

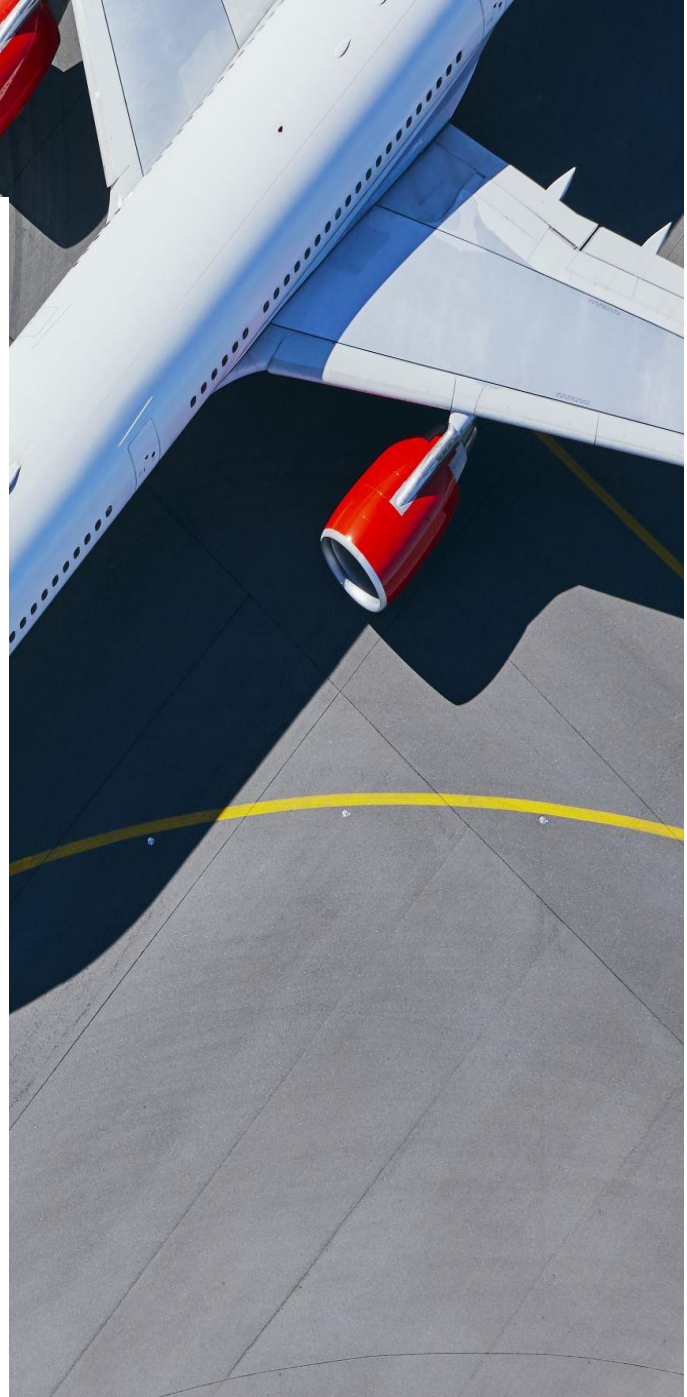
# Manual do Programador

---

24 JUNHO

---

Aero Portugal



---

## Índice

<b>Main Project – C .....</b>	<b>3</b>
<b>Main.C.....</b>	<b>3</b>
<b>int firstmenu() – Login/Signup .....</b>	<b>3</b>
<b>Resolução da Consola.....</b>	<b>4</b>
<b>Structures.h.....</b>	<b>4</b>
<b>Funcoes.h .....</b>	<b>5</b>
<b>void hidecursor().....</b>	<b>5</b>
<b>Int numfiles() .....</b>	<b>5</b>
<b>Int exists().....</b>	<b>6</b>
<b>Layouts.h.....</b>	<b>6</b>
<b>Int first_layout().....</b>	<b>6</b>
<b>Venda Bilhetes.h.....</b>	<b>8</b>
<b>Email.h .....</b>	<b>9</b>
<b>Consulta_reservas.h.....</b>	<b>9</b>
<b>Horarios.h .....</b>	<b>10</b>
<b>void horarios() .....</b>	<b>10</b>
<b>Perfil_cliente.h .....</b>	<b>11</b>
<b>Mail Sender – Python .....</b>	<b>13</b>

