



UNIVERSIDAD NACIONAL ABIERTA
VICERRECTORADO ACADÉMICO
ÁREA: INGENIERÍA

TRABAJO PRÁCTICO

ASIGNATURA: **COMPUTACIÓN I**

CÓDIGO: **323**

FECHA DE ENTREGA AL ESTUDIANTE:

Adjunto a la Primera Prueba Parcial

FECHA DE DEVOLUCIÓN POR PARTE DEL ESTUDIANTE:

Adjunto a la Prueba Integral.

NOMBRE DEL ESTUDIANTE: JOSE RAFAEL GUTIERREZ BLANCO

CÉDULA DE IDENTIDAD: V-15.196.382.

CORREO ELECTRÓNICO DEL ESTUDIANTE: josefo727@gmail.com

TELÉFONO: 0414-4983652 / 0245-9961487.

CENTRO LOCAL: CARABOBO

CARRERA: **-126-**

LAPSO: **2014/1**

NÚMERO DE ORIGINALES: 1 TP DIGITAL EN CD, 1 TP EN FISICO.

FIRMA DEL ESTUDIANTE:

UTILICE ESTA MISMA PÁGINA COMO CARÁTULA DE SU TAREA O
TRABAJO PRÁCTICO

RESULTADOS DE CORRECCIÓN:

| OBJ. N° | | 5 | 6 | 7 | 8 |
|---------|-----|---|---|---|---|
| 0:NL | 1:L | | | | |

República Bolivariana de Venezuela
Ministerio del PP para la Educación Universitaria
Universidad Nacional Abierta
Centro Local Carabobo

Trabajo Práctico
De Computación I

Nombre: José Rafael
Apellidos: Gutierrez Blanco
C.I.: V-15.196.382
e-mail: josefo727@gmail.com
Télef.: 0414-4983652
Carrera: Lic. en Matemáticas -126-
Asignatura: Computación I -323-
Prof.: Ing. Lola Párraga

-Valencia, 20 de Julio de 2014-

Tabla de contenido

| | |
|---|----|
| Introducción | 5 |
| Planteamiento del Problema | 6 |
| Solución analítica del problema..... | 8 |
| Datos de entrada:..... | 8 |
| Datos auxiliares o intermedios: | 8 |
| Datos de salida:..... | 8 |
| Proceso: | 9 |
| Resolución del problema usando la metodología MAPS..... | 10 |
| Etapa I: El Dialogo..... | 10 |
| ¿Se hará mantenimiento a la base de datos? | 10 |
| ¿El uso de cada submenú requiere que se use en un solo orden? | 11 |
| ¿Se harán necesarias otras opciones en el menú principal? | 11 |
| Datos de Entrada:..... | 11 |
| Datos de Salida:..... | 12 |
| Etapa II: Las Especificaciones..... | 14 |
| Precondiciones de entrada: | 14 |
| Postcondiciones de entrada:..... | 15 |
| Precondiciones de salida: | 15 |
| Postcondiciones de salida:..... | 16 |
| Etapa III: La Partición..... | 18 |
| Etapa IV: Definición de Abstracciones..... | 21 |
| Etapa V: Codificación..... | 25 |
| Menú Principal | 26 |
| Etapa VI: Prueba y Verificación | 32 |
| Etapa VII: Presentación..... | 32 |
| Diseño del Algoritmo en Pseudocódigo | 32 |
| Declaración de Variables y Tipos | 32 |
| Diseño del Algoritmo..... | 36 |

| | |
|--|----|
| Prueba del Algoritmo | 75 |
| Prueba de Robustez del Algoritmo | 75 |
| Ingreso de nombre y apellido: | 75 |
| Ingreso de C.I.: | 76 |
| Ingreso de la edad | 78 |
| Ingreso de fechas | 79 |
| Codificación del Algoritmo | 82 |
| Conclusión | 83 |

Introducción

Aprender a programar es en sí un proceso que tiene un algoritmo a seguir para su feliz término. No se trata de comprender las técnicas, los métodos, los medios y las herramientas, más que eso, se trata de su correcta utilización: de saber llevar la secuencia de lo aprendido, de comprender la utilización de cada herramienta y aplicarla a las debidas y complicadas situaciones planteadas, y por último, usarlas todas en conjuntos para resolver situaciones más complejas; se debe saber que si no se sigue debidamente tal algoritmo no se puede aprender a programar.

Es por ello que en la asignatura Computación I se enseñan los métodos, los medios, las formas y las soluciones, pero también se pide la realización de un trabajo práctico mediante el cual no sólo se evaluará el dominio de los objetivos 5, 6, 7 y 8, sino que también sirve como medio de sedimentación al alumno de lo que ha aprendido, no para que cierre su ciclo de aprendizaje, sino más bien para que pase a otra etapa en la cual se espera aprender más.

Planteamiento del Problema

La Clínica “La Mejor”, requiere un sistema automatizado para registrar a los pacientes que son atendidos, de tal modo, que se puedan obtener datos específicos tales como: las estadísticas en atención a menores de edad, adultos mayores, tanto femenino como masculino, los ingresos recaudados mensualmente según los tipos de patologías atendidas, entre otros. Es importante destacar, que se han clasificado estas patologías en tres categorías: las tipo A, las B y las tipo C; y se deberán tomar en cuenta las siguientes especificaciones, al momento de ser atendidas en la Clínica:

- Los pacientes deberán cancelar Bs. 500 por ingreso a la clínica.
- Deberán pagar diariamente:
 - Bs. 550 si son atendidos por patologías tipo A.
 - Bs. 430 si se trata de las tipo B.
 - Bs. 620 si es por las tipo C.

Dicho Sistema Automatizado, debe cumplir con las siguientes pautas:

- 1) Presentar un menú de mantenimiento con las opciones siguientes:
 - a) Incluir
 - b) Consultar
 - c) Modificar
 - d) Eliminar
 - e) Reporte
 - f) Salida
- 2) Debe registrar, con la opción Incluir, los siguientes datos:
 - a) Cédula de Identidad.
 - b) Apellidos y Nombres.

- c) Edad.
 - d) Sexo.
 - e) Días de hospitalización.
 - f) Tipo de patologías (A/B/C).
- 3) Debe ordenar el registro de pacientes por el número de la cédula de identidad.
- 4) Mostrar por pantalla, cuando se consulte a un paciente por su número de cédula de identidad, toda la información referente al mismo.
- 5) Mostrar por pantalla, en la opción Modificar y en la opción Eliminar, un mensaje de confirmación, donde el usuario ratifica la acción de modificar o eliminar datos en el sistema.
- 6) Las opciones Consultar, Modificar y Eliminar, harán uso de los mismos datos registrados en la opción Incluir.
- 7) La opción Reporte, debe generar un listado impreso que presente la siguiente información:
- a) Ingresos recaudados por patologías tipo A.
 - b) Ingresos recaudados por patologías tipo B.
 - c) Ingresos recaudados por patologías tipo C.
 - d) Número de pacientes atendidos menores de 18 años.
 - e) Número de pacientes atendidos mayores de 65 años sexo femenino.
 - f) Número de pacientes atendidos mayores de 65 años sexo masculino.
- 8) En base a lo planteado, realice lo siguiente:
- a) Resuelva el problema algorítmicamente, usando la metodología MAPS.

- b) Diseñe un algoritmo en pseudocódigo donde se muestre el uso correcto de las técnicas de programación estructurada.
- c) Pruebe el algoritmo diseñado usando el concepto de robustez.
- d) Codifique el algoritmo en lenguaje de programación PASCAL.

Solución analítica del problema

Datos de entrada:

Nombre y apellido del paciente, cédula de identidad, edad, sexo, patología, fecha de ingreso y de alta o egreso.

Datos auxiliares o intermedios:

Días hospitalizado, importe diario, importe total, categoría según edad.

