

---

**registro (campos)**

idia	imes	iano	dia	mes	ano	nome	origem	destino	data	contvoo	contespera	lista[15]
------	------	------	-----	-----	-----	------	--------	---------	------	---------	------------	-----------

voo[0].  
voo[1].

origem	destino	data	contvoo	contespera	lista[15]
origem	destino	data	contvoo	contespera	lista[15]

lista[15] (voo+fila de espera)

**fila de voo**

0	
1	
2	
3	
4	

**fila de espera**

5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

**typedef struct Tdados{ //registro**

int idia;//dia inicial do sistema  
int imes;//mes inicial do sistema  
int iano;//ano inicial do sistema

int dia;  
int mes;  
int ano;

char nome[50];  
char data[11];  
int origem;  
int destino;

int contvoo;//contador de voo  
int contespera;//contador de fila de espera

char lista[16][50]; //lista de passageiros para o voo e fila de espera

**}Tdados; // registro de dados**

C:\GB\_Algoritmos\_Registro\Projeto\_GB\bin\Debug\Projeto\_GB.exe

-----  
DATA DO SISTEMA ..: 01/12/2012  
-----

----- SEJA BEM VINDO -----  
                                  A  
----- VOE FÁCIL LINHAS AÉREAS INTELIGENTES -----  
-----

----- MENU PRINCIPAL -----  
-----

- 1 - Agendar voo.
- 2 - Cancelar passageiro.
- 3 - Lista de relatórios.
- 0 - Sair.

-----  
Escolha uma das opções acima: \_

```
26
27 int main(void) {
28     setlocale(LC_ALL, "Portuguese"); //acentua as palavras corretamente
29
30     //declaracao de variaveis
31     int flag,i;
32     Tdados voo[TAM]; //variavel do tipo registro
33     Tdados aux; //variavel do tipo registro
34
35     //inicializacao de variaveis
36     aux=IniciaTdados(); //zerando a variavel auxiliar
37
38     for(i=1;i<TAM;i++){ //inicia zerado
39         voo[i]=IniciaTdados(); //zerando todos campos de cada posicao do vetor de registro
40     }
41     atualdata(voo,&aux); //chamando data atual
42
43     flaq=1;
44     while(flag!=0){ //enquanto for verdadeiro
45
46         flaq=menu(voo,&aux); //chama o menu principal
47
48     }
49
50     return(0); //retorno da funcao do main
51 }
52
```

```
C:\GB_Algoritmos_Registro\Projeto_GB\bin\Debug\Projeto_GB.exe

-----
Nome do passageiro .:
eron

-----
1 Aracajú
2 Belém
3 Belo Horizonte
4 Brasília
5 Florianópolis
6 Fortaleza
7 Goiânia
8 João Pessoa
9 Natal
10 Porto Alegre
11 Recife
12 Rio de Janeiro
13 Salvador
14 São Luiz
15 São Paulo
-----

Informe o numero correspondente a origem .:
1
```

```

77 char *valornome() { //captura o valor de voo
78     espacos(); // inserir alguns enter na tela
79     char nome[50];
80     printf("-----\n");
81     printf(" Nome do passageiro .: \n");
82     printf("-----\n");
83     fflush(stdin); //limpar buffer de entrada antes da leitura
84     gets(nome); //capturar variavel
85    strupr(nome); //deixar string maiuscula
86     fflush(stdin); //limpar buffer depois da leitura
87     espacos(); //inserir alguns enter na tela
88
89     return nome; //retorna valor da funcao
90 }
91
92 char *validanome() { //validacao de nome
93     char nome[50];
94     int flag;
95     flaq=0;
96     do { //repita enquanto o nome estiver VAZIO
97         system("cls");
98         if (flaq==1) { //se tiver vazio aparece essa mensagem
99             printf("\n\n\n-----\n");
100             printf(" !!! O CAMPO NOME NÃO PODE SER VAZIO !!!\n");
101             printf("-----\n");
102         }
103         strcpy(nome, valornome()); //atribui o valor retornado da funcao para nome
104
105         if (strcmp(nome, "")==0) { //se tiver vazio flaq recebe 1
106             flaq=1;
107         } else { //se nao tiver vazio flaq recebe zero
108             flag=0;
109         }
110         while (flaq==1); //compara se o nome esta vazio enquanto for 1 repete
111
112     return nome; //retorno da funcao