# Main Project – C

## Main.C

### int firstmenu() – Login/Signup

```
if(option==1) ///Login
{

    printf("Insira o seu email: ");
    gets(email_input);

    printf("\nInsira a pass: ");
    gets(password_input);
    for(int a=0;a<strlen(password_input);a++)
    {
        password_input[a]=password_input[a]-3;
    }
}
```

Nesta parte do Código o programa irá pedir ao utilizador o seu login e pass e irá codificá-la, tirando 3 posições a cada letra (ex: caso seja inserido a pass 456 a pass codificada será 123).

```
if((logon=fopen(file_input,"r"))!=NULL)
{
    fscanf(logon,"%i %i %s %s %i %s %s", &login.user_type, &id_cliente, login.nome1_cliente, login.nome2_cliente, &login.nif_cliente, login.encrypted_password, pais_cliente);
    fclose(logon);
    fflush(stdin);
    if(strcasecmp(login.encrypted_password, password_input)==0)
    {
        printf("\nLogado: %s\n", email_input);
        sleep(1);
        strcpy(login.email_cliente, email_input);
        strcpy(login.pais_cliente, pais_cliente);
        login.num_cliente=id_cliente;
        checker=true;
        return option;
    }
    else
    {
        checker=false;
        printf("Palavra Pass errada, por favor confira os seus dados\n\n");
        sleep(2);
        system("cls");
    }
}
else{
    checker=false;
    printf("\nCliente não existente, por favor confira os seus dados\n\n");
    sleep(2);
    system("cls");
}
```

Aqui, o programa irá comparar do ficheiro do user se o user e a pass coincidem, caso o user exista dirá password errada, caso contrário irá dizer não existente.

## Resolução da Consola

```
//Determina o tamanho da consola para o menu não ficar desformado em outros dispositivos
HWND console = GetConsoleWindow();
RECT ConsoleRect;
GetWindowRect(console, &ConsoleRect);
MoveWindow(console, ConsoleRect.left, ConsoleRect.top, 970/*largura em pixels*/, 500/*altura em pixels*/, TRUE);
//
```

Nesta parte o programa executa comandos da consola e irá alterar a resolução dela. Caso no futuro se pretenda alterar a resolução mude os valores na função `MoveWindow()`.

## Structures.h

```
struct Login{
    int user_type;
    int num_cliente;
    char nome1_cliente[255];
    char nome2_cliente[255];
    char email_cliente[255];
    char pais_cliente[255];
    char encrypted_password[255];
    int nif_cliente;
};

struct Dados_voo{
    char destino[99];
    int precos[2];
};
```

Neste header estão as structures usadas no programa, que estando aqui passam a ser globais

## Funcoes.h

### void hidecursor()

```
void hidecursor()
{
    HANDLE consoleHandle = GetStdHandle(STD_OUTPUT_HANDLE);
    CONSOLE_CURSOR_INFO info;
    info.dwSize = 100;
    info.bVisible = FALSE;
    SetConsoleCursorInfo(consoleHandle, &info);
}
```

Esta função dá uso à library windows.h e faz com que o cursor pare de piscar na consola

### Int numfiles()

```
int numfiles(char path[255])
{
    struct dirent *de; // Pointer para o directorio
    int numficheiros=0;
    DIR *dr = opendir(path);

    if (dr == NULL)
    {
        printf("Não foi possível abrir o directorio, por favor contacte o administrador");
        return 0;
    }

    while ((de = readdir(dr)) != NULL)
    {
        if ( !strcmp(de->d_name, ".") || !strcmp(de->d_name, "..") )
        {
            // vai ignorar os ficheiros "." e ".." pois não são ficheiros mas sim os directorios "base"
        }
        else
        {
            numficheiros++;
        }
    }
    closedir(dr);
    return numficheiros;
}
```

Esta função irá ler e retornar o número de ficheiros de um determinado directorio, esta função dá uso à biblioteca dirent.h

Esta função irá também ignorar os ficheiros fantasma "." e ".." dado que não são ficheiros reais, mas sim o número de directorios desde o executável até ao directorio pretendido.