Datos de salida:

Nombre y apellido del paciente, cédula de identidad, edad, sexo, patología, fecha de ingreso y de alta o egreso, importe diario, importe total, días hospitalizado, monto total. Lista del registro de pacientes ordenado por su número de cédula. Ingresos recaudados por patologías tipo A, ingresos recaudados por patologías tipo B, ingresos recaudados por patologías tipo C, número de pacientes atendidos menores de 18 años, número de pacientes atendidos mayores de 65 años sexo femenino, número de pacientes atendidos mayores de 65 años sexo masculino.

Proceso:

Al integrar los datos de un paciente al momento de su ingreso a la clínica, estos se guardarán en un record o registro contentivo de doce campos de variables de diferentes categorías en un archivo base de datos, lo que hará que estén disponibles dichos registros después de cerrar y abrir el programa. Estando hospitalizado el paciente debe ser posible consultarlo por su número de cédula de identidad, permitiendo verificar su estatus, días hospitalizado, monto acumulado, patología, entre otros. Al dar de alta al paciente se registrará en la base de datos, no sin antes calcular los montos totales definitivos y mostrarlo en pantalla, lo que permitirá al operador obtener estos montos y datos de manera automática.

El sistema debe permitir la modificación de los datos de un paciente hospitalizado o de alta así como su eliminación. Este mantenimiento se deberá realizar siempre que haya posibles errores en el ingreso o alta de paciente, de la misma manera que en cualquiera de los datos aportados por el operador. Así mismo deberá ordenar los registros de los pacientes por su número de cédula de identidad de manera ascendente y mostrarlo, al igual que generar el reporte estadístico especificado por la administración de la clínica. Para ello:

- ✓ Todas las entradas deben ser validadas antes de ingresar al registro según la naturaleza que le corresponda.
- ✓ Para modificar se ingresa por teclado en número de C.I., se realiza una búsqueda en el registro, se muestran los datos encontrados y luego se pregunta cuál campo quiere modificar. Si no existe el registro se mostrará en pantalla.
- ✓ El método de búsqueda del paciente por su C.I. en el registro consiste en un recorrido total del mismo hasta conseguir un valor coincidente, en caso contrario indicar con un mensaje que no se ha encontrado.
- ✓ Tanto al ingresar como al modificar los datos de un paciente deben validarse los datos "ingresados" uno a uno para mantener la consistencia de los mismos.
- ✓ Para eliminar el registro de un paciente se lee por teclado la C.I., se realiza una búsqueda y si no existen los datos del paciente en el registro se notifica por mensaje, si existe, se pide confirmación de eliminación y luego se procederá a eliminar el registro del paciente.
- ✓ Al modificar los datos de un paciente ya dado de alta, que estén relacionados con el importe total pagado, como lo es el caso de las fechas de ingreso o egreso (días hospitalizado) y/o patología, deberá recalcularse el monto y

reescribirse en el registro, para así no generar inconsistencias en la consulta de datos y en el reporte estadísticos.

- ✓ El registro de paciente se ordenará realizando comparaciones binomiales de los números de cédula de identidad de cada paciente hasta recorrerlos todos y lograr un nuevo registro ordenado.
- ✓ El reporte estadístico se logrará haciendo un recorrido por cada registro de pacientes, formando agrupaciones en conjuntos según las características o especificaciones planteadas.

Resolución del problema usando la metodología MAPS.

(Unidad 5)

Etapas I: El Dialogo.

El Problema consiste en elaborar un software de control para la clínica “La Mejor” que permita el registro de todos los pacientes que ingresan a dicho centro de salud; mediante entrada de datos por el teclado el software debe registrar del paciente, la C.I., el nombre, el apellido, el sexo, la edad, tipo de patología o enfermedad por la que fue ingresado, fecha de ingreso, fecha de egreso o de alta, y a partir de esos datos, calcular los días hospitalizado, clasificación según su edad, es decir, si es menor de edad, adulto o adulto mayor y, el monto a pagar; además mostrarlo en pantalla.

¿Se hará mantenimiento a la base de datos?

Sí, mediante el software se realizarán ciertos mantenimientos a la base de datos, que incluyen: modificar la información o datos de un paciente, eliminar la información o datos de un paciente y dar de alta a un paciente hospitalizado; así como mostrar en pantalla al consultar la información de un paciente ingresando su número de cédula, realizar un reporte mensual con ciertos datos estadísticos y mostrarlo, y ordenar el registro de los pacientes de manera ascendente por su número de cédula.

¿El uso de cada submenú requiere que se use en un solo orden?

No, el sistema debe permitir que terminado el registro de un paciente pueda procederse inmediatamente con la inclusión de otro, incluso sin haber dado de alta al primero, de la misma manera debe permitir el uso de uno y otro submenú de manera indistinta para realizar registros, consultas, reportes o modificaciones a la base de datos sin que se alteren los datos de los demás pacientes registrados y sin necesidad que se siga el orden en que aparecen en el menú general o principal.

¿Se harán necesarias otras opciones en el menú principal?

Para opciones de registro y actualización de los datos de cada paciente se pide un menú mantenimiento con las opciones siguientes: **Incluir, Consultar, Modificar, Eliminar, Reporte y Salida**, sin embargo, y debido a las especificaciones del problema planteado, se hará a dicho menú la integración de otras opciones, quedando de ésta manera: **Incluir, Consultar, Modificar, Eliminar, Alta, Ordenar Registro, Reporte Mensual y Salida**.

Datos de Entrada:

1. Al ingresar un paciente:
 - a. Nombre del paciente.
 - b. Apellido del paciente.
 - c. Cédula de identidad del paciente.
 - d. Edad.
 - e. Sexo.
 - f. Patología.
 - g. Fecha de Ingreso.
2. Al dar de alta un paciente:
 - a. Fecha de alta o egreso.
3. Al consultar los datos de un paciente:
 - a. Cédula de identidad del paciente.

4. Para modificar los datos de un paciente:
 - a. Cédula de identidad del paciente.
5. Para eliminar el registro de un paciente:
 - a. Cédula de identidad del paciente.
6. Para dar de alta a un paciente:
 - a. Cédula de identidad del paciente.
7. Al pedir reporte estadístico de un mes:
 - a. Ingresar el número del mes a consultar.

Datos de Salida:

1. Al dar de alta a un paciente:
 - a. Fecha de ingreso.
 - b. Fecha de alta o egreso.
 - c. Patología.
 - d. Días hospitalizado.
 - e. Monto total.
2. Al consultar un paciente hospitalizado:
 - a. Nombre del paciente.
 - b. Apellido del paciente.
 - c. Cédula de identidad del paciente.
 - d. Edad.
 - e. Sexo.
 - f. Patología.
 - g. Clasificación según edad.
 - h. Fecha de Ingreso.
 - i. Fecha actual.
 - j. Días hospitalizado.

- k. Importe de ingreso.
 - l. Importe diario.
 - m. Importe total actual (lo que deberá pagar si es dado de alta).
3. Al consultar un paciente dado de alta:
- a. Nombre del paciente.
 - b. Apellido del paciente.
 - c. Cédula de identidad del paciente.
 - d. Edad.
 - e. Sexo.
 - f. Patología.
 - g. Clasificación según edad.
 - h. Fecha de ingreso.
 - i. Fecha de egreso.
 - j. Días hospitalizado.
 - k. Importe de ingreso.
 - l. Importe diario.
 - m. Importe total.
4. Al pedir ordenar el registro:
- a. La lista del registro de pacientes ordenado por su número de cédula, seguido de su nombre y apellido.
5. Al pedir el reporte de un mes:
- a. Ingresos recaudados por patologías tipo A.
 - b. Ingresos recaudados por patologías tipo B.
 - c. Ingresos recaudados por patologías tipo C.
 - d. Número de pacientes atendidos menores de 18 años.
 - e. Número de pacientes atendidos mayores de 65 años sexo femenino.
 - f. Número de pacientes atendidos mayores de 65 años sexo masculino.

Etapa II: Las Especificaciones.

