

José Antonio Ruíz Miñán

```
void menuPasajeroPerfil(Estr_Usuario *usuario, int numUsuarios, Estr_Localidad *localidad, int numLocalidades, int i) 1
{
    int opc;

    while(opc!=5)
    {
        system("cls"); 2
        color(0, 3);
        printf("ID de usuario: "); 3
        color(0, 15);
        printf("%s\n", usuario[i].id_usuario);
        color(0, 3);
        printf("Nombre completo: "); 4
        color(0, 15);
        printf("%s\n", usuario[i].nonb_usuario);
        color(0, 3);
        printf("Localidad de residencia: "); 5
        color(0, 15);
        printf("%s\n", usuario[i].localidad);
        color(0, 3);
        printf("Tipo de perfil: "); 6
        color(0, 15);
        printf("%s\n", usuario[i].perfil);
        color(0, 3);
        printf("Usuario: "); 7
        color(0, 15);
        printf("%s\n", usuario[i].usuario);
        color(0, 15);
        printf("¿Que quiere hacer?\n"); 8
        color(0, 2);
        printf("(1)Modificar nombre completo.\n"); 9
        color(0, 4);
        printf("(2)Modificar localidad de residencia.\n"); 10
        color(0, 11);
        printf("(3)Modificar usuario.\n"); 11
        color(0, 6);
        printf("(4)Modificar contrasena.\n"); 12
        color(0, 14);
        printf("(5)Volver.\n"); 13
        color(0, 15);
        fflush(stdin);
        scanf("%i", &opc); 14
        system("cls"); 15
        switch(opc)
        {
            case 1:
                modificarPerfilNombre(usuario, numUsuarios, i); 16
                break;
            case 2:
                modificarPerfilLocalidad(usuario, numUsuarios, localidad, numLocalidades, i); 17
                break;
            case 3:
                modificarPerfilUsuario(usuario, numUsuarios, i); 18
                break;
            case 4:
                modificarPerfilContrasena(usuario, numUsuarios, i); 19
                break;
            case 5:
                return; 20
                break;
        }
    }
}
```

```

void modificarPerfilNombre(Estr_Usuario *usuario, int numUsuarios, int i) 16
{
    char nomb[21];

    printf("Introduzca su nuevo nombre completo (Maximo de 20 caracteres):\n"); 16.1
    pregunta(nomb, 21); //Pedimos el nuevo nombre. 16.2

    strcpy(usuario[i].nomb_usuario, nomb); //Copiamos el nuevo nombre en la posicion de la estructura correspondiente. 16.3
    actualizarUsuario(usuario, numUsuarios); //Actualizamos el fichero usuarios.txt 16.4

    printf("Su nombre completo se ha actualizado correctamente.\n"); 16.5
    system("PAUSE"); 16.6
}

void modificarPerfilLocalidad(Estr_Usuario *usuario, int numUsuarios, Estr_Localidad *localidad, int numLocalidades, int i) 17
{
    char loc[21];

    printf("Introduzca su nueva localidad de residencia (3 siglas):\n"); 17.1
    fflush(stdin);
    pregunta_localidad(localidad, numLocalidades, loc); //Pedimos la nueva localidad. 17.2

    strcpy(usuario[i].localidad, loc); //Copiamos la nueva localidad en la posicion de la estructura correspondiente. 17.3
    actualizarUsuario(usuario, numUsuarios); //Actualizamos el fichero usuarios.txt 17.4

    printf("Su localidad de residencia se ha actualizado correctamente.\n"); 17.5
    system("PAUSE"); 17.6
}

void modificarPerfilUsuario(Estr_Usuario *usuario, int numUsuarios, int i) 18
{
    int k=0, encontrado=0; 18.1
    char usua[6];

    printf("Introduzca su nuevo nombre de usuario (Maximo de 5 caracteres):\n"); 18.2
    pregunta(usua, 6); //Pedimos el nuevo username. 18.3

    for(k=0; k<numUsuarios&&encontrado==0; k++) //Comprobamos que no esta usado. 18.4
    {
        if(strcmp(usua, usuario[k].usuario)==0) 18.5
        {
            encontrado=1; 18.6
        }
    }

    if(encontrado==0) //Si no esta usado 18.7
    {
        printf("El nombre de usuario es valido.\n"); 18.8
        strcpy(usuario[i].usuario, usua); //Copiamos el nuevo username en la posicion de la estructura correspondiente. 18.9
        actualizarUsuario(usuario, numUsuarios); //Actualizamos el fichero usuarios.txt 18.10

        printf("Su nombre de usuario se ha actualizado correctamente.\n"); 18.11
        system("PAUSE"); 18.12
    }
    else 18.13
    {
        printf("El nombre de usuario ya esta siendo usado.\n"); 18.14
        system("PAUSE"); 18.15
    }
}