## Int exists()

```
int exists(const char *fname) ///Verifica se o ficheiro existe
{
    FILE *file;
    if ((file = fopen(fname, "r")) != NULL)
    {
        fclose(file);
        return 1;
    }
    return 0;
}
```

Esta função irá verificar se certo ficheiro existe ou não retornando 1 se existir, caso contrário retornará 0.

## Layouts.h

### Int first\_layout()

```
int option=1;

char getcharacter;
//ponteiro para struct que armazena data e hora
struct tm *data_hora_atual;

//variável do tipo time_t para armazenar o tempo em segundos
time_t segundos;
while(1)
{
    //obtendo o tempo em segundos
    time(&segundos);

    //para converter de segundos para o tempo local
    data_hora_atual = localtime(&segundos);
}
```

Nesta parte da função, o programa irá inicializar a variável da opção para 1 e irá também “iniciar” as variáveis necessárias à biblioteca time.h para ir buscar o tempo no formato de 24 horas.

```

if(option==1){
    while(!kbhit()){
        time(&segundos);data_hora_atual = localtime(&segundos);
hidecursor(); //Caso esta função não seja chamada. O cursor irá ficar "maluco" a percorrer todas as linhas da consola pois não tem tempo para chegar ao fim antes do reinici
gotoxy(2,1);printf("
");textcolor(LIGHTGREEN);printf(".....");textcolor(GREEN+BLINK);printf("....."); textcolor(RED);printf
gotoxy(2,2);printf(" /
");textcolor(LIGHTGREEN);printf(".....");textcolor(GREEN+BLINK);printf("////");textcolor(LIGHTGREEN);pri
gotoxy(2,3);printf(" /
");textcolor(LIGHTGREEN);printf(".,.");textcolor(GREEN+BLINK);printf("/*");textcolor(LIGHTGREEN);print
gotoxy(2,4);printf(" |
");textcolor(LIGHTGREEN);printf(".,.");textcolor(GREEN+BLINK);printf("/*");textcolor(LIGHTGREEN);print
gotoxy(2,5);printf(" |
");textcolor(LIGHTGREEN);printf(".,.");textcolor(GREEN+BLINK);printf("/*");textcolor(LIGHTGREEN);print
gotoxy(2,6);printf(" |
");textcolor(LIGHTGREEN);printf(".,.");textcolor(GREEN+BLINK);printf("/*");textcolor(LIGHTGREEN);print
gotoxy(2,7);printf(" |
");textcolor(LIGHTGREEN);printf(".,.");textcolor(GREEN+BLINK);printf("/*");textcolor(LIGHTGREEN);print
gotoxy(2,8);printf(" |
");textcolor(LIGHTGREEN);printf(".,.");textcolor(GREEN+BLINK);printf("/*");textcolor(LIGHTGREEN);print
gotoxy(2,9);printf(" |
");textcolor(LIGHTGREEN);printf(".,.");textcolor(GREEN+BLINK);printf("/*");textcolor(LIGHTGREEN);print
gotoxy(2,10);printf(" ");gotoxy(25,10);printf("
São %i:%i:%i do dia %i/%i/%i
",data_hora_atual->tm_hour,data_ho
gotoxy(2,11);printf(" |"); gotoxy(25,11); puts("
");
gotoxy(2,12);printf(" |"); gotoxy(25,12); puts("
");
gotoxy(2,13);printf(" |"); gotoxy(25,13); puts("
");
gotoxy(2,14);printf(" |"); gotoxy(25,14); puts("
");
gotoxy(2,15);printf(" |"); gotoxy(25,15); printf("
");textbackground(LIGHTBLUE);printf(" Login ");textbackground(BLACK);printf("
");
gotoxy(2,16);printf(" |"); gotoxy(25,16); puts("
");
gotoxy(2,17);printf(" |"); gotoxy(25,17); printf("
");textbackground(LIGHTBLUE);printf(" Sign Up ");textbackground(BLACK);printf("
");
gotoxy(2,18);printf(" |"); gotoxy(25,18); puts("
");
gotoxy(2,19);printf(" |"); gotoxy(25,19); puts("
");textbackground(LIGHTBLUE);printf(" Ver o Manual do Utilizador ");textbackground(BLACK);printf("
");
gotoxy(2,20);printf(" |"); gotoxy(25,20); puts("
");
gotoxy(2,21);printf(" |"); gotoxy(25,21); printf("
");textbackground(LIGHTBLUE);printf(" Saia ");textbackground(BLACK);printf("
");
gotoxy(2,22);printf(" |"); gotoxy(25,22); puts("
");
gotoxy(2,23);printf(" |"); gotoxy(25,23); puts("
");
gotoxy(2,24);printf(" |"); gotoxy(25,24); puts("
");
gotoxy(2,25);printf(" |"); gotoxy(25,25); puts("
");
gotoxy(2,26);printf(" |"); gotoxy(25,26); puts("
");
gotoxy(2,27);printf(" |"); gotoxy(25,27); puts("
");
gotoxy(2,28);printf(" |"); gotoxy(25,28); puts("
");
time(&segundos);data_hora_atual = localtime(&segundos);
}