Aun cuando se cuenta con array's (arreglos) como posible medio de registro de pacientes para realizar dicho sistema, se ha elegido usar record debido a que la principal limitación de un array es que todos los datos que contiene deben ser del mismo tipo. En el presente problema nos interesa agrupar datos de distinta naturaleza, como pueden ser el nombre y la edad de un paciente, que serían del tipo string e integer respectivamente. Entonces se empleará el record o registro, que se definen indicando el nombre y el tipo de cada campo (cada dato que guardamos en el registro), y se accederá a estos campos indicando el nombre de la variable y el del campo separados por un punto.

Precondiciones de entrada:

| Entradas (Datos del paciente) | Criterio de Validez |
|-------------------------------|--|
| Nombre | No debe admitir números ni caracteres o símbolos especiales, sólo espacios y caracteres alfabéticos (letras). |
| Apellido | No debe admitir números ni caracteres o símbolos especiales, sólo espacios y caracteres alfabéticos (letras). |
| Cédula de identidad | No debe admitir letras ni caracteres o símbolos especiales ni espacios, sólo números enteros. |
| Edad | No debe admitir letras ni caracteres o símbolos especiales ni espacios, sólo números enteros: $(0 \leq \text{Edad} \leq 120)$ |
| Sexo | Sólo debe admitir un carácter, de dos caracteres: F[emenino] o M[asculino] |
| Patología | Sólo debe admitir un carácter, de tres caracteres: A, B o C. |
| Fecha de Ingreso/egreso | No puede ser letras, se admite sólo el formato dd/mm/aaaa, pudiéndose ingresar d/m/aaaa para días y meses de un solo dígito. $(1 \leq d \leq 31 \text{ y } 1 \leq m \leq 12)$, la fecha debe introducirse completa incluyendo la barra inclinada o slash "/" como separador entre día y mes, y mes y año. |

Postcondiciones de entrada:

Al haber ingresado al paciente, siempre y cuando no se haya dado de alta, el paciente estará en status de hospitalizado, al ser dado de alta cambiará su status, por lo que aquí entra en juego una variable booleana para saber si el paciente está activo (hospitalizado) o inactivo (dado de alta).

El record puede ser representado de la manera siguiente:

$$\{ (\{ \text{status}_i \} : \text{booleano}) ; (\{ \text{nombre}_i, \text{apellido}_i \} : \text{secuencia de caracteres}) ; (\{ \text{cédula}_i \} : \text{entero largo}) ; (\{ \text{edad}_i \} : \text{entero}) ; \{ \text{sexo}_i, \text{patología}_i \} : \text{carácter}) ; \{ \text{ingreso}_i, \text{total}_i \} : \text{real}) ; \{ \text{fecha_ingreso}_i, \text{fecha_egreso}_i \} : \text{secuencia de caracteres}) \} \forall i \in \mathbb{N}$$

Precondiciones de salida:

Para la salida de datos a pantalla, existen ciertas condiciones que se dan en el acceso a los submenús, por ejemplo, si no hay registro alguno, los submenús “Consultar”, “Modificar”, “Eliminar”, “Alta”, “Ordenar Registro” y “Reporte Mensual” (todos con datos de salida) no tendrán nada que mostrar. Para ingresar a los submenús “Consultar”, “Modificar” y “Eliminar”, debe ingresarse el número de cédula de un paciente registrado, sin importar si está o no hospitalizado. Para ingresar al submenú “Alta”, debe ingresarse el número de cédula de un paciente registrado que esté hospitalizado. La única condición para ingresar al submenú “Ordenar Registro” es que haya registro de al menos un paciente; y por último, para ingresar al submenú “Reporte Mensual”, debe existir el registro de al menos un paciente y ser ingresado el número del mes del que se requiera el reporte.

| Precondiciones de Salida | | | | |
|--------------------------|-----------|--------------------|-------------------|----------|
| Precondición 1 | Submenú | Precondición 2 | Precondición 3 | Acceso |
| Registro = Vacío | Consultar | Ninguna | Ninguna | Denegado |
| Registro \neq Vacío | Consultar | C.I. Registrada | Ninguna | Aceptado |
| Registro \neq Vacío | Consultar | C.I. No registrada | Ninguna | Denegado |
| Registro = Vacío | Modificar | Ninguna | Ninguna | Denegado |
| Registro \neq Vacío | Modificar | C.I. Registrada | Ninguna | Aceptado |
| Registro \neq Vacío | Modificar | C.I. No registrada | Ninguna | Denegado |
| Registro = Vacío | Eliminar | Ninguna | Ninguna | Denegado |
| Registro \neq Vacío | Eliminar | C.I. Registrada | Ninguna | Aceptado |
| Registro \neq Vacío | Eliminar | C.I. No registrada | Ninguna | Denegado |
| Registro = Vacío | Alta | Ninguna | Ninguna | Denegado |
| Registro \neq Vacío | Alta | C.I. Registrada | Status = Activo | Aceptado |
| Registro \neq Vacío | Alta | C.I. Registrada | Status = Inactivo | Denegado |
| Registro \neq Vacío | Alta | C.I. No registrada | Ninguna | Denegado |
| Registro = Vacío | Ordenar | Ninguna | Ninguna | Denegado |
| Registro \neq Vacío | Ordenar | Ninguna | Ninguna | Aceptado |
| Registro = Vacío | Reporte | Ninguna | Ninguna | Denegado |
| Registro \neq Vacío | Reporte | Nro. de Mes | Ninguna | Aceptado |

Postcondiciones de salida:

Del record (paciente) los únicos registros que pudiesen estar vacíos son el de la fecha de egreso o alta del paciente y del total, y esto es siempre que el status sea hospitalizado, para dar de alta al paciente se llenará obligatoriamente como condición sine qua non la fecha para terminar el proceso de salida o alta del paciente en el registro, e inmediata y automáticamente será calculado el monto total a pagar y será guardado en la variable “total”.

Si ya ha sido guardado el primer registro, y aun cuando el paciente registrado haya sido dado de alta, se podrá hacer uso de los submenús “Consultar”, “Modificar”, “Alta”, “Eliminar” e incluso “Ordenar Registro” y “Reporte Mensual”.

- a) Consultar: Nos lleva a un submenú que nos muestra del paciente: el estado (si está hospitalizado o ya fue dado de alta), el nombre, el apellido, la cédula de identidad, la edad, el sexo, la patología por la que fue ingresado, la clasificación según edad, la fecha de su ingreso, la fecha actual si está hospitalizado y los días hospitalizados hasta la fecha actual, sino, la fecha de alta y los días que estuvo hospitalizado, el importe de ingreso, el importe diario, y el importe total actual (lo que deberá pagar si es dado de alta en ese momento), o el importe total si ya ha sido dado de alta.
- b) Modificar: Es un submenú de entrada/salida, pues al ingresar a él, nos muestra una consulta del nombre, el apellido, la cédula de identidad, la edad, el sexo, la patología por la que fue ingresado, la fecha de su ingreso, la fecha actual si está hospitalizado, sino, la fecha de alta. Y mostrada esa pequeña consulta, permitirá modificar el/los dato/s específico/s que se seleccione/n. Al guardar los cambios, muestra una consulta completa donde se reflejan las modificaciones realizadas.
- c) Eliminar: Aunque es un submenú de mantenimiento, los es también de salida, pues antes de la eliminación del registro de un paciente, llama una consulta del mismo para que sea verificado si son los datos exactos del paciente cuyo registro se desea eliminar.
- d) Alta: Es un submenú de entrada/salida, primero muestra el nombre y apellido del paciente que se desea dar de alta (salida), luego pide ingresar la fecha de alta (entrada), para terminar con una mini consulta (salida) que muestra la fecha de ingreso, fecha de alta o egreso, días hospitalizado, patología y monto total.