```

```
C:\GB_Algoritmos_Registro\Projeto_GB\bin\Debug\Projeto_GB.exe

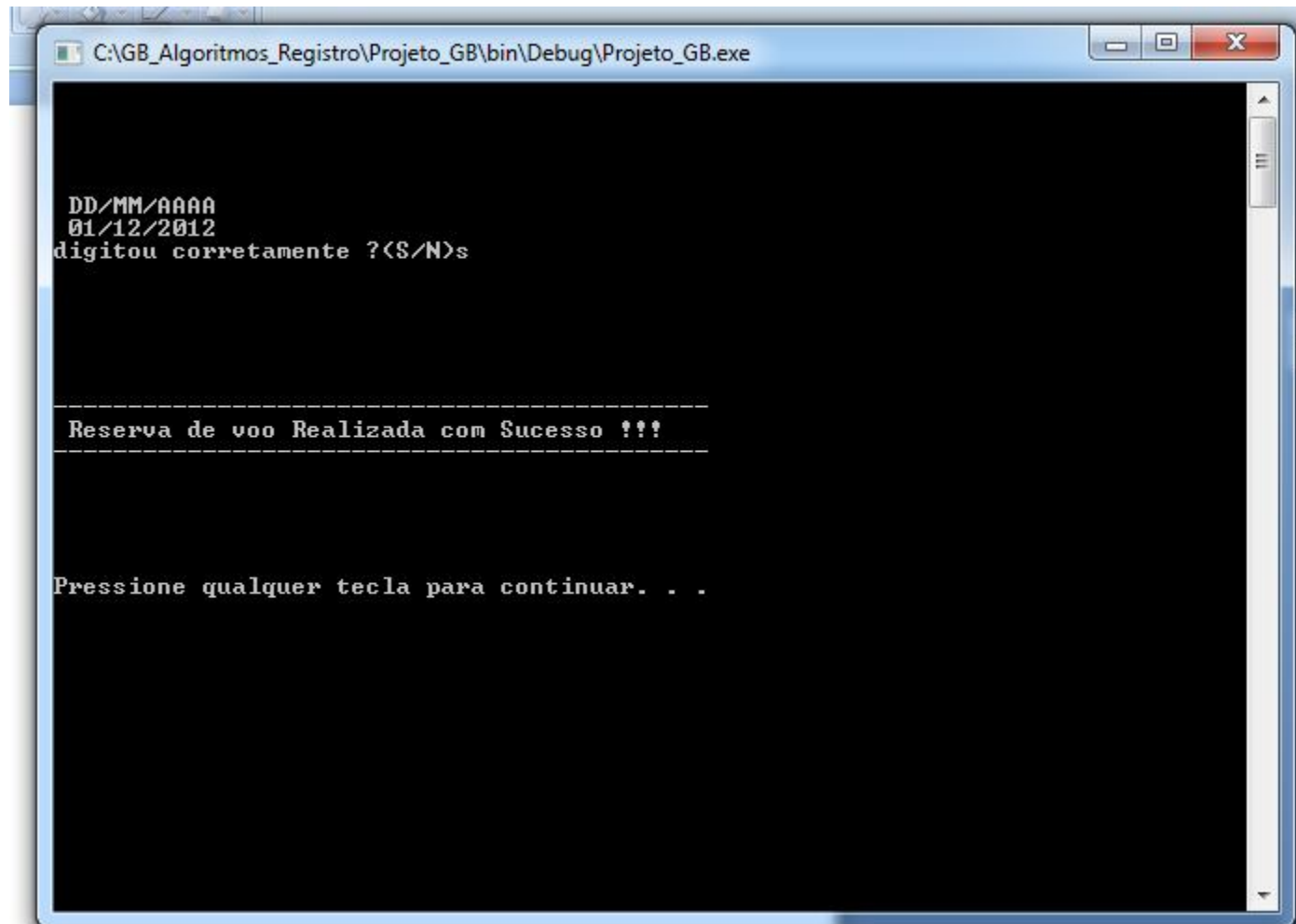
9 Natal
10 Porto Alegre
11 Recife
12 Rio de Janeiro
13 Salvador
14 São Luiz
15 São Paulo
-----
Informe o numero correspondente a origem .:
1
-----
2 Belém
3 Belo Horizonte
4 Brasília
5 Florianópolis
6 Fortaleza
7 Goiânia
8 João Pessoa
9 Natal
10 Porto Alegre
11 Recife
12 Rio de Janeiro
13 Salvador
14 São Luiz
15 São Paulo
-----
Informe o numero correspondente ao destino desejado .:
2
```



```

1
2  char v[16][20]={ // declaracao de array global
3  "",//0
4  "Aracajú",//1
5  "Belém",//2
6  "Belo Horizonte",//3
7  "Brasília",//4
8  "Florianópolis",//5
9  "Fortaleza",//6
10 "Goiânia",//7
11 "João Pessoa",//8
12 "Natal",//9
13 "Porto Alegre",//10
14 "Recife",//11
15 "Rio de Janeiro",//12
16 "Salvador",//13
17 "São Luiz",//14
18 "São Paulo",//15
19 };|
20
21 typedef struct Tdados{ //registro
22     int idia;//dia inicial do sistema
23     int imes;//mes inicial do sistema
24     int iano;//ano inicial do sistema
25     int dia;
26     int mes;
27     int ano;
28     char nome[50];
29     char data[11];
30     int origem;
31     int destino;
32     int contvoo;//contador de voo
33     int contespera;//contador de fila de espera
34     char lista[16][50]; //lista de passageiros para o voo e fila de espera
35 }Tdados;// registro de dados
36