```

```

void modificarPerfilContraseña(Estr_Usuario *usuario, int numUsuarios, int i)19
{
    int encontrado=3, x=0; 19.1
    char contra[9], contra2[9]; 19.2

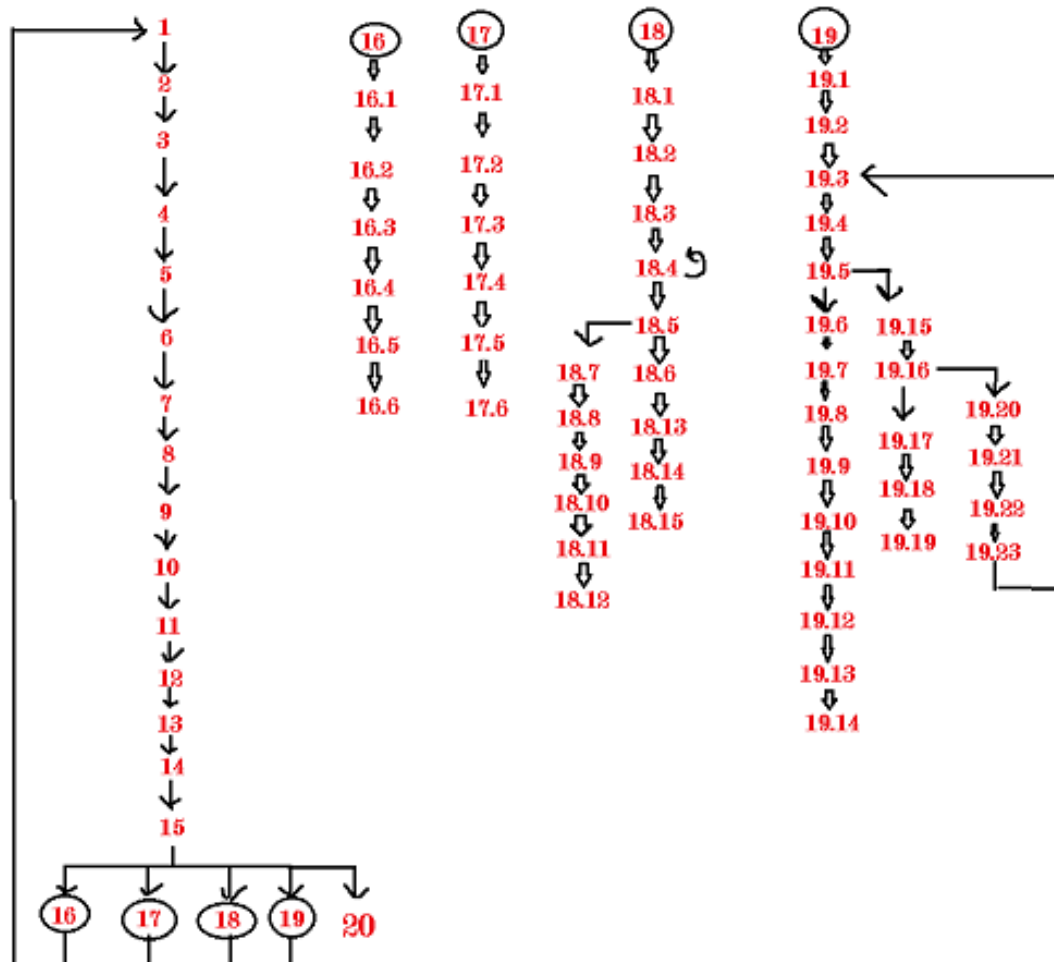
    while(encontrado>0&&x==0) 19.3
    {
        printf("Introduzca su antigua contraseña (Maximo de 8 caracteres):\n"); 19.4
        preguntar_contraseña(contra); 19.5

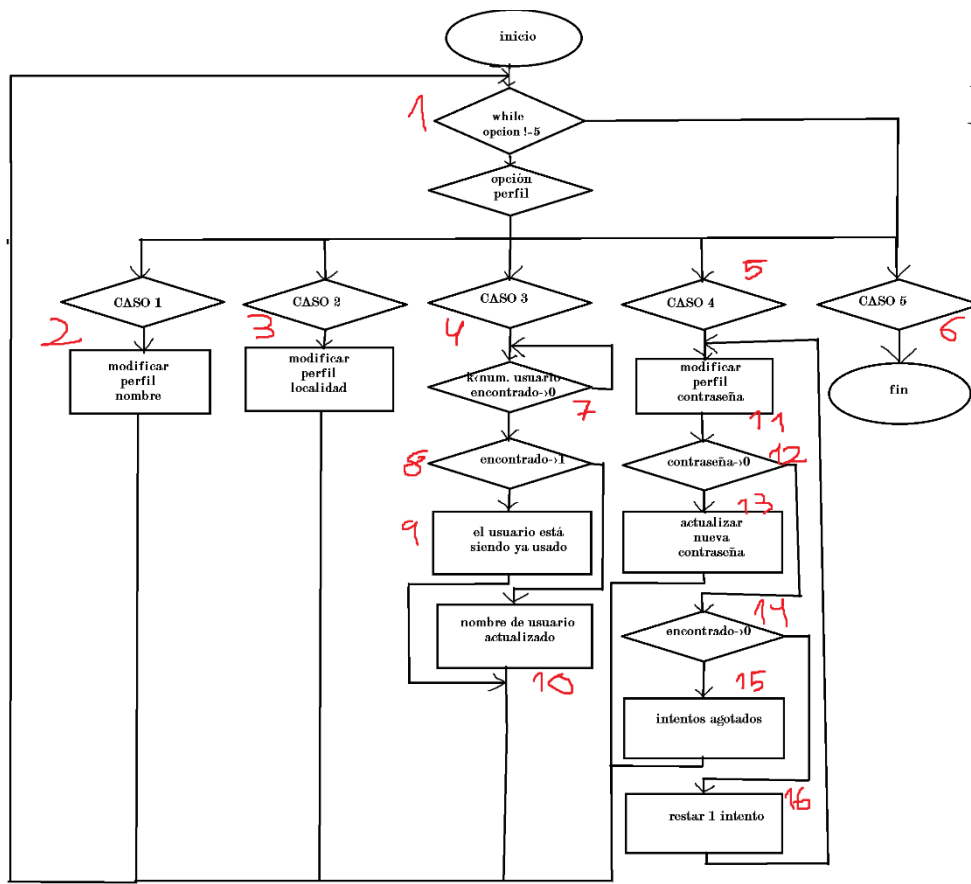
        if(strcmp(usuario[i].contrasena,contra)==0) //si la contraseña introducida es igual a la que tiene. 19.6
        {
            system("cls"); 19.7
            x=1; 19.8
            printf("Introduzca su nueva contraseña (Maximo de 8 caracteres):\n"); 19.9
            preguntar_contraseña(contra2); 19.10

            strcpy(usuario[i].contrasena, contra2); 19.11
            actualizarUsuario(usuario, numUsuarios); 19.12

            printf("\nSu contraseña se ha actualizado correctamente.\n"); 19.13
            system("PAUSE"); 19.14
        }
        else //si no es correcta 19.15
        {
            encontrado--; 19.16
            if(encontrado==0) //Cuando no tiene intentos 19.17
            {
                printf("\nIntentos agotados! :(\n"); 19.18
                system("PAUSE"); 19.19
            }
            else //si tiene intentos, se van restando. 19.20
            {
                *contra=NULL;
                printf("\nQueda(n) %i intentos.\n", encontrado); 19.21
                system("PAUSE"); 19.22
                system("cls"); 19.23
            }
        }
    }
}