- e) Ordenar Registro: Es un submenú de mantenimiento/salida, pues ordena el registro en orden ascendente por el número de cédula de los pacientes y luego muestra dicho registro siguiendo el formato: C.I. Nombre Apellido.
- f) Reporte Mensual: Nos muestra un reporte estadístico mensual de los Ingresos recaudados por patologías tipo A, Ingresos recaudados por patologías tipo B, Ingresos recaudados por patologías tipo C, Número de pacientes atendidos menores de 18 años, Número de pacientes atendidos mayores de 65 años sexo femenino, Número de pacientes atendidos mayores de 65 años sexo masculino.

Etapa III: La Partición.

El problema planteado puede dividirse en 6 rutinas principales que se describen a continuación:

1. Incluir
2. Consultar
3. Modificar
4. Eliminar
5. Alta
6. Ordenar registro
7. Reporte Mensual

Cuyas rutinas pueden a su vez, seccionarse o dividirse en las siguientes subrutinas:

1. Incluir
 - 1.1. Leer nombre del paciente
 - 1.2. Leer apellido del paciente

- 1.3. Leer número de cédula del paciente
- 1.4. Leer edad del paciente
- 1.5. Leer genero del paciente
- 1.6. Leer patolog ía por la que ingresa
- 1.7. Leer fecha de ingreso
2. Consultar
 - 2.1. Mostrar nombre del paciente
 - 2.2. Mostrar apellido del paciente
 - 2.3. Mostrar género del paciente
 - 2.4. Mostrar número de cédula del paciente
 - 2.5. Mostrar categor ía según edad
 - 2.6. Mostrar patolog ía por la que fue ingresado
 - 2.7. Mostrar fecha de ingreso
 - 2.8. Mostrar fecha de egreso o fecha actual
 - 2.9. Mostrar días hospitalizado
 - 2.10. Mostrar importe de ingreso
 - 2.11. Mostrar importe diario
 - 2.12. Mostrar importe total o actual.
3. Modificar
 - 3.1. Consulta
 - 3.1.1. Mostrar status del paciente.
 - 3.1.2. Mostrar nombre del paciente
 - 3.1.3. Mostrar apellido del paciente
 - 3.1.4. Mostrar número de cédula de identidad del paciente
 - 3.1.5. Mostrar edad del paciente
 - 3.1.6. Mostrar genero del paciente
 - 3.1.7. Mostrar patolog ía por la que fue ingresad

- 3.1.8. Mostrar fecha de ingreso del paciente
- 3.1.9. Mostrar fecha de alta del paciente (si ya ha sido dado de alta)
- 3.2. Modificar
 - 3.2.1. Nombre del paciente
 - 3.2.2. Apellido del paciente
 - 3.2.3. Número de cédula de identidad del paciente
 - 3.2.4. Edad del paciente
 - 3.2.5. Genero del paciente
 - 3.2.6. Patología por la que fue ingresad
 - 3.2.7. Fecha de ingreso del paciente
 - 3.2.8. Fecha de alta del paciente (si ya ha sido dado de alta)
- 3.3. Mostrar consulta (rutina 2) de datos del paciente para verificar
- 4. Eliminar
 - 4.1. Mostrar consulta (rutina 2) de datos del paciente
 - 4.2. Confirmar eliminación
 - 4.3. Eliminar registro del paciente
- 5. Alta
 - 5.1. Mostrar nombre y apellido del paciente
 - 5.2. Confirmar alta del paciente
 - 5.3. Dar de alta al paciente
- 6. Ordenar Registro
 - 6.1. Ordenar registro de los pacientes por su número de cédula
 - 6.2. Mostrar registro ordenado
- 7. Reporte mensual
 - 7.1. Mostrar reporte mensual del mes indicado
 - 7.2. Mostrar opción imprimir el reporte

Etapa IV: Definición de Abstracciones.

Para la solución informática al problema planteado se usará TP 7.0 (Turbo Pascal 7.0) y por ende, necesitaremos recurrir una y otra vez a sus códigos, comandos, y adaptarse a su estructura para poder llevar a feliz término el desarrollo del software solicitado. Existen algunas subrutinas que se usarán con frecuencia a lo largo del programa, al invocar rutinas, es por ello que se han definido como abstracciones que serán llamadas en su momento para realizar ciertas operaciones específicas.

Tal es el caso de la rutina que lee la fecha actual del sistema y la convierte al formato dd/mm/aaaa para luego guardarla en una cadena de caracteres definida como de longitud = 12 (string[12]). Para ello se definió una función llamada “diahoy” que realiza tal operación cuando es invocada.

Función diahoy : cadena de caracteres[12];

variables

mm, dd : cadena de caracteres[2];

aaaa : cadena de caracteres[4];

Inicio

leer fecha del sistema (año,mes,dia);

aaaa ← año ;

mm ← mes;

dd ← dia;

si longitud de dd = 1 entonces

insertar un ‘0’ antes de dd;

fin_si

si longitud de mm = 1 entonces

insertar un ‘0’ antes de mm;

```
fin_si  
diahoy ← dd/mm/aaaa;  
fin;
```

Para el ingreso de fecha manual se presentó otro reto, y es el de validar que el formato de entrada sea correcto y además que la fecha sea correcta. Por ejemplo, el usuario puede escribir 2/5/2014 en lugar de 02/05/2014, para ello se necesita una conversión de 2/5/2014 a 02/05/2014, pero hay otro detalle, con la fecha leída del sistema es evidente que se obtenga una fecha correcta, pero si la fecha es ingresada manualmente puede guardarse un registro de formato de fecha incorrecto o fecha incorrecta, que como consecuencia afectará los cálculos y operaciones del sistema.

Ejemplo de formato de fecha incorrecta: 22/O3/2014, tiene en el mes la letra “O” en lugar del “0” (cero), de modo que hay que validar que obligatoriamente se introduzcan enteros positivos en cada espacio de dd/mm/aaaa salvo los espacios que deben ocupar los slash ‘/’ como separadores de los datos de la fecha.

Ejemplo de fecha incorrecta: 29/02/2014, puesto que 2014 no es bisiesto, o 31/09/2014, e incluso 12/15/2013 pueden ser ejemplos de fecha de entrada manual que no son válidas.

Como la verificación de la fecha de entrada manual se realiza como subrutina en varias rutinas del sistema, se definió como abstracción y se elaboró una función cuyo algoritmo permite convertir una fecha tipo d/m/aaaa en dd/mm/aaaa, verifica que dd, mm y aaaa son números enteros positivos y además calcula si la fecha es o no válida incluso para los casos de los 29/02/aaaa, la función devuelve un valor booleano de verdadero si la fecha es correcta y falso si no lo es, luego el sistema se vale de ciertos ciclos para pedir al usuario que ingrese nuevamente la fecha hasta que ésta sea correcta.

Función fechval (fech : string[12]) : booleano; { fech = dd/mm/aaaa }

variables

error1, error2, error3, a, m, d : integer;

bisiesto : booleano;

max : byte;

aux : cadena de caracteres[2];

año : cadena de caracteres[4];

Inicio

si longitud de dd = 1 hacer

insertar un '0' antes de dd;

fin_si

si longitud de mm = 1 hacer

insertar un '0' antes de mm;

fin_si

aux ← dd de fech;

d ← aux, sino error1 = 1;

aux ← mm de fech;

m ← aux, sino error2 = 1;

año ← aaaa de fech;

a ← año, sino error3 = 1;

Si ((error1 + error2 + error3) = 0) entonces

Inicio

bisiesto ← ((a mod 400 = 0) y (a mod 4000 < > 0)) o

((a mod 4 = 0) y (a mod 100 < > 0));

En caso de m

1,3,5,7,8,10,12: hacer max ← 31;

4,6,9,11: hacer max ← 30;

2: hacer: Inicio

```

Si (bisiesto = verdadero) entonces
    max ← 29
Sino
    max ← 28;
fin_si
Fin;

Fin;
fechval ← (( m ≥ 1 ) y ( m ≤ 12 )) y (( d ≥ 1 ) and ( d ≤ max ));
Fin
Sino
    fechval ← false;
Fin_si
Fin;

```

De la misma manera se definió como abstracción la función “guardardatos” que es invocada en su momento por el procedimiento “ingresopaciente”, cuya función es guardar los datos del paciente recién ingresado al sistema si el operador confirma el ingreso de datos; esto se ha hecho debido a que el ingreso de un paciente al sistema y por ende al archivo de datos o base de datos (BD), requiere de un procedimiento especial de verificación de existencia el archivo o BD, reseteo, apertura y guardado y posterior cierre, subrutina que hace más engorrosa e inentendible al propio procedimiento “ingresopaciente”.