```

Aqui está um exemplo de um dos layouts, ao início do layout o tempo é atualizado assim como no fim. Este layout ficará em loop até que uma tecla seja pressionada.

Quando uma tecla for pressionada

```

getcharacter=getch();
if(getcharacter == 's' || getcharacter == 'S')
{
    if(option>=4)
    {
        option=4;
    }
    if(option<4)
    {
        option++;system("cls");
    }
    else{}
}

else if(getcharacter == 'w' || getcharacter == 'W')
{
    if(option<=1)
    {
        option=1;
    }
    if(option>1)
    {
        option--;system("cls");
    }
}

else if(getcharacter == '\r')
{
    system("cls");
    return option;
}

else
{
    option=option;
}

```

O programa irá identificar se é o W, o S, ou o ENTER; caso seja o W, irá retirar um valor à var option, subindo assim a seleção. Caso pressione S, irá incrementar o valor da option, descendo assim a seleção. Caso pressione a tecla ENTER irá retornar a opção selecionada.

# Venda Bilhetes.h

[illegible]

Nesta parte, o programa irá ler do txt com o nome do país inserido linha por linha o nome do país, o valor da viagem de ida e o valor da viagem de volta.

**Estes valores serão depois armazenados num array.**

**Caso o programa leia na string algum hífen ele irá automaticamente substituir por um espaço. Não se poderia escrever diretamente no txt com espaços pois o programa só iria ler até ao primeiro.**

**Selecionado o Destino o programa entrará na função `recibo()` passando os parâmetros de: origem, destino, ida e volta ou passagem única, preço de ida, preço de volta, email do cliente, NIF do cliente.**



