Equipo

[Nombre de la empresa]  [Dirección de la compañía]

Manual usuario/tecnico

Contenido

[Bienvenido al Sistema!!!! 3](#_Toc75524019)

[Clientes 3](#_Toc75524020)

[1.1 Altas 3](#_Toc75524021)

[1.2 Listados 3](#_Toc75524022)

[1.3 Búsquedas 3](#_Toc75524023)

[1.4 Modificaciones 4](#_Toc75524024)

[Consumos 5](#_Toc75524025)

[2.1 Altas 5](#_Toc75524026)

[2.2 Listados 5](#_Toc75524027)

[2.3 Búsquedas 5](#_Toc75524028)

[2.4 Modificaciones 5](#_Toc75524029)

[2.5 Carga de consumos ramdom 6](#_Toc75524030)

[Proyecto “TP-Final”: “MANUAL TECNICO” 7](#_Toc75524031)

[“Clientes” 7](#_Toc75524032)

[Funciones 7](#_Toc75524033)

[Clasificación de las funciones utilizadas según la tarea a realizar: 7](#_Toc75524034)

[Validación 7](#_Toc75524035)

[Búsqueda de Registro 7](#_Toc75524036)

[Muestra de Registro 7](#_Toc75524037)

[Modifica Registro 7](#_Toc75524038)

[“Consumos” 7](#_Toc75524039)

[Funciones 7](#_Toc75524040)

[Clasificación de las funciones utilizadas según la tarea a realizar: 7](#_Toc75524041)

[Validación 7](#_Toc75524042)

[Búsqueda de Registro 8](#_Toc75524043)

[Muestra de Registro 8](#_Toc75524044)

[Modifica Registro 8](#_Toc75524045)

[¡Problemas……… SUPERADOS! 9](#_Toc75524046)

[Función “modifica campo” 9](#_Toc75524047)

[Función “valida e-mail” 9](#_Toc75524048)

[Función “valida Id Cliente” 9](#_Toc75524049)

[Función “validación de fecha” 9](#_Toc75524050)

[Función “carga random” 9](#_Toc75524051)

[Esquemas de algunas funciones desarrolladas para este proyecto 10](#_Toc75524052)

[Ámbito “Clientes” 10](#_Toc75524053)

[Ámbito “Consumos” 13](#_Toc75524054)

# 

# Bienvenido al Sistema!!!! “Manual de Usuario”

Aquí nos encontramos con la primera interfase del proyecto, en el cual nos encontramos con las dos primeras opciones para elegir entre el “ámbito cliente” y el “ámbito consumos”, según la necesidad del usuario. Lo único que se debe hacer es optar por la opción 1 o 2 presionando la tecla correspondiente.

## Clientes

En esta interfase el usuario se encontrará con cuatro opciones (tipo de acciones):

1. **Altas**
2. **Listados**
3. **Búsquedas**
4. **Modificaciones**

### 1.1 Altas

Una vez seleccionado este ítem se iniciará el proceso de carga de datos(información) del cliente, que serán almacenados el campo correspondiente.

nroCliente: 1

nombre: Juan

apellido: Pérez

email: jper@mail.com

domicilio: calle 23

móvil: 223465789

estado:0

0=activo 1=baja

### 1.2 Listados

#### 1.2.1. Lista Clientes Activos

Una vez seleccionada esta opción, el sistema muestra un listado de los clientes activos.

#### 1.2.2. Lista Clientes Inactivos

Una vez seleccionada esta opción, el sistema muestra un listado de los clientes inactivos.

#### 1.2.3. Lista Total de Clientes

Hecha esta selección, el sistema muestra el listado de la base de datos almacenada.

### 1.3 Búsquedas

#### 1.3.1. Búsqueda de Cliente ingresando número de Cliente

El sistema requiere del ingreso del número de cliente que se desea buscar para realizar la tarea de búsqueda.

#### 1.3.2. Búsqueda de Cliente ingresando número de D.N.I.

El sistema requiere del ingreso del número de DNI que se desea buscar para realizar la tarea de búsqueda.

#### 1.3.3. Búsqueda de Cliente ingresando número de Móvil

El sistema requiere del ingreso del número de móvil que se desea buscar para realizar la tarea de búsqueda.

#### 1.3.4. Búsqueda de Cliente ingresando la dirección de e-mail

El sistema requiere del ingreso de la dirección de e-mail que se desea buscar para realizar la tarea de búsqueda.

### 1.4 Modificaciones

#### 1.4.1 Modificación de dirección de e-mail

El sistema requiere del ingreso de la dirección de e-mail que se desea cambiar para realizar la tarea de modificación.

#### 1.4.2 Modificación del número de móvil

El sistema requiere del ingreso del número de móvil que se desea cambiar para realizar la tarea de modificación.

#### 1.4.3 Modificación del número de D.N.I.

El sistema requiere del ingreso del número de DNI que se desea cambiar para realizar la tarea de modificación.

#### 1.4.4 Modificación del estado (activo=0/baja=1)

El sistema requiere del ingreso del **número**, que corresponde a los dos estados disponibles, que se desea cambiar para realizar la tarea de modificación del estado de cliente.

## 

# Consumos

En esta interfase el usuario se encontrará con cinco opciones (tipo de acciones):

1. **Altas**
2. **Listados**
3. **Búsquedas**
4. **Modificaciones**
5. **Carga de Consumos Random**

### 2.1 Altas

Una vez seleccionado este ítem se iniciará el proceso de carga de datos(consumos) del cliente, que serán almacenados el campo correspondiente.

id Cliente: 1

año: 2021

mes:12

día: 15

datos consumidos: 369

estado:0

* 0=activo 1=baja

### 2.2 Listados

#### 2.2.1. Lista Consumos en estado Activo

Una vez seleccionada esta opción, el sistema muestra un listado de los consumos activos.

#### 2.2.2. Lista Consumos en estado Inactivos

Una vez seleccionada esta opción, el sistema muestra un listado de los consumos inactivos.

#### 2.2.3. Lista Total de Consumos

Hecha esta selección, el sistema muestra el listado de la base de datos almacenada.

### 2.3 Búsquedas

#### 2.3.1. Búsqueda de Consumos ingresando fecha

El sistema requiere del ingreso del ingreso de la fecha que se desea buscar (los consumos) para realizar la tarea de búsqueda.

#### 2.3.2. Búsqueda de Consumos por cliente

El sistema requiere del ingreso del número de id Cliente que se desea buscar (los consumos) para realizar la tarea de búsqueda.

### 2.4 Modificaciones

#### 2.4.1 Modificación del estado de consumo

El sistema requiere del ingreso del id de consumo para encontrar, si existe, el registro de consumo al que se le quiere modificar el estado. El sistema, antes de ejecutar dicha acción pregunta al usuario “Seguro desea modificar???”, donde el mismo debe optar por oprimir la tecla que desea (1 o 2).

#### 2.4.2 Modificación fecha de consumo

El sistema requiere del ingreso del id de consumo para encontrar, si existe, el registro de consumo al que se le quiere modificar la fecha de consumo. El sistema, antes de ejecutar dicha acción pide al usuario la fecha (aa,mm,dd) para realizar la modificación deseada.

#### 2.4.3 Modificación de un consumo del cliente.

El sistema requiere del ingreso del número de DNI que se desea cambiar para realizar la tarea de modificación.

#### 2.4.4 Modificación del consumo por Id

El sistema requiere del ingreso del número de id, que corresponde a los diferentes consumos realizado por el cliente en cuestión. De esta manera, el sistema solicita al usuario el ingreso de la nueva carga de consumos para el cliente, siendo esta acción confirmada por el sistema.

### 2.5 Carga de consumos ramdom

# La función carga datos aleatorios a la estructura de consumo sin superar la cantidad de registros existente en el archivo de clientes. Esta función valida la fecha respecto a la actual y a la cantidad de días correspondientes a cada mes. Y por último comprueba que si la fecha cargada es la misma a que contiene un registro del archivo correspondiente a un mismo cliente, esta suma los datos consumidos; de caso contrario , crea un registro nuevo.

# Proyecto “TP-Final”: “MANUAL TECNICO”

*Este proyecto gira en torno a dos estructuras fundamentales “Clientes” y “Consumos”, de esa forma se realizará la explicación de cada una de estas partes.*

## “Clientes”

## Funciones

# Clasificación de las funciones utilizadas según la tarea a realizar:

### Validación

1. Validación de e-mail (en la carga de datos y la búsqueda.
2. Validación de numero de cliente.

Las funciones que validan e-mail son prácticamente idénticas, pero difieren entre si en el “tipo” de parámetro que reciben, para la carga de información en el “alta” de un cliente se recibe del tipo “estructura” a diferencia de la usada para realizar una búsqueda a través del e-mail que recibe del tipo “int”.

### Búsqueda de Registro

*Búsqueda de un registro a través de*:

1. Numero de cliente.
2. Numero de móvil.
3. Número de dni.
4. Dirección de e-mail.

La función que busca un registro a través del numero de cliente es la única que sufre una variante ya que recibe un parámetro del tipo “int”, por esa razón a la hora de realizar la búsqueda, la comparación es ejecutada con otro tipo de criterio.

### Muestra de Registro

*Muestra en pantalla información hacer de los registros del archivo*

1. Muestra un registro(cliente).
2. Muestra lista de registros (cliente).
3. Cantidad de registros en el archivo.
4. Muestra listado de registros según su estado (activo/baja).

Estas funciones son utilizadas para mostrar un registro en particular, un listado de clientes, la cantidad existente, y también según su estado. Aquí encontramos, también, variante según el tipo de parámetro (char/int)

### Modifica Registro

1. Modifica campo de e-mail.
2. Modifica campo de numero de móvil.
3. Modifica campo de estado.

Estas funciones modifican la información contenida en los campos, ya mencionados, del registro. Al igual que las funciones que realizan búsquedas, aquí encontramos dos variantes según el tipo de parámetro ingresado (char/int).

## “Consumos”

## Funciones

## Clasificación de las funciones utilizadas según la tarea a realizar:

### Validación

1. Validación de Id Cliente (en la carga de datos).
2. Validación de fecha (día, mes y año).

La validación de fecha modulariza con las herramientas que nos proporciona la librería “TIME”.

La validación de “id cliente”, solo recibe un único parámetro del tipo “int” ya que solo es utilizada por otra función principal como “altaconsumo”.

### Búsqueda de Registro

*Búsqueda de un registro a través de*:

1. Por fecha de consumo.
2. De consumo por Id Cliente.
3. De consumo por Id.

Estas funciones son utilizadas para buscar un registro en particular, por fecha en particular, por consumo, y por el ID. Aquí encontramos variante según el tipo de parámetro que recibe (char/int)

### Muestra de Registro

*Muestra en pantalla información hacer de los registros del archivo*

1. Muestra un registro(consumo).
2. Muestra lista de registros (consumos).
3. Muestra listado de registros (consumos) según su estado (activo/baja).

Estas funciones son utilizadas para mostrar un registro(consumo) en particular, un listado de registros (consumos), y también según su estado. Aquí encontramos, también, variante según el tipo de parámetro (int) que se debe ingresar para mostrar listado según el estado.

### Modifica Registro

1. Modifica campo de consumo de un cliente.
2. Modifica campo de estado de consumo.
3. Modifica campo de fecha de consumo.

Estas funciones modifican la información contenida en los campos, ya mencionados, del registro. A la diferencia de las funciones que realizan modificaciones en el ámbito “cliente”, aquí solo necesitamos recibir el segundo parámetro del tipo int.

# ¡Problemas……… SUPERADOS!

### Función “modifica campo”

Durante la primera parte del desarrollo de esta función encontramos algunos inconvenientes ya que logramos hacerla funcionar, pero de forma parcial, se lograba con éxito la modificación del campo elegido, pero cuando iba a corroborar la acción realizada nos encontrábamos con que se había modificado el campo, pero no del registro deseado sino del último del archivo.

La primera versión de esta función, dependía de otra que realizara la búsqueda del registro a modificar, esta relación generaba un aparente conflicto que involucraba a el movimiento y/o ubicación del cursor.

Con ayuda de la clase 31 del profe Sergio y de un video de “YouTics” logramos encontrar la falla y realizamos la modificación pertinente, que consistía en agregar un 3er parámetro (dato que apunta al campo del registro a modificar) que serviría para encontrar el campo deseado y así poder sobrescribirlo con el nuevo contenido, todo, desde la misma función.

### Función “valida e-mail”

Para la validación de e-mail, se recurrió a Google para investigar acerca de este tipo de validación y se encontró una forma de identificar si la cadena de caracteres ingresada contiene el símbolo @(arroba), y de esa forma valida lo ingresado por el usuario.

### Función “valida Id Cliente”

Función “compruebaIdcliente” valida si existe el Id cliente en el archivo.

### Función “validación de fecha”

Función “anioActual”, “mesActual”, “diaActual”, utilizadas por “cargaAño”,” cargaMes” y” cargaDia”.

Encontramos interesante profundizar en lo que respecta “validación de fecha” a la hora de la carga/alta en consumos y para la “cargaRandom “, volviendo a recurrir a fuentes externas encontramos la librería “TIME”, en ella encontramos herramientas muy útiles para resolver, o, mejor dicho, potenciar la validación de la fecha.

### Función “carga random”

Esta función modulariza con la función “cuentaRegistro”, con las herramientas de la librería TIME, con “cargaDatosConsumidos y con “cargaNuevoConsumo”.

Esta función, si bien fue la que más inconvenientes nos trajo; nos resultó muy útil para poder entender mejor el movimiento que se hace dentro de un archivo. No podíamos resolver que sobrescriba el archivo existente, o que cree uno nuevo respetando el id autoincremental .

# Esquemas de algunas funciones desarrolladas para este proyecto

### Ámbito “Clientes”

#### /// VALIDA QUE EL MAIL INGRESADO CONTENGA @

int validaEmail(stClientes cliente) {

int flag=0;

char caracter = '@';

char caracterEnCadena[] = {caracter, '\0'};

if (strstr(cliente.email, caracterEnCadena) != NULL) {

flag=1;

}

return flag;

}

#### /// VALIDA QUE EL MAIL INGRESADO CONTENGA @ (recibe por parámetro un dato del tipo char)

int validaEmailChar(char eMailNew []) {

int flag=0;

char caracter = '@';

char caracterEnCadena[] = {caracter, '\0'};

if (strstr(eMailNew, caracterEnCadena) != NULL) {

flag=1;

}

return flag;

}

#### /// VALIDA SI NUM DE CLIENTE YA EXISTE

int compruebaIdCliente(int idCliente) {

int flag=0;

stClientes c;

stConsumos con;

FILE \*archi= fopen("clientes.dat", "rb");

if (archi) {

while(fread(&c, sizeof(stClientes), 1, archi)>0 && flag==0) {

if(c.id==idCliente) {

con.idCliente=idCliente;

flag=1;

if(c.estado==1){

flag=0;

}

}

}

fclose(archi);

}

return flag;

}

#### ///BUSCA REGISTRO POR NRO DE CLIENTE

void ubicaRegistroNroCliente (char archivo [], int nroCliente) {

int flag=0;

stClientes cliente;

FILE \*pArchi = fopen(archivo, "rb");

if(pArchi) {

while(fread(&cliente, sizeof(stClientes), 1, pArchi)>0) {

if(cliente.nroCliente == nroCliente) {

muestraUnCliente(cliente);

flag=1;

}

}

fclose(pArchi);

} else {

printf("/n ERROR DE APERTURA DE ARCHIVO ");

}

if(flag==0) {

printf("\nNo existe");

}

}

#### ///BUSCA REGISTRO POR EMAIL

void ubicaRegistroEmail (char archivo [], char email []) {

int flag=0;

stClientes cliente;

FILE \*pArchi = fopen(archivo, "rb");

if(pArchi) {

while(fread(&cliente, sizeof(stClientes), 1, pArchi)>0) {

if(strcmp(cliente.email,email)==0) {

muestraUnCliente(cliente);

flag=1;

}

}

fclose(pArchi);

} else {

printf("/n ERROR DE APERTURA DE ARCHIVO ");

}

if(flag==0) {

printf("\nNo existe");

}}

#### ///MODIFICA CAMPO EMAIL

void modificaEmail(char archivo[], char eMailNew [], char eMailActual []) {

stClientes cliente;

int flag=0;

int pos=0;

FILE \*pArchi = fopen(archivo, "r+b");

if(pArchi) {

while(flag==0 && fread(&cliente, sizeof(stClientes), 1, pArchi)>0) {

if(strcmp(cliente.email, eMailActual)==0) {

strcpy(cliente.email, eMailNew);

pos=ftell(pArchi)-sizeof(stClientes);

fseek(pArchi,pos,SEEK\_SET);

fwrite(&cliente, sizeof(stClientes), 1, pArchi);

flag=1;

}

}

}

fclose(pArchi);

}

### 

### Ámbito “Consumos”

#### ///VALIDACION DE ID CLIENTE

int compruebaIdCliente(int idCliente) {

int flag=0;

stClientes c;

stConsumos con;

FILE \*archi= fopen("clientes.dat", "rb");

if (archi) {

while(fread(&c, sizeof(stClientes), 1, archi)>0 && flag==0) {

if(c.id==idCliente) {

con.idCliente=idCliente;

flag=1;

} }

fclose(archi);

}

return flag;

}

#### ///VALIDACION DE FECHA

int anioActual() {

time\_t ahora = time(NULL);

struct tm\* ahoralocal = localtime(&ahora);

return ahoralocal->tm\_year+1900;

}

int mesActual() {

time\_t ahora = time(NULL);

struct tm\* ahoralocal = localtime(&ahora);

return ahoralocal->tm\_mon +1;

}

int diaActual() {

time\_t ahora = time(NULL);

struct tm\* ahoralocal = localtime(&ahora);

return ahoralocal->tm\_mday;

}

#### ///CARGA RANDOM

void cargaConsumoAleatorio (char archivo []) {

int cliente=cuentaRegistros("clientes.dat", sizeof(stClientes));

int diasDelMes=0;

int id=cuentaRegistros("consumos.dat",sizeof(stConsumos));

int mes[12]= {31,28,31,30,31,30,31,31,30,31,30,31};

int anioHoy=anioActual();

int mesHoy=mesActual();

int diaHoy=diaActual();

stConsumos con;

int idCliente=1;

int flag=0;

do {

for( int i=0; i<10; i++) {

con.idCliente=rand()%(50)+1;

con.anio=anioHoy;

con.mes=rand()%(mesHoy)+1;

diasDelMes=mes[con.mes];

if (con.mes==mesHoy) {

con.dia=rand()%(diaHoy)+1;

} else {

con.dia=rand()%(diasDelMes)+1;

}

con.datosConsumidos=rand()%(1000)+100;

flag=cargaDatosConsumidos("consumos.dat", con);

con.estado=0;

if (flag==1) {

id++;

con.id=id;

cargaNuevoConsumo("consumos.dat", con);

}

}

idCliente++;

} while(idCliente<=cliente);

}