Otra subrutina a la que se recurre con frecuencia en el sistema es al procedimiento “consultar”, que además de ser el procedimiento por el cual se consulta los datos de un paciente, es invocado por los procedimientos “eliminar” y “modificar”, en el primero para que sean verificados los datos del paciente que se desea eliminar del sistema y así confirmar, el segundo para mostrar los datos de un

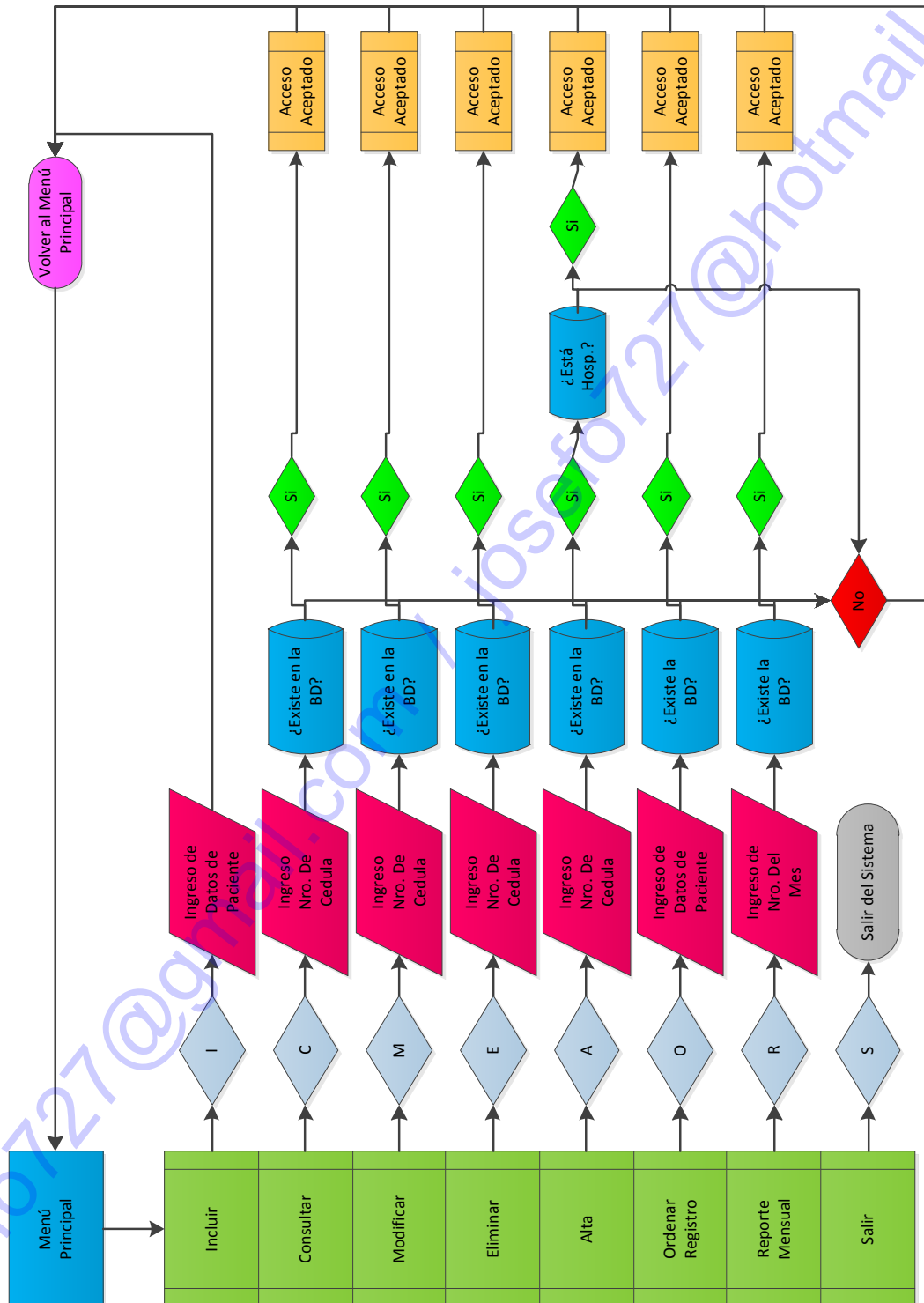
paciente inmediatamente después de hacerles modificaciones; es por ello que también se le ha definido como abstracción del sistema en construcción.

En el uso de cada rutina, subrutina, procedimientos y funciones del sistema de registro de pacientes de la Clínica La Mejor, se encuentran más subrutinas que son utilizadas y reutilizadas constantemente, como por ejemplo la validación de ingreso de nombre, apellido, edad, sexo, patología, entre otras, que se realizan con la plena intención de generar robustez al ingreso de datos y así maximizar su eficiencia y eficacia al minimizar el margen de error por registros de datos inconsistentes. Para ello se recurrió con reiteración a ciclos repetitivos como desde...hasta...hacer (for...to...do), mientras...hacer (while...do), ciclos repetitivos repetir...hasta (repeat...until); condicionales como si...entonces...sino (if...then...else), ciclos selectivos como “en caso de...hacer” (case of...do). Todos ellos conjugados e incluso algunas veces combinados adecuadamente y lógicamente estructurados definen más de una decena de abstracciones de uso recurrente a lo largo de la ejecución del sistema, de sus rutinas y sus subrutinas.

Etapas V: Codificación

En esta etapa se elabora el desarrollo del algoritmo en el lenguaje de programación a realizarse, en este caso Turbo Pascal. Sin embargo, por corresponderse al Objetivo 8 de la asignatura, y debido a que este se presentará en la codificación del archivo *.pas y su respectivo *.exe, se dejará para entonces y así no realizar redundancias innecesarias; no obstante se presentará a continuación diagramas de flujos del funcionamiento que se espera del software que servirán de enfoque del uso del sistema y además presentarán una estructura lógica a seguir, para así desarrollar claramente el Objetivo 6 (programación estructurada, técnicas de programación y programación modular).

Menú Principal



Submenú Incluir:



Sub rutinas: a.- Validar ingreso de nombre; b.- Validar ingreso de apellido; c.- Validar ingreso de nro. de cédula; d.- Verificar si la C.I. está en la BD; e.- Validar ingreso de edad; f.- Validar ingreso de género sexual; g.- Validar ingreso de patología; h.- Leer fecha actual (automática) o validar formato de la fecha ingresada.

```

graph TD
    Start([Submenú Consultar]) --> Input[Digite la C.I. del Paciente]
    Input --> Positivo{¿Entero Positivo?}
    Positivo -- No --> Exit([Regresar al Menú Principal])
    Positivo -- Sí --> BD{¿Está en la BD?}
    BD -- No --> Exit
    BD -- Sí --> Hospitalizado{¿Está Hospitalizado?}
    Hospitalizado -- No --> ShowTable1([Mostrar Consulta])
    Hospitalizado -- Sí --> ShowTable2([Mostrar Consulta])
    ShowTable1 --> Exit
    ShowTable2 --> Exit