```





$$V(G) \Rightarrow NA - NN + 2 \Rightarrow 29 - 21 + 2 \Rightarrow 10$$

$$V(G) \Rightarrow \text{Numero de regiones} \Rightarrow 10$$

$$V(G) \Rightarrow NNP + 1 \Rightarrow 9 + 1 \Rightarrow 10$$

- 1) 1-6
- 2) 1-2-1-6
- 3) 1-3-1-6
- 4) 1-4-7-8-9-1-6
- 5) 1-4-7-8-10-1-6
- 6) 1-4-7-7-8-9-1-6
- 7) 1-4-7-7-8-10-1-6
- 8) 1-5-11-12-13-1-6
- 9) 1-5-11-12-14-16-11-12-14-15-1-6
- 10) 1-5-11-12-14-16-11-12-13-1-6

Pruebas de caja negra

Cabecera: void menuPasajeroPerfil(Estr_Usuario *usuario, int numUsuarios, Estr_Localidad *localidad, int numLocalidades, int i)

Precondición: Haber iniciado sesión con un usuario, no un admin y seleccionar el menú de pasajero, no el de conductor.

Postcondición: Mostrar y editar los datos del usuario en el fichero "usuarios.txt".

- **Modificar nombre completo:** Escanea un nombre completo nuevo y sobrescribe el existente.
- **Modificar localidad de residencia:** Muestra una tabla con todas las localidades disponibles y al elegir una de ellas sobre escribe la anterior.
- **Modificar usuario:** Pregunta un nickname y cambia el actual.
- **Modificar contraseña:** Primero pregunta la antigua contraseña, a diferencia de los otros, si esta contraseña es incorrecta vuelve a pedírsela con un máximo de 3 intentos, si se le acaban los intentos vuelve al perfil, si acierta la contraseña pregunta la nueva contraseña y sobrescribe la antigua.