1=num1-num2;
                     sprintf(cad,"%ld",mvalor1);
                     copia_temp(est1,cad);
                    }
                 break;
          }
 if (ind==OUTRO)
   return;
 if (total==1)
     soma_tot(est1,cad);
 if (total_col==1)
     soma_totcol(est1,nano,cad);
 if (ind==HORIZ){
            for (l=0;vals_comps[l].fontes[0]!='\0';l++)
             {  vals_comps[l].fontes[0]='\0';
               vals_comps[l].val=0;
             }
            }
  len=strlen(cad);
  des_jan(cad,comp_campo,ind);
}


cel_invtot(inv1,inv2,nano,ind)
char inv1[ARGMAX],inv2[ARGMAX];
int nano,ind;
{
   long t1,num1,num2,mvalor1;
   char cad[NMAX];
   int l;

   t1=encontra("sub_tot");
   if (t1==EOF)
     cel_subtot("sub_tot",inv2,nano,OUTRO);
   celula_1("auto_finan",inv2,nano,OUTRO);
   num1=encontra("sub_tot");
   num2=encontra("auto_finan");
```

```
switch(num1){
        case EOF: if (num2==EOF){
                    if (ind==OUTRO)
                      return;
                    else{ des_jan("'\0'",comp_campo,ind);
                       return;
                       }
                    }
                  else{ sprintf(cad,"%ld",num2);
                     copia_temp(inv1,cad);
                     }
                  break;
         default: if (num2==EOF)
                  { sprintf(cad,"%ld",num1);
                   copia_temp(inv1,cad);
                  }
                 else {  mvalor1=num1+num2;
                      sprintf(cad,"%ld",mvalor1);
                      copia_temp(inv1,cad);
                     }
                 break;
        }
  if(ind==OUTRO)
    return;
  if (total==1)
    soma_tot(inv1,cad);
  if (total_col==1)
    soma_totcol(inv1,nano,cad);
  if (ind==HORIZ){
              for(l=0;vals_comps[l].fontes[0]!='\0';l++)
                { vals_comps[l].fontes[0]='\0';
                 vals_comps[l].val=0;
                }
             }
  len=strlen(cad);
  des_jan(cad,comp_campo,ind);
}
```

```
actrjan(pa,pb,alt,compri)
int pa,pb,alt,compri;
{
   lista_jan[n].coord_x=pa;
   lista_jan[n].coord_y=pb;
   lista_jan[n].altura=alt;
   lista_jan[n].comp=compri;
   n++;
}

cel_tot(c1,ind)
char c1[ARGMAX];
int ind;
{
   char cad[ARGMAX];
   int l;

    for(l=0;g_tot[l].fonte[0];l++)
      {
        if (!strcmp(g_tot[l].fonte,c1)){
                    sprintf(cad,"%ld",g_tot[l].total);
                    len=strlen(cad);
                    des_jan(cad,comp_campo+2,ind);
                    return;
                              }
      }
    des_jan("'\0'",comp_campo+2,ind);
 }


 subprogramas(no_reg)
 char no_reg[3];
 {
   int l;
   strcpy(select,"SELECT DISTINCT c_prog FROM invest WHERE c_subprog=");
   strcat(select,no_reg);

   EXEC SQL PREPARE S1 FROM :select;
   EXEC SQL DECLARE C2 CURSOR FOR S1;
   EXEC SQL OPEN C2;
   EXEC SQL WHENEVER NOT FOUND GOTO fim1;
      {
```

```
fim1:
  for (l=0;regiao[l]!='\0';l++)
      regiao1[l]=regiao[l];
  if (regiao1[0]=='\0')
              {
               printf("Nao existem programas para a regiao:%s\n",no_reg);
               endwin();
               exit(0);
               }
}

zeros()
{
  int l;

     for(l=0;l<=NMAX;l++){
                    g_tot[l].fonte[0]='\0';
                    vals_comps[l].fontes[0]='\0';
                    vals_comps[l].val=0;
                   }
}

soma_totcol(o_fonte,nano,val1)
char *val1,*o_fonte;
int nano;
{
  int l;

    for (l=0;g_totcol[l].fonte[0];l++)
     {
      if (!strcmp(g_totcol[l].fonte,o_fonte) && g_totcol[l].ano==nano)
        {     strcpy(g_totcol[l].fonte,o_fonte);
              g_totcol[l].ano=nano;
              g_totcol[l].total+=atol(val1);
              return;
         }
      }
      strcpy(g_totcol[l].fonte,o_fonte);
      g_totcol[l].ano=nano;
      g_totcol[l].total=atol(val1);
 }
```

```
cel_totcol(c1,c2,ind)
char c1[ARGMAX];
int c2,ind;
{
  char cad[ARGMAX];
  int l;

   for(l=0;g_totcol[l].fonte[0]!='\0';l++)
     if (!strcmp(g_totcol[l].fonte,c1) && g_totcol[l].ano==c2)
       {
        sprintf(cad,"%ld",g_totcol[l].total);
        len=strlen(cad);
        des_jan(cad,comp_campo,ind);
        return;
        }
   des_jan("'\0'",comp_campo,ind);
}

actrjan1(b,c,alt,compri)
int b,alt,compri;
{
  jan_sup[n1].coord_x=b;
  jan_sup[n1].coord_y=c;
  jan_sup[n1].altura=alt;
  jan_sup[n1].comp=compri;
  n1++;
}

jan_mlat(no_jan,compri,str)
int no_jan,compri;
char str[20];
{
    int a,b,c,d;

   d=strlen(str);
   a=lista_jan[no_jan].coord_x;
   b=lista_jan[no_jan].coord_y-compri+1;
   term(a,b);
   c=lista_jan[no_jan].altura;
```

```
 jan[ind_jan_act]=newwin(c,compri,a,b);
  box(jan[ind_jan_act],0,0);
  mvwaddstr(jan[ind_jan_act],c/2,compri/2-d/2,str);
  wrefresh(jan[ind_jan_act]);
  actrjan(a,b,c,compri);
  ind_jan_act++;
}

jan_nmlat(no_jani,no_janf,compri,hv,str)
int no_jani,no_janf,compri,hv;
char str[20];
{
   int a,b,c,d,d1,r,l=0,c1=0;

   a=strlen(str);
   r=no_janf-no_jani+1;
   c=lista_jan[no_jani].coord_x;
   d1=lista_jan[no_jani].coord_y-compri+1;
   term(c,d1);
   for(l=no_jani;l<=no_janf;l++)
     c1=c1+lista_jan[l].altura;
   l=0;
   b=c1-r+1;    /* altura da nova janela */
   jan[ind_jan_act]=newwin(b,compri,c,d1);
   box(jan[ind_jan_act],0,0);
   if (hv==0)
      mvwaddstr(jan[ind_jan_act],b/2,compri/2-a/2+1,str);
   else{
        do{  mvwaddch(jan[ind_jan_act],b/2-a/2+l+1,compri/2-1,str[l]);
           if (str[++l]=='.')
             mvwaddch(jan[ind_jan_act],b/2-a/2+l,compri/2,str[l]);
           else
             l--;
           l++;
         }while(str[l]);
       }
   wrefresh(jan[ind_jan_act]);
   actrjan(c,d1,b,compri);
   ind_jan_act++;
 }
```

```
jan_msup(no_jan,alt,str)
int no_jan,alt;
char str[20];
{
 int a,b,c,d;

  a=strlen(str);
  b=jan_sup[no_jan].coord_x-alt+1;    /* Novo x */
  c=jan_sup[no_jan].comp;
  d=jan_sup[no_jan].coord_y;
  term(b,d);
  jan[ind_jan_act]=newwin(alt,c,b,d);
  mvwaddstr(jan[ind_jan_act],alt/2,c/2-a/2+1,str);
  box(jan[ind_jan_act],0,0);
  wrefresh(jan[ind_jan_act]);
  actrjan1(b,d,alt,c);
  ind_jan_act++;
}

jan_vmsup(no_jani,no_janf,alt,str,no_part)
int no_jani,no_janf,alt,no_part;
char str[20];
{
  int a,l,c1=0,c2,r,c,d,b,l1=0,a1,l2;
  char *str1;

  a=strlen(str);
  for(l=no_jani;l<=no_janf;l++)
     c1=c1+jan_sup[l].comp;
  b=jan_sup[no_jani].coord_x-alt+1;
  d=jan_sup[no_jani].coord_y;
  term(b,d);
  r=no_janf-no_jani+1;
  c2=c1/r;
  c=jan_sup[no_janf].coord_y+c2-d;
  term(d,c);
  jan[ind_jan_act]=newwin(alt,c,b,d);
  box(jan[ind_jan_act],0,0);
  l=0;
  if(a>c){
```

```
  ciclo: do{
              str1[l1++]=str[l++];
              l2=l;
               }while(str[l2]!='\n');
             str1[--l1]='\0';
             l1=0;
             a1=strlen(str1);
             if (a1>c){
                    endwin();
                    puts("String de comprimento maior do que a janela'\n'");
                    exit(1);
                   }
             mvwaddstr(jan[ind_jan_act],alt/2-no_part/2,c/2-a1/2,str1);
             no_part-=2;
             if(str[l])
               goto ciclo;
          }
else
  mvwaddstr(jan[ind_jan_act],alt/2,c/2-a/2,str);
wrefresh(jan[ind_jan_act]);
actrjan1(b,d,alt,c);
ind_jan_act++;
}

term(a,b)
int a,b;
{
  if (a<=0 || b<=0) {
                endwin();
                printf("Coordenadas negativas nao sao validas\n");
                printf("x=%d  e  y=%d\n",a,b);
                exit(1);
               }
}

jan_maisup(ref_ini,jan_ref,prop,jan_ini,alt,str)
int jan_ref,jan_ini,alt;
char str[20];
float prop;
{
   int a,b,c,d,l,c1=0;
```

```
a=jan_sup[jan_ini].coord_x-alt+1;
 b=jan_sup[jan_ref].comp*prop;
 for(l=ref_ini;l<jan_ref;l++)
   c1+=jan_sup[l].comp;
 c1+=b;
 c=jan_sup[jan_ini].coord_y;
 term(a,c);
 d=strlen(str);
 jan[ind_jan_act]=newwin(alt,c1,a,c);
 box(jan[ind_jan_act],0,0);
 mvwaddstr(jan[ind_jan_act],alt/2,c1/2-d/2,str);
 wrefresh(jan[ind_jan_act]);
 actrjan1(a,c,alt,c1);
 ind_jan_act++;
}

jan_valtsup(arr_refalt,n1,arr_refcomp,n2,str,hv)
int *arr_refalt,*arr_refcomp;
char str[20];
int hv,n1,n2;
{
   int l,c,d,e,m1,m2,a=0,b=0;
  e=strlen(str);
  for(l=0;l<n1;l++)
    {m1=arr_refalt[l];
      a=a+jan_sup[m1].altura;
    }
  a=a-n1+1;       /*altura da janela*/
  for(l=0;l<n2;l++)
    { m2=arr_refcomp[l];
      b=b+jan_sup[m2].comp;
     }
  b=b-n2+1;
  m1=arr_refalt[0];
  m2=arr_refcomp[0];
  c=jan_sup[m1].coord_x;
  d=jan_sup[m2].coord_y;
  term(c,d);
  jan[ind_jan_act]=newwin(a,b,c,d);
  box(jan[ind_jan_act],0,0);
  l=0;
```

```
if(hv==0)
    mvwaddstr(jan[ind_jan_act],a/2,b/2-e/2,str);
 else{
     do{
        mvwaddch(jan[ind_jan_act],a/2-e/2+l,b/2-1,str[l]);
        if (str[++l]=='.')
           mvwaddch(jan[ind_jan_act],a/2-e/2+l,b/2,str[l]);
        else
           l--;
        l++;
       }while(str[l]);
    }
  wrefresh(jan[ind_jan_act]);
  actrjan1(c,d,a,b);
  ind_jan_act++;
}

jan_supaltlat(arr_refalt,n1,arr_refcomp,n2,str,hv)
int *arr_refalt,*arr_refcomp;
char *str;
int n1,n2,hv;
{
   int l,m1,a=0,b=0,c,d,e;

   e=strlen(str);

   for(l=0;l<n1;l++){
     m1=arr_refalt[l];
     a=a+jan_sup[m1].altura;
      }
   a=a-n1+1;
   for(l=0;l<n2;l++)
    { m1=arr_refcomp[l];
     b=b+lista_jan[m1].comp;
     }
   b=b-n2+1;
   c=jan_sup[arr_refalt[0]].coord_x;
   d=lista_jan[arr_refcomp[0]].coord_y;
   jan[ind_jan_act]=newwin(a,b,c,d);
   box(jan[ind_jan_act],0,0);
   l=0;
   if(hv==0)
```

```
else{
      do{
        mvwaddch(jan[ind_jan_act],a/2-e/2+1+l,b/2-1,str[l]);
        if (str[++l]=='.')
            mvwaddch(jan[ind_jan_act],a/2-e/2+l,b/2,str[l]);
        else
            l--;
        l++;
       } while(str[l]);
      }
  wrefresh(jan[ind_jan_act]);
  actrjan(c,d,a,b);
  ind_jan_act++;
}

jan_vlatssup(arr_refalt,n1,arr_refcomp,n2,str,hv)
int *arr_refalt,*arr_refcomp;
char *str;
int n1,n2,hv;
{
  int e,a=0,b=0,c,d,l;

  e=strlen(str);
  for (l=0;l<n1;l++)
   a=a+lista_jan[arr_refalt[l]].altura;
  a=a-n1+1;
  for(l=0;l<n2;l++)
   b=b+lista_jan[arr_refcomp[l]].comp;
  b=b-n2+1;
  c=lista_jan[arr_refalt[0]].coord_x;
  d=lista_jan[arr_refcomp[0]].coord_y;
  term(c,d);
  jan[ind_jan_act]=newwin(a,b,c,d);
  box(jan[ind_jan_act],0,0);
  l=0;
  if(hv==0)
    mvwaddstr(jan[ind_jan_act],a/2,b/2-e/2-1,str);
  else{
      do{
        mvwaddch(jan[ind_jan_act],a/2-e/2+1+l,b/2-1,str[l]);
        if (str[++l]=='.')
          mvwaddch(jan[ind_jan_act],a/2-e/2+l,b/2,str[l]);
```

BIBLIOTECA F.E. PORTO

```
  else
          l--;
         l++;
         }while(str[l]);
      }
  wrefresh(jan[ind_jan_act]);
  actrjan(c,d,a,b);
  ind_jan_act++;
}

novo_ecra()
{
   n=0;
   n1=0;
   getch();
   scroll(stdscr);
   wrefresh();
   actual_x=1;
   actual_y=1;
}
```

FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO
BIBLIOTECA
0000101606