```

Subrutinas: a.- Validar ingreso de nro. de cédula; b.- Verificar si la C.I. está en la BD; c.- Verificar si el paciente está actualmente hospitalizado. En caso de estar hospitalizado: d.- Clasificar al paciente en categoría según edad; e.- Leer fecha actual; f.- Calcular días hospitalizado; g.- Calcular importe actual. En caso de no estar hospitalizado: d.- Clasificar al paciente en categoría según edad; e.- Leer fecha de alta o egreso; f.- Calcular días hospitalizado; g.- Calcular importe total.

```

graph TD
    Start([Submenú Consultar]) --> Input[Digite la C.I. del Paciente]
    Input --> Positivo{¿Entero Positivo?}
    Positivo -- No --> End([Regresar al Menú Principal])
    Positivo -- Sí --> BD{¿Está en la BD?}
    BD -- No --> End
    BD -- Sí --> Hospitalizado{¿Está Hospitalizado?}
    Hospitalizado -- No --> MostrarConsultaNo([Mostrar Consulta])
    Hospitalizado -- Sí --> MostrarConsultaSi([Mostrar Consulta])
    MostrarConsultaNo --> End
    MostrarConsultaSi --> End
  
```

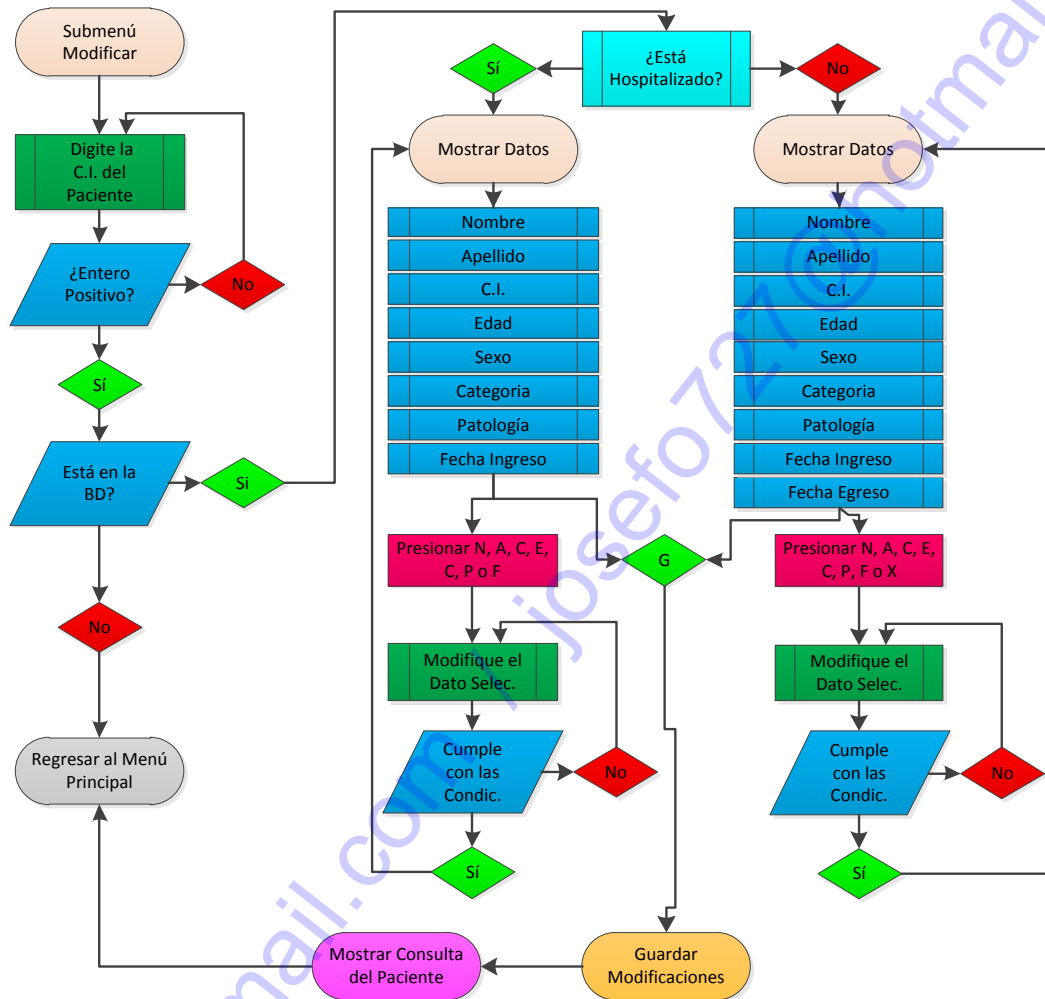
Subrutinas:

- a.- Validar ingreso de nro. de cédula;
- b.- Verificar si la C.I. está en la BD;
- c.- Verificar si el paciente está actualmente hospitalizado. En caso de estar hospitalizado:
- d.- Clasificar al paciente en categoría según edad;
- e.- Leer fecha actual;
- f.- Calcular días hospitalizado;
- g.- Calcular importe actual.

En caso de no estar hospitalizado:

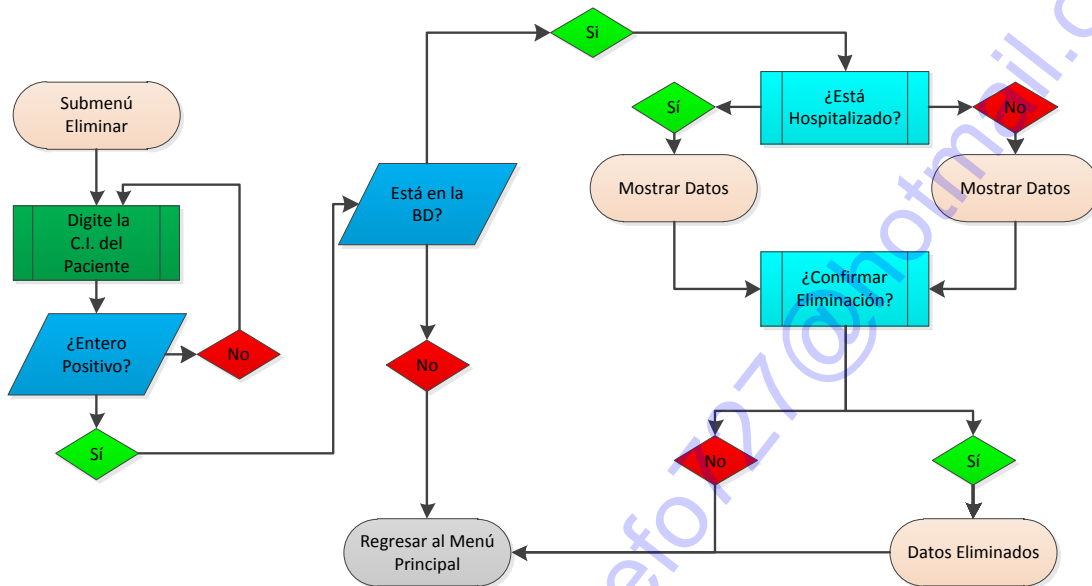
- d.- Clasificar al paciente en categoría según edad;
- e.- Leer fecha de alta o egreso;
- f.- Calcular días hospitalizado;
- g.- Calcular importe total.