## Email.h

```

encomenda_cliente=fopen(encomenda_cliente,"a+");
fprintf(encomenda_cliente,"%i\n",increase+1);
fclose(encomenda_cliente);
ficheiro=fopen(fnum_encomenda_final,"w");
ficheiro_temp=fopen("Files\\TEMP\\recibo.txt","w");
info_mail=fopen("Files\\TEMP\\email.txt","w");
fprintf(info_mail,email);

total=preco_voo_ida+preco_voo_volta;
pre_tax_price=total-(total/(1+0.23)*0.23);

if(roundtrip==1)
{
fprintf(ficheiro,"%s \nNIF: %i\nRecibo do voo:\n\n Voo de %s a %s: %.2f EUR\n Voo de %s a %s: %.2f EUR\n\n Iva Aplicado de acordo com a Taxa legal de 23% \nTotal Sem IVA: %.2f EUR\n\n Total: %.2f EUR\n",info_mail);
fprintf(ficheiro_temp,"%s \nNIF: %i\nRecibo do voo:\n\n Voo de %s a %s: %.2f EUR\n Voo de %s a %s: %.2f EUR\n\n Iva Aplicado de acordo com a Taxa legal de 23% \nTotal Sem IVA: %.2f EUR\n\n Total: %.2f EUR\n",info_mail);
}
else if(roundtrip==0)
{
fprintf(ficheiro,"%s \nNIF: %i\nRecibo do voo:\n\n Voo de %s a %s: %.2f Eur\n\n Iva Aplicado de acordo com a Taxa legal de 23% \nTotal Sem IVA: %.2f EUR\nTotal: %.2f Eur\n",info_mail);
fprintf(ficheiro_temp,"%s \nNIF: %i\nRecibo do voo:\n\n Voo de %s a %s: %.2f Eur\n\n Iva Aplicado de acordo com a Taxa legal de 23% \nTotal Sem IVA: %.2f EUR\n Total: %.2f EUR\n",info_mail);
}

fclose(info_mail);
fclose(ficheiro);
fclose(ficheiro_temp);

```

O programa aqui está a imprimir em 2 ficheiros, um que fica nos registos como encomenda numero x.txt e outro que é o temporário que o script em Python enviará. O programa também calcula aqui o valor sem IVA do voo usando uma fórmula matemática.

## Consulta reservas.h

```
strcpy(path1, "");
strcat(path1, "../Files/Users/Encomendas/");
strcat(path1, login.email_cliente);
strcat(path1, ".txt");

if(exists(path1)==1)
{
    file1=fopen(path1, "r");

    while (fgets (line, sizeof line, file1) != NULL )
    {
        loop++;
    }
    fclose(file1);
    int reservas[loop];
    strcpy(line, "");
    loop=0;

    file1=fopen(path1, "r");

    while (fgets ( line, sizeof line, file1) != NULL )
    {
        sscanf(line, "%i", &reservas[loop]);
        fflush(stdin);
        loop++;
    }
    fclose (file1);
}
```

---

O programa aqui irá ler do ficheiro do user que contém as reservas linha por linha e depois irá as abrir mostrando-as na tela.

## Horarios.h

void horarios()

O programa aqui irá perguntar ao utilizador que país ele pretende ver os horarios, caso o país não esteja na base de dados, irá apresentar uma mensagem de erro e retornar ao menu.

```
printf("Insira o Pais para o qual deseja consultar os horarios: ");
gets(origem);
strcat(path, "./Files/Voos/");
strcat(path, origem);

DIR *dr = opendir(path);    //Utilização da biblioteca dirent.h, define qual o directorio pretendido.

if (dr == NULL) // Irá retornar NULL caso não seja possivel abrir o directorio
{
    printf("Não temos atualmente os horarios para esse pais, por favor contacte o administrador.\n\n");
    sleep(2);
    return;
}
```

Caso contrário ele irá ler os ficheiros de chegadas e de partidas e imprimir no ecrã o correspondente aos voos das 3 horas seguintes, incluindo a atual.

```

FILE *partidas=fopen(filename_1,"r");

while (fgets ( line, sizeof line,partidas) != NULL )
{
    sscanf(line,"%s",horario[loop][0]);
    fflush(stdin);
    loop++;
}

fclose (partidas);
puts("\n\n    Partidas:\n");
for(int a=start;a<finish;a++){
    switch(a){
        case 24:
            printf("        %s 00:00\n",horario[0][0]);
            break;
        case 25:
            printf("        %s 01:00\n",horario[1][0]);
            break;
        default:
            if(a<10){
                printf("        %s 0%i:00\n",horario[a][0],a);
            }
            else{
                printf("        %s %i:00\n",horario[a][0],a);
            }
            break;
    }
}

FILE *chegadas=fopen(filename_2,"r");