```





```
115  int valororigem() { //captura valor de origem
116      int origem;
117      listaaeroporos(0); //exibe lista de aeroportos SEM exclusao de nomes
118      printf("-----\n");
119      printf(" Informe o numero correspondente a origem .: \n");
120      printf("-----\n");
121      scanf("%d",&origem); //captura o valor para origem
122      fflush(stdin); //limpar buffer depois da leitura
123      espacos(); //inserir enter na tela
124
125      return origem; //retorno da funcao
126  }
127
128  int validaorigem() { //validacao de origem
129      int valor, flag;
130      flag=0;
131      do { //repita ate que o valor seja entre 1 e 15
132
133          if(flag==1) { //se repetiu aparece a mensagem ...
134              system("cls");
135              printf("\n\n\n-----\n");
136              printf(" !!! O CAMPO ORIGEM DEVE TER APENAS UM DOS NUMEROS LISTADOS !!!\n");
137              printf("-----\n");
138          }
139          valor=valororigem(); //atribui valor retornado da funcao para a variavel valor
140
141          if(valor<1 || valor>15) { //se o valor eh menor que 1 ou maior que 15 repete
142              flag=1;
143          } else { //senao sai do loop
144              flag=0;
145          }
146      } while(flag==1);
147      return valor; //retorno da funcao
148  }
149
```

```

150 int valordestino(int origem){//captura valor de destino
151     int destino,flag;
152     flag=0;
153     do{//REPITA enquanto DESTINO for igual a ORIGEM
154
155         if(flag==1){//se repitiu aparece a mensagem ...
156             system("cls");//limpa tela
157             printf("\n\n\n-----\n");
158             printf(" !!! O CAMPO DESTINO NÃO PODE SER IGUAL A ORIGEM !!!\n");
159             printf("-----\n");
160         }
161         listaaeroportos(origem);//mostra lista de aeroportos excluindo a origem
162         printf("-----\n");
163         printf(" Informe o numero correspondente ao destino desejado .: \n");
164         printf("-----\n");
165         scanf("%d",&destino);//captura valor de destino
166         fflush(stdin);//limpar buffer depois da leitura
167         espacos();//inserir enter na tela
168
169         if(origem==destino){//se o valor de origem é igual a destino flag recebe
170             flag=1;
171         }else{//senao sai do loop
172             flag=0;
173         }
174
175     }while(flag==1); //repete enquanto origem for igual a destino
176     return destino;//retorno da funcao
177 }
178
179 int validadestino(int dest){//validacao de destino
180     int valor,flag;
181     flag=0;
182     do{//REPITA ate que destino seja entre 1 e 15
183         if(flag==1){//se repitiu aparece a mensagem ...
184             printf("\n\n\n-----\n");
185             printf(" !!! O CAMPO DESTINO DEVE TER APENAS UM DOS NUMEROS LISTADOS !!!\n");
186             printf("-----\n");
187         }
188
189         valor=valordestino(dest);//recebe retorno da funcao de destino
190         if(valor<1 || valor>15){//se o valor eh menor que 1 ou maior que 15 repete
191             flag=1;
192         }else{//senao sai do loop
193             flag=0;
194         }
195     }while(flag==1); //sai do loop somente quando for DIFERENTE de 1
196     return valor;//retorno da funcao

```

```
C:\GB_Algoritmos_Registro\Projeto_GB\bin\Debug\Projeto_GB.exe

-----
DATA DO SISTEMA ..: 01/12/2012
-----

----- SEJA BEM VINDO -----
              A
----- VOE FÁCIL LINHAS AÉREAS INTELIGENTES -----

----- MENU PRINCIPAL -----

1 - Agendar voo.
2 - Cancelar passageiro.
3 - Lista de relatórios.
0 - Sair.

Escolha uma das opções acima: 2_
```

C:\GB\_Algoritmos\_Registro\Projeto\_GB\bin\Debug\Projeto\_GB.exe

-----  
Nome do passageiro .:

paulo

- 
- 1 Aracajú
  - 2 Belém
  - 3 Belo Horizonte
  - 4 Brasília
  - 5 Florianópolis
  - 6 Fortaleza
  - 7 Goiânia
  - 8 João Pessoa
  - 9 Natal
  - 10 Porto Alegre
  - 11 Recife
  - 12 Rio de Janeiro
  - 13 Salvador
  - 14 São Luiz
  - 15 São Paulo
- 

Informe o numero correspondente a origem .:

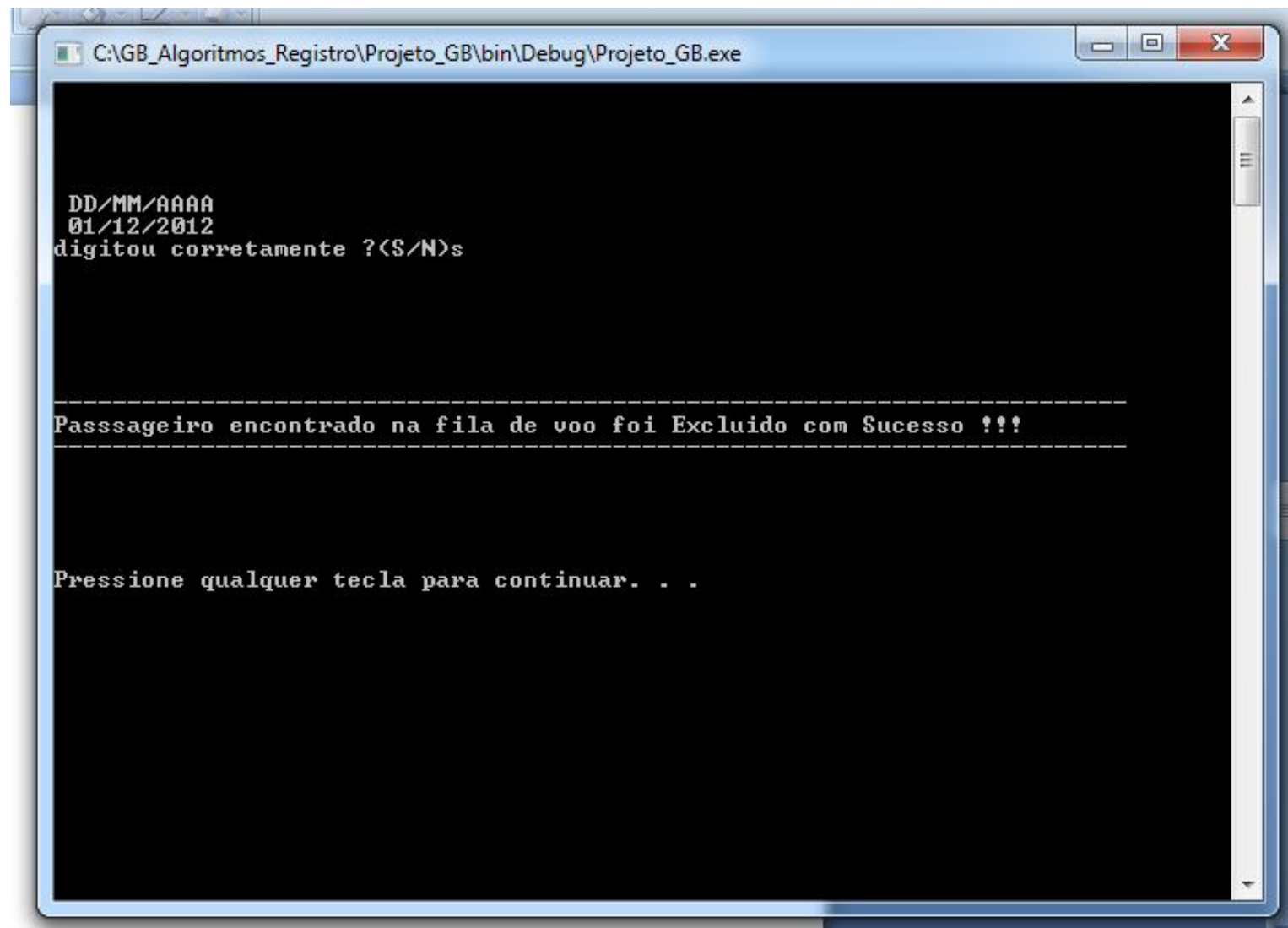
1



```

913
914 void metodoexclusao(Tdados voo[TAM], Tdados *aux) { //procedimento de exclusao
915     char flag[15];
916     int posvoocad, pe; //posicao voo e posicao exclusao
917
918     inserirpass(voo, aux); //capturar nome, origem, destino, data
919     posvoocad = getPosvoo(voo, aux); //encontrar a posicao do registro que encontra esse passageiro
920     pe = getPosExcluido(voo[posvoocad].lista, aux); //encontrar a posicao da lista que encontra esse passageiro
921
922     if (pe != -1) { //encontrou a posicao na lista de passageiros
923         excluirmnome(voo[posvoocad].lista[pe]); //excluir o passageiro
924         copiavetor(voo[posvoocad].lista); //copia a lista original para um segundo vetor ignorando campos vazios e devolve a lista
        original por referencia
925
926         if (pe >= 0 && pe <= 4) { strcpy(flag, "fila de voo"); } //se a posicao que foi encontrada estiver entre 0 e 4 entao foi excluido
        do voo
927     else { strcpy(flag, "Fila de Espera"); } //se a posicao que foi encontrada estiver entre 5 e 14 entao foi excluido da fila de
        espera
928
929     voo[posvoocad].contvoo = getnumeropassvoo(voo[posvoocad].lista, aux); //obtem o numero de passageiros atual de fila de voo
930     voo[posvoocad].contespera = getnumeropassvooespera(voo[posvoocad].lista, aux); //obtem o numero de passageiros atual de fila de
        espera
931     espacos(); //inserir enter na tela
932     printf("-----\n");
933     printf("Passageiro encontrado na %s foi Excluido com Sucesso !!!\n", flag);
934     printf("-----\n");
935     espacos(); //inserir enter na tela
936     system("pause"); //pausar a tela para permitir que o usuario leia
937 } else { //se nao encontrou o nome do passageiro fornecido entao diga ...
938     espacos(); //inserir enter na tela
939     printf("-----\n");
940     printf("Não foi possível cancelar pois, \n");
941     printf("o usuario informado não foi encontrado nessa origem, destino e data !!!\n");
942     printf("-----\n");
943     espacos(); //inserir enter na tela
944     system("pause"); //pausar a tela para permitir que o usuario leia
945 }
946
947 }
948

```



```
C:\GB_Algoritmos_Registro\Projeto_GB\bin\Debug\Projeto_GB.exe

DD/MM/AAAA
01/12/2012
digitou corretamente ?(S/N)s

-----
Passageiro encontrado na fila de voo foi Excluido com Sucesso !!!
-----

Pressione qualquer tecla para continuar. . .
```