Submenú Modificar:



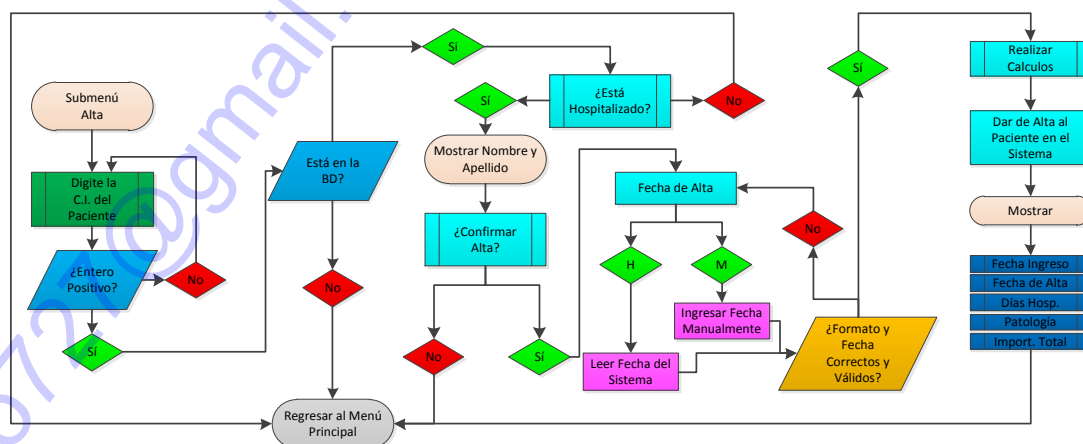
Sub rutinas: a.- Validar ingreso de nro. de cédula; b.- Verificar si la C.I. está en la BD; c.- Verificar si el paciente está actualmente hospitalizado; d.- Mostrar mini consulta que visualiza los datos a modificar; e.- Validar la entrada del dato modificado condicionado a cada variable de manera análoga a las validaciones del submenú incluir; f.- Volver a mini consulta para modificar más datos, ó g.- Guardar y salir; h.- Mostrar consulta del paciente para verificar los datos modificados; i.- Regresar al menú principal.

Submenú Eliminar:



Sub rutinas: a.- Validar ingreso de nro. de cédula; b.- Verificar si la C.I. está en la BD; c.- Verificar si el paciente está actualmente hospitalizado; d.- Mostrar consulta que visualiza los datos del paciente a eliminar; e.- Confirmar eliminación; f.- Eliminar datos o registro del paciente; h.- Regresar al menú principal.

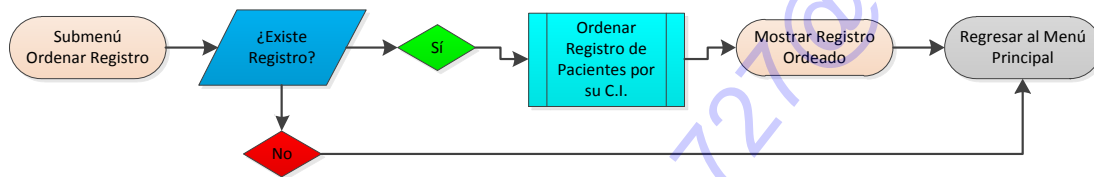
Submenú Alta:



Sub rutinas: a.- Validar ingreso de nro. de cédula; b.- Verificar si la C.I. está en la BD; c.- Verificar si el paciente está actualmente hospitalizado; d.- Mostrar

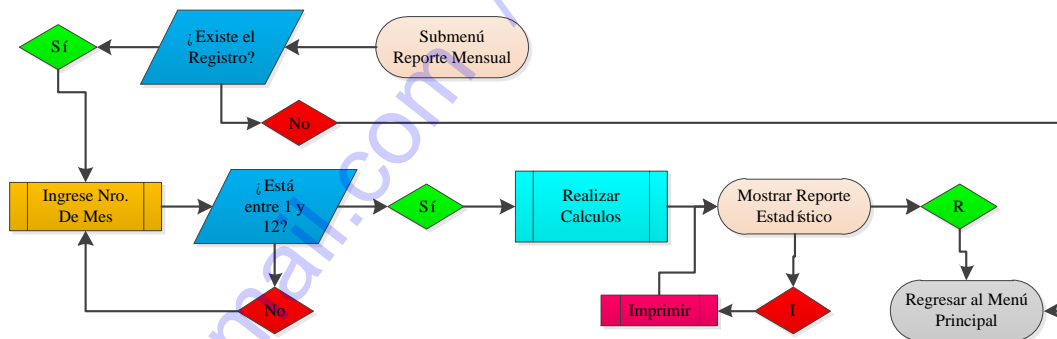
consulta que visualiza nombre y apellido del paciente; e.- Confirmar egreso o alta; f.- Seleccionar ingreso de fecha de alta; h.- Verificar validez del formato de fecha y de la fecha; i.- Realizar cálculos; j.- Dar de alta al paciente; k.- Mostrar totalización; l.- Regresar al menú principal.

Submenú Ordenar Registro:



Sub rutinas: a.- Verificar si existe el registro; b.- Ordenar el registro; c.- Mostrar el registro; d.- Regresar al menú principal.

Submenú Reporte Mensual:



Sub rutinas: a.- Verificar si existe el registro; b.- Verificar si el mes está entre el rango; c.- Realizar cálculos; d.- Mostrar reporte; e.- Imprimir reporte; f.- Regresar al menú principal.

Opción Salir:



Sub rutina: Salir del sistema.

Etapa VI: Prueba y Verificación

En esta etapa probaremos el buen funcionamiento del algoritmo implementado en Turbo Pascal ya ejecutándose el sistema: ingresando datos y verificando las salidas, corroborando que las operaciones realizadas por el sistema sean correctas y cumplen las pre y post condiciones pautadas. Aquí no solo se comprueba la robustez del sistema sino también la sencillez de su uso y la amigabilidad que hace con el usuario al arrojar los mensajes necesarios para guiarlo en el proceso de aprendizaje y de su propio uso. Estas comprobaciones se estarán analizando en el desarrollo del Objetivo 7 de éste informe.

Etapa VII: Presentación

Aquí se añadirá los comentarios necesarios al sistema para identificar la fecha, el nombre del autor, y las respectivas aclaraciones y explicaciones de las funciones de las rutinas y subrutinas del sistema. Esta etapa se incluye como complementaria de los objetivos 6 y 8.

Diseño del Algoritmo en Pseudocódigo (Unidad 6)

Declaración de Variables y Tipos

| Constantes del sistema | | |
|------------------------|-------|-------------------------------------|
| Nombre | Valor | Descripción |
| patoA | 550 | Importe diario por patología tipo A |
| patoB | 430 | Importe diario por patología tipo B |
| patoC | 620 | Importe diario por patología tipo C |

| | | |
|--------|-------------|--|
| Ingres | 500 | Importe a pagar al ingresar a la clínica |
| Archi | 'BDpac.dat' | Archivo Base de Datos |

| Tipos del sistema | | |
|-------------------|------------|--|
| Nombre | Tipo | Descripción |
| string12 | string[12] | Variable a usar en varios módulos del programa |
| Paciente | record | Registro de paciente a guardar en la BD |
| Activo | boolean | Indica si está o no hospitalizado |
| nombre, apell | string[20] | Nombre y apellido respectivamente |
| Numpac | longint | Número de cédula del paciente |
| Edad | integer | Edad del paciente |
| sexo, patolg | char | Género sexual, tipo de patología |
| total, ingreso | real | Monto total y monto por ingreso a la clínica |
| fechent, fechalt | string[12] | Fecha de entrada y salida respectivamente |

| Variables del Sistema | | |
|-----------------------|----------|--|
| Nombre | Tipo | Descripción |
| F | archivo | Archivo del registro 'paciente' |
| Dat, datos | paciente | Un record completo auxiliar de 'paciente' |
| ayo, mes, dia, sem | word | Para uso del comando getdate (fecha del sistema) |

| Función: guardardatos(dd : paciente) : boolean; | | |
|---|----------|---|
| Nombre | Tipo | Descripción |
| Dt | paciente | Un record completo auxiliar de 'paciente' |
| Faux | archivo | Archivo auxiliar copia de la BD |

| Función diahoy : string12 | | |
|---------------------------|-----------|---|
| Nombre | Tipo | Descripción |
| comm, comd | string[2] | Variables de fecha: mes y día respectivamente |
| Coma | string[4] | Variable de fecha: año |

| Función fechval (fech : string12) : boolean | | |
|---|---------|---|
| Nombre | Tipo | Descripción |
| a, m, d | integer | Fecha: año mes y día respectivamente |
| error1, error2 y error3 | integer | Errores en la conversión de copia de string a enteros |