```

## Perfil\_cliente.h

O programa irá perguntar ao user se quer ou não alterar os seus dados pessoais. Caso pressione 1 ser-lhe-á pedido para reinserir a sua palavra pass.

```

getchar=getch();
if(getchar=='1')
{
    puts("\n\nInsira a sua password atual para confirmar a sua identidade");
    gets(password_input);
    if(strcmp(password_input,decrypted_password)==0)
    {
        strcpy(decrypted_password,"");

        puts("\n\nDeseja alterar o seu nome? (S/n): ");
        getchar=getch();
        if(getchar=='S' || getchar=='s')
        {
            puts("\nInsira o seu Nome:");
            gets(login.nome1_cliente);
            puts("\nInsira o seu Apelido:");
            gets(login.nome2_cliente);
        }

        puts("\n\nDeseja alterar o seu país? (S/n): ");
        getchar=getch();
        if(getchar=='S' || getchar=='s')
        {
            puts("\nInsira o seu País:");
            gets(login.pais_cliente);
        }

        puts("\n\nDeseja alterar o seu NIF? (S/n): ");
        getchar=getch();
        if(getchar=='S' || getchar=='s')
        {
            puts("\nInsira o seu NIF:");
            scanf("%i",&login.nif_cliente);
            fflush(stdin);
        }

        puts("\n\nDeseja alterar a senha? (S/n): ");
        getchar=getch();
        if(getchar=='S' || getchar=='s')
        {
            do{
                puts("\nInsira a sua nova Senha:");
                gets(password_input);
                puts("\nRepita a sua nova Senha:");
                gets(password_input_2);
            } while(strcmp(password_input,password_input_2)!=0);
        }
    }
}

```

Depois caso a pass esteja correta irá perguntar se deseja alterar as várias coisas tendo de responder com Sim ou não às perguntas. Caso responda outra letra, será considerado como não.

# Mail Sender – Python

Este script foi criado com o âmbito de ter funcionalidades de rede na aplicação base de modo a conseguir enviar os recibos ao cliente.

Este script utilizou as bibliotecas “os.path” e “smtplib”

Foi utilizada a biblioteca para ler o path de onde o programa está a ser executado para conseguir abrir a subpasta no lugar certo

```
path=os.getcwd()+ '\\Files\\TEMP\\email.txt'
path1=os.getcwd()+ '\\Files\\TEMP\\recibo.txt'
```

A biblioteca smtplib foi utilizada para fazer a comunicação com os servidores de email.

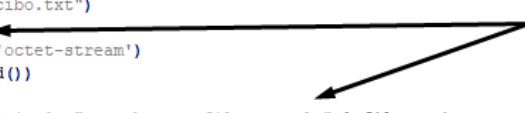
```
#codigo que faz a comunicação com o servidor
msg = MIMEMultipart()
msg['From'] = email
msg['To'] = send_to_email
msg['Subject'] = subject
msg = MIMEMultipart()
msg['From'] = email
msg['To'] = send_to_email
msg['Subject'] = subject

msg.attach(MIMEText(message, 'plain'))

# Setup the attachment
filename = os.path.basename("recibo.txt")
attachment = open(path1, "rb")
part = MIMEBase('application', 'octet-stream')
part.set_payload(attachment.read())
encoders.encode_base64(part)
part.add_header('Content-Disposition', "attachment; filename= %s" % filename)

# Attach the attachment to the MIMEMultipart object
msg.attach(part)

#deverá se substituir o ip do server smtp caso exita mudança de provedor de email. Neste caso,o ip é o do server de smtp da Amen.pt.
server = smtplib.SMTP('81.88.48.66')
server.starttls()
server.login(email, password)
text = msg.as_string()
server.sendmail(email, send_to_email, text)
server.quit()
```



Faz com que envie o recibo em anexo no email