C:\GB\_Algoritmos\_Registro\Projeto\_GB\bin\Debug\Projeto\_GB.exe

-----  
DATA DO SISTEMA ..: 01/12/2012  
-----

----- SEJA BEM VINDO -----  
                                  A  
----- VOE FÁCIL LINHAS AÉREAS INTELIGENTES -----  
-----

----- MENU PRINCIPAL -----  
-----

- 1 - Agendar voo.
  - 2 - Cancelar passageiro.
  - 3 - Lista de relatórios.
  - 0 - Sair.
- 

Escolha uma das opções acima: 3\_



```

1028 int menu(Tdados voo[TAM], Tdados *aux) { //funcao seletiva do menu principal
1029 int select;
1030
1031     do{
1032         select=m1(voo,aux); //atribui o retorno da funcao de menu para a variavel
1033     }while(!(select>=0 && select<=3)); // repete enquanto nao estiver no intervalo de 0 a 3
1034
1035     switch (select){
1036     case 1:
1037         metodocadastro(voo,aux); // Chamando a lista de marcacao voo
1038         break; //interrompendo o switch
1039     case 2:
1040         metodoexclusao(voo,aux); // chamando a cancelamento passageiro
1041         break; //interrompendo o switch
1042     case 3:
1043         relatorios(voo,aux); // chamando a lista de relatorio
1044         break; //interrompendo o switch
1045     case 0:
1046         exit(0); // Fechar do programa (ignorando tudo)
1047         break; //interrompendo o switch
1048     }
1049     return select; //retorno da funcao
1050 }
1051

```

C:\GB\_Algoritmos\_Registro\Projeto\_GB\bin\Debug\Projeto\_GB.exe

----- VOE FÁCIL LINHAS AÉREAS INTELIGENTES -----

----- MENU PRINCIPAL -----

- 1 - Agendar voo.
- 2 - Cancelar passageiro.
- 3 - Lista de relatórios.
- 0 - Sair.

Escolha uma das opções acima: 3

----- LISTA DE RELATORIOS -----

- 1 - Lista de vôos criados.
- 2 - Lista de vôos Confirmados.
- 3 - Lista dos passageiros de um determinado voo.
- 4 - Lista da espera de um determinado voo.
- 0 - Voltar ao menu anterior.

Escolha uma das opções acima: 1

```

1051
1052 void relatorios(Tdados voo[TAM], Tdados *aux) {
1053     int select;
1054
1055     do{
1056         select=m2(); //atribui o retorno da funcao
1057     }while(!(select>=0 && select<=4)); // repete ateh que numero do relatorio desejado esteja no intervalo de 0 a 4
1058
1059     switch (select){
1060         case 1:
1061             exhibe_r1(voo,aux); // Chamando a lista de Voos Criados
1062             system("pause"); //pausar a tela para permitir que o usuario leia
1063             break; //interrompendo o switch
1064         case 2:
1065             exhibe_r2(voo,aux); // Chamando a lista de Voos Confirmados
1066             system("pause"); //pausar a tela para permitir que o usuario leia
1067             break; //interrompendo o switch
1068         case 3:
1069             exhibe_r3_lista(voo,aux); // Chamando a lista de passageiros de um determinado Voo
1070             system("pause"); //pausar a tela para permitir que o usuario leia
1071             break; //interrompendo o switch
1072         case 4:
1073             exhibe_r4_lista(voo,aux); // Chamando a lista de Espera de um determinado Voo
1074             system("pause"); //pausar a tela para permitir que o usuario leia
1075             break;
1076         case 0:
1077             menu(voo,aux); // voltar para menu principal
1078             break; //interrompendo o switch
1079     }
1080
1081 }
1082

```

C:\GB\_Algoritmos\_Registro\Projeto\_GB\bin\Debug\Projeto\_GB.exe

### Lista de Vôos Criados

```
-----  
Origem .....: Aracajú  
Destino .....: Belém  
Data .....: 01/12/2012  
Nº de passageiros .....: 2  
Nº de passageiros na fila de espera ..: 0  
-----
```

Pressione qualquer tecla para continuar. . .

```

758
759 void cabecalhorelatorio1(char palavra[20], Tdados voo[TAM], Tdados *aux, int total) { //procedimento de cabecalho para o relatorio 1
760 int i;
761     if(total== -1) { // nao axou posicao livre com getpossilivre voo é sinal de que todas todos registros estao lotados entao
imprimi tudo...
762         total=TAM;
763     }
764
765     printf(" Lista de %s\n", palavra);
766     printf("-----\n");
767     for(i=0; i<total; i++) { //imprime todos voos cadastrados no sistema...
768     printf(" Origem .....: %s\n", v[voo[i].origem]);
769     printf(" Destino .....: %s\n", v[voo[i].destino]);
770     printf(" Data .....: %s\n", voo[i].data);
771     printf(" N° de passageiros .....: %d\n", voo[i].contvoo);
772     printf(" N° de passageiros na fila de espera ...: %d\n", voo[i].contespera);
773     printf("-----\n");
774     }
775     if(i==0) { //se o indice i do for continuar zero é por que nao tem nenhum voo criados ainda..
776     espacos(); //insere enter na tela
777     printf("-----\n");
778     printf(" NO MOMENTO NÃO EXISTEM VÔOS CRIADOS !!! \n");
779     printf("-----\n");
780     espacos(); //insere enter na tela
781     }

```

C:\GB\_Algoritmos\_Registro\Projeto\_GB\bin\Debug\Projeto\_GB.exe

### Lista de Vôos Confirmados

```
-----  
Origem .....: Aracajú  
Destino .....: Belém  
Data .....: 01/12/2012  
Nº de passageiros .....: 2  
Nº de passageiros na fila de espera ...: 0  
-----
```

Pressione qualquer tecla para continuar. . . \_



```

784
785 void cabecalhorelatorio2(char palavra[20],Tdados voo[TAM],Tdados *aux,int total){//procedimento de cabecalho para o relatorio 2
786 int i,flag;//variaveis
787 flag=0;//atribuindo zero a flag
788 if(total==-1){// nao axou posicao livre com getposslivre voo é sinal de que todas todos registros estao lotados entao
imprimi tudo...
789     total=TAM;
790 }
791
792 printf(" Lista de %s\n",palavra);
793 printf("-----\n");
794 for(i=0;i<total;i++){//imprimi todos voo cadastrados porem ...
795 if(voo[i].contvoo>=2){//..so imprime se for MAIOR ou IGUAL a 2
796 printf(" Origem .....: %s\n",v[voo[i].origem]);
797 printf(" Destino .....: %s\n",v[voo[i].destino]);
798 printf(" Data .....: %s\n",voo[i].data);
799 printf(" N° de passageiros .....: %d\n",voo[i].contvoo);
800 printf(" N° de passageiros na fila de espera ...: %d\n",voo[i].contespera);
801 printf("-----\n");
802 flag=1;//atribui 1 a variavel para evitar que imprima o proximo if
803 }
804
805 }
806 if(flag==0){//se o indice i do for continuar zero é por que nao tem nenhum voo criados ainda..
807 espacos();//inserir enter na tela
808 printf("-----\n");
809 printf(" NO MOMENTO NÃO EXISTEM VÔOS CONFIRMADOS !!! \n");
810 printf("-----\n");
811 espacos();//inserir enter na tela
812 }
813 }
814
815 void exibe r1(Tdados voo[TAM],Tdados *aux){//procedimento de exibicao de relatorio 1
816 system("cls");//limpa a tela
817 int total;//variavel de limite para usar na no procedimento de cabecalho
818 total=getPosLivre voo(aux);//captura uma posicao livre de voo
819 cabecalhorelatorio1("Vôos Criados",voo,aux,total);//chama o procedimento de relatorio enviando total por parametro
820
821 printf("\n\n\n\n\n\n\n\n\n\n\n");//imprime alguns enter na tela
822 }
823
824 void exibe r2(Tdados voo[TAM],Tdados *aux){//procedimento de exibicao de relatorio 2
825 system("cls");//limpa tela
826 int total;//variavel de limite para usar na no procedimento de cabecalho
827 total=getPosLivre voo(aux);//captura uma posicao livre de voo
828 cabecalhorelatorio2("Vôos Confirmados",voo,aux,total);//chama o procedimento de relatorio enviando total por parametro
829

```



C:\GB\_Algoritmos\_Registro\Projeto\_GB\bin\Debug\Projeto\_GB.exe

# Lista de Vôo

-----  
Origem .: Aracajú  
Destino .: Belém  
Data .: 01/12/2012  
-----

1º. ERON  
2º. NICHOLAS

Pressione qualquer tecla para continuar. . .

```
749 void cabecalhorelatorio(char palavra[20], Tdados *aux) { //procedimento de cabecalho para o relatorio 3 e 4
750
751     printf(" Lista %s \n", palavra);
752     printf("-----\n");
753     printf(" Origem ...: %s\n", v[aux->origem]);
754     printf(" Destino ..: %s\n", v[aux->destino]);
755     printf(" Data ....: %s\n", aux->data);
756     printf("-----\n");
757 }
758
```

C:\GB\_Algoritmos\_Registro\Projeto\_GB\bin\Debug\Projeto\_GB.exe

### Lista da fila de Espera

-----  
Origem .: Aracajú

Destino .: Belém

Data .: 01/12/2012  
-----

1º. WILLIAM

2º. ERON

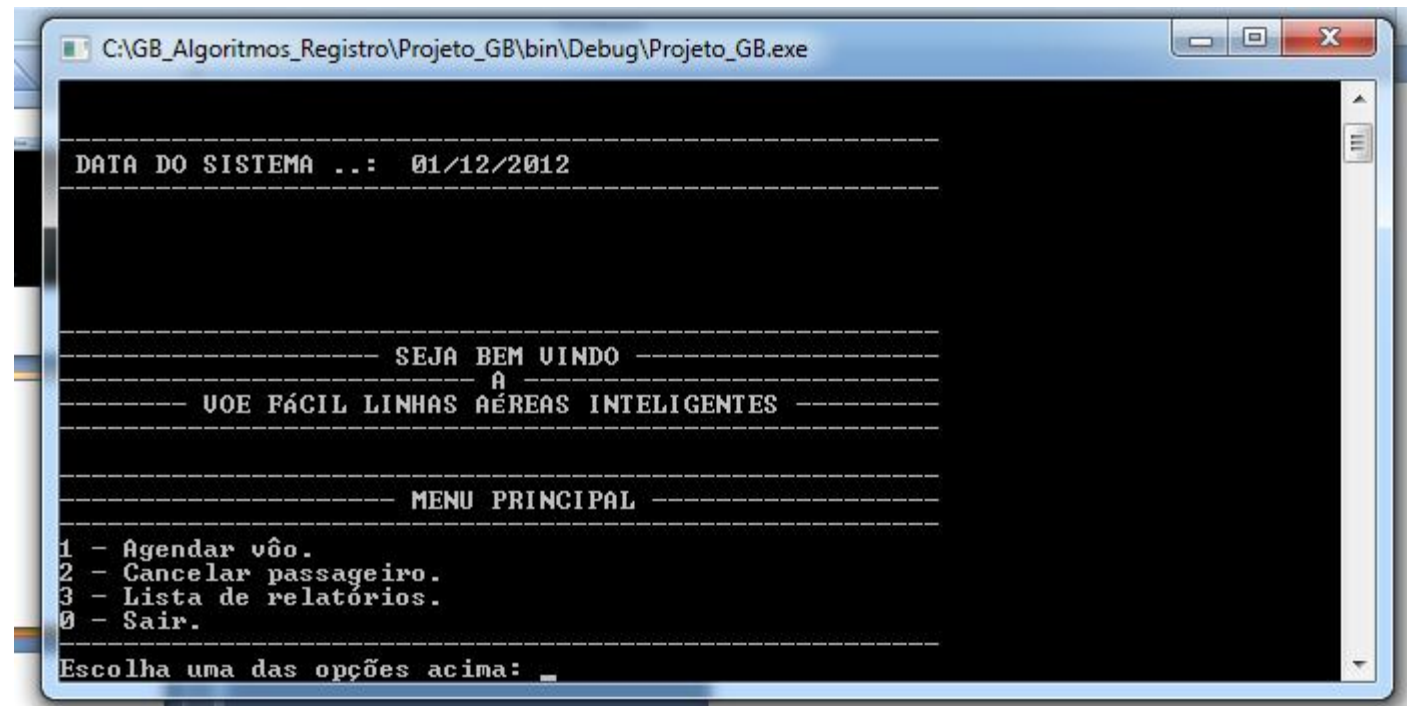
3º. PAULO

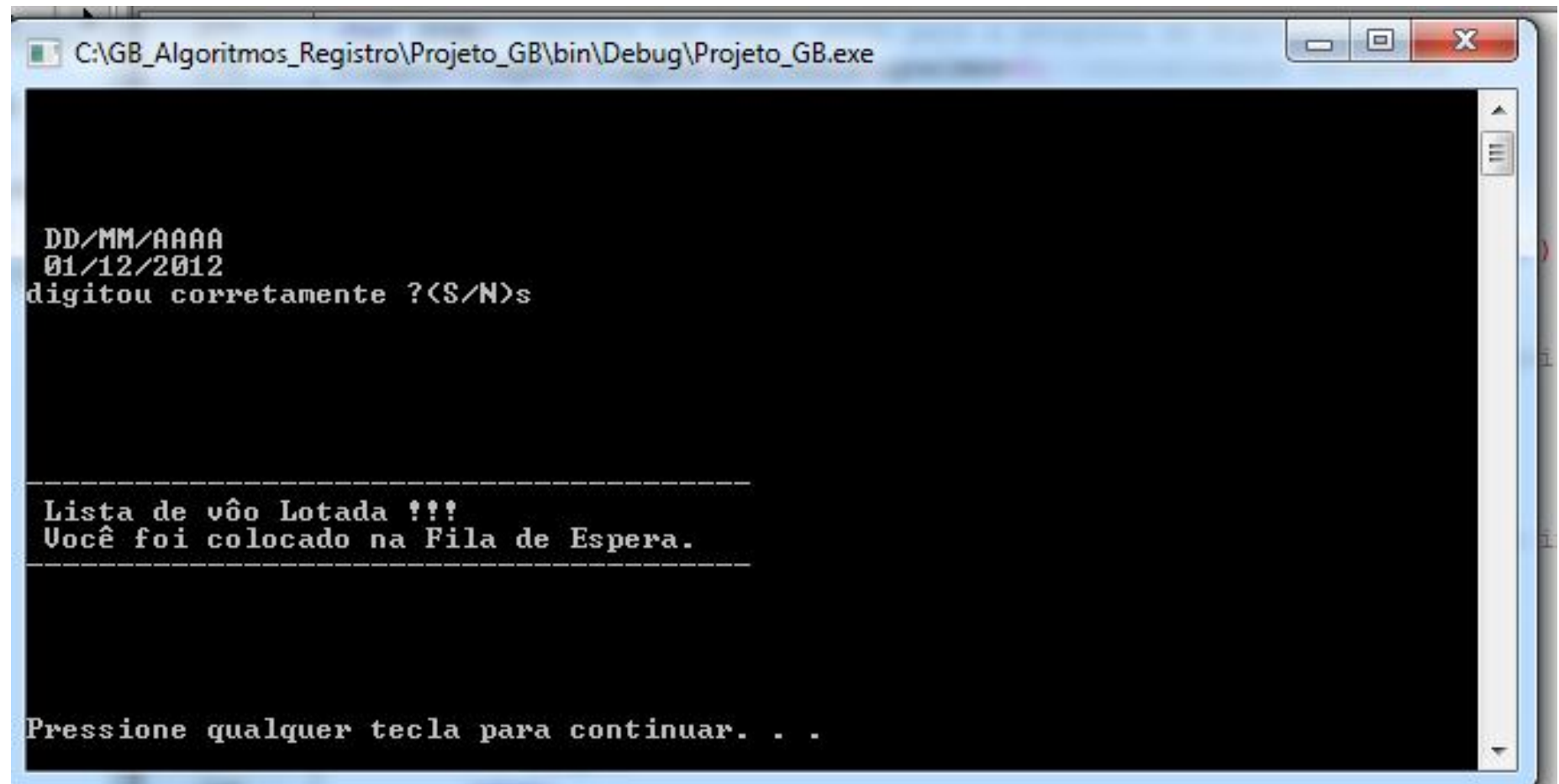
4º. IURI

5º. NICOLAS

Pressione qualquer tecla para continuar. . .

```
749 void cabecalhorelatorio(char palavra[20], Tdados *aux) { //procedimento de cabecalho para o relatorio 3 e 4
750
751     printf(" Lista %s \n", palavra);
752     printf("-----\n");
753     printf(" Origem ...: %s\n", v[aux->origem]);
754     printf(" Destino ..: %s\n", v[aux->destino]);
755     printf(" Data ....: %s\n", aux->data);
756     printf("-----\n");
757 }
758
```





```
C:\GB_Algoritmos_Registro\Projeto_GB\bin\Debug\Projeto_GB.exe

DD/MM/AAAA
01/12/2012
digitou corretamente ?(S/N)s

-----
Lista de voo Lotada !!!
Você foi colocado na Fila de Espera.
-----

Pressione qualquer tecla para continuar. . .
```

```

464 void datatostring(Tdados voo[TAM], Tdados *aux) { //converte os campos de data inteiro e os joga para uma string
465 vdata(voo, aux); //chama funcao de ler data
466 char data[11], dia[3], mes[3], ano[5];
467 itoa(aux->dia, dia, 10); // converte inteiro dia para string dia
468 itoa(aux->mes, mes, 10); // converte inteiro mes para string mes
469 itoa(aux->ano, ano, 10); // converte inteiro ano para string ano
470 if(aux->dia < 10) { //se o dia for menor que DEZ atribui um digito ZERO a primeira posicao da string
471     strcpy(aux->data, "0"); //adiciona ZERO a primeira posicao da string data
472     strcat(aux->data, dia); //concatena(soma, junta) a string data com a string dia
473 }
474 else { //senao copia normalmente o dia na sequencia dia/mes/ano
475     strcpy(aux->data, dia); //copia o valor da string dia para a string data
476 }
477 strcat(aux->data, "/"); //concatena(soma, junta) a string data um caracter '/'
478
479 if(aux->mes < 10) { //se o dia for menor que DEZ atribui um digito ZERO a primeira posicao da string
480     strcat(aux->data, "0"); //concatena(soma, junta) a string data o caracter '0'
481     strcat(aux->data, mes); //concatena(soma, junta) a string data a string mes
482 }
483 else { //senao copia normalmente o mes na sequencia dia/mes/ano
484     strcat(aux->data, mes); //concatena(soma, junta) a string data a string mes
485 }
486 strcat(aux->data, "/"); //concatena(soma, junta) a string data um caracter '/'
487
488 if(aux->ano < 10) { //se o dia for menor que DEZ atribui um digito ZERO a primeira posicao da string
489     strcat(aux->data, "00"); //concatena(soma, junta) a string data os caracteres '00'
490     strcat(aux->data, ano); //concatena(soma, junta) a string data a string ano
491 }
492 else {
493     if(aux->ano < 100) { //se o dia for menor que DEZ atribui um digito ZERO a primeira posicao da string
494         strcat(aux->data, "00"); //concatena(soma, junta) a string data os caracteres '00'
495         strcat(aux->data, ano); //concatena(soma, junta) a string data a string ano
496     } else {
497         if(aux->ano < 1000) { //se o dia for menor que DEZ atribui um digito ZERO a primeira posicao da string
498             strcpy(aux->data, "0"); //concatena(soma, junta) a string data o caracter '0'
499             strcat(aux->data, ano);
500         } else { //senao copia normalmente o ano na sequencia dia/mes/ano
501             strcat(aux->data, ano); //concatena(soma, junta) a string data a string ano
502         }
503     }
504 }
505 }
506
507 void atualdata(Tdados voo[TAM], Tdados *aux) { //captura a data atual para o sistema
508 vdata(voo, aux); //chamando a leitura de data
509 aux->idia = aux->dia; //joga o dia digitado para o dia inicial do programa
510 aux->imes = aux->mes; //joga o mes digitado para o mes inicial do programa
511 aux->iano = aux->ano; //joga o ano digitado para o ano inicial do programa
512
513 }

```