| | | |
|----------|-----------|--|
| Bisiesto | boolean | Guarda el resultado si el año es o no bisiesto |
| Max | byte | Máximo de días en un mes |
| Dar | string[2] | Auxiliar para la conversión de días y meses |
| Ay | string[4] | Auxiliar para la conversión de años |

| Procedimiento: ingresopaciente | | |
|--------------------------------|------------|---|
| Nombre | Tipo | Descripción |
| tec, pul | Char | Reconocimiento de tecleo de una tecla |
| num, tt | longint | Auxiliar de validación de número de cédula del paciente e iniciador de contador respectivamente |
| Dto | paciente | Record o registro completo de un paciente |
| Ver | boolean | Variable auxiliar de verificación |
| cont, cont1, error, t | integer | Contadores, verificador de error y entero de iteraciones de verificación |
| Aux | string | Auxiliar de lectura de datos para verificaciones |
| Fechent | string[12] | Lectura de fecha de entrada |

| Procedimiento: consultar (num : longint) | | |
|--|-----------|--|
| Nombre | Tipo | Descripción |
| Patolog | Char | Letra que se designa a la patología del paciente |
| Tt | longint | Contador de iteraciones |
| Dto | paciente | Record o registro completo de un paciente |
| Si | boolean | Variable de verificación de registro de la C.I. |
| totaldias, error | integer | Variable del total de los días hospitalizado de un paciente y verificador de error |
| Dar | string[2] | Axiliar de la conversión de días y meses |
| Ay | string[4] | Axiliar de la conversión de años |
| d, m, an, d1, m1, an1 | word | Auxiliar para el cálculo con día, mes y año respectivamente |
| costdia, total | real | Costo por día y costo total respectivamente |

| Procedimiento: modificadatos (num : longint) | | |
|--|----------|---|
| Nombre | Tipo | Descripción |
| pul, aux, deci, tec | char | Auxiliares de lectura de tecla/s específica/s |
| kk, jh, num1 | longint | Contadores de iteraciones y auxiliar de lectura de número de cédula |
| mo, mdi | paciente | Record o registro completo de un paciente: mo (modificándose los datos) mdi (sin modificar) |
| term, ver | boolean | Variable de verificación de registro de la C.I. y |

| | | |
|-------------------------------------|-----------|--|
| | | verificador auxiliar respectivamente |
| totaldias, cont, cont1, t, error | integer | Variable del total de los días hospitalizado de un paciente, contadores y verificador de error |
| Dar | string[2] | Axiliar de la conversión de días y meses |
| Ay | string[4] | Axiliar de la conversión de años |
| d, m, an, d1, m1, an1 | word | Auxiliar para el cálculo con día, mes y año respectivamente |
| fechent, fechalt, aux1 | string | Auxiliar de lectura de fecha de entrada y de alta, y de C.I. y edad respectivamente |

| Procedimiento: eliminapaciente (num : longint) | | |
|--|----------|--|
| Nombre | Tipo | Descripción |
| Tec | char | Auxiliar de lectura de tecla/s específica/s |
| hh, índice | longint | Contador de iteraciones y auxiliar de número de índice de registro de paciente |
| datosE | paciente | Record o registro completo del paciente a eliminar |
| Hallado | boolean | Variable de verificación de registro de la C.I. |
| Faux | archivo | Archivo base de datos de pacientes |

| Procedimiento: alta (num : longint) | | |
|-------------------------------------|-----------|--|
| Nombre | Tipo | Descripción |
| Pul | char | Auxiliar de lectura de tecla/s específica/s |
| tt, pos | longint | Contador de iteraciones y auxiliar de número de índice de registro de paciente |
| Sil | boolean | Auxiliar de verificación de registro de la C.I. |
| totaldias, cont, cont1, error | integer | Variable del total de los días hospitalizado de un paciente, contadores y verificador de error |
| Dar | string[2] | Axiliar de la conversión de días y meses |
| Ay | string[4] | Axiliar de la conversión de años |
| d, m, an, d1, m1, an1 | word | Auxiliar para el cálculo con día, mes y año respectivamente |
| Fechalt | string | Auxiliar de lectura de fecha de alta |
| Totl | real | Monto total a pagar |

| Procedimiento: reporte (mes : string) | | |
|---------------------------------------|----------|---|
| Nombre | Tipo | Descripción |
| Busca | paciente | Record completo de un paciente |
| hs, mins, segs, | word | Lectura de la hora del sistema en términos de |

| | | |
|--------------------------------------|------|---|
| segs100 | | horas, minutos, segundos y centésimas de seg. |
| patA, patB, patC, men, mayM, mayF | real | Ingresos por patología A, B y C, número de menores de edad atendidos, números de adultos mayores atendidos y números de adultas mayores atendidas |
| Tec | char | Auxiliar de lectura de tecla/s específica/s |

| Procedimiento: mostrar | | |
|------------------------|---------|-------------------------|
| Nombre | Tipo | Descripción |
| Z | longint | Contador de iteraciones |

| Procedimiento: menú | | |
|---------------------|-----------|--|
| Nombre | Tipo | Descripción |
| tecla, tec | char | Auxiliar de lectura de tecla/s específica/s |
| ver, sal | boolean | Auxiliar de verificación y de salida del sistema respectivamente |
| nnm, num | longint | Auxiliares de lectura de números y asignación respectivamente |
| Mes | string[2] | Asignación de mes para generar el reporte |
| aux1 | string | Auxiliar de lectura |
| cont, cont1, error | integer | Contadores y verificador de error en conversiones |

Diseño del Algoritmo

Programa Clínica; { Software hecho por José R. Gutierrez para la Clínica "La Mejor" }

uses

crt, dos, printer;

constantes

patoA = 550;

patoB = 430;

patoC = 620;

ingres = 500;

archi = 'BDpac.dat';

tipos

cadena de caracteres12 = cadena de caracteres[12];

paciente = registro;

```

activo : booleano;
nombre, apell : cadena de caracteres[20];
numpac : entero largo;
edad : entero;
sexo, patolg : caracter;
total, ingreso : real;
fechent, fechalt : cadena de caracteres[12];
fin_registro;

```

variables

```

f : archivo de paciente;
Dat, datos : paciente;
ayo, mes, día, sem : entero sin signo;

```

función guardardatos(dd : paciente) : booleano; { *Esta es la función para guardar los datos de un paciente* }

variables

```

dt : paciente;
faux : archivo de paciente;

```

inicio

```

guardardatos ← falso;
asignar a f ← archi;
resetear f;
Si no existe el archivo "BDpac.dat" entonces

```

inicio

```

reescribir(f);
escribir(f,dd);
cerrar(f);
guardardatos ← verdadero;

```

fin_inicio

Sino

inicio

```

faux ← 'auxiliar.dat';
reescribir(faux);
mientras no eof(f) hacer

```

inicio

```

leer(f,dt);
escribir(faux,dt);

```

fin_inicio;

fin_mientras

```

escribir(faux,dd);

```

```

cerrar(f);

```

```

eliminar(f);

```

```

    cerrar(faux);
    renombrar faux a archi;
    guardardatos ← verdadero;
    fin_inicio;
    fin_si;
    fin_inicio;

```

función diahoy : cadena de caracteres12; { *Esta es la función para leer la fecha del día presente y llevarla a formato dd/mm/aaaa* }

variables

```

    mm, dd : cadena de caracteres[2];
    aaaa : cadena de caracteres[4];

```

inicio

```

    leer fecha del sistema (año,mes,dia);
    aaaa ← año ;
    mm ← mes;
    dd ← día;
    si longitud de dd = 1 hacer
        insertar un '0' antes de dd;
    fin_si;
    si longitud de mm = 1 hacer
        insertar un '0' antes de mm;
    fin_si;
    diahoy ← dd/mm/aaaa;
    fin;

```

Función fechval (fech : cadena de caracteres[12]) : booleano; { fech = dd/mm/aaaa }

variables

```

    error1, error2, error3, a, m, d : entero;
    bisiesto : booleano;
    max : byte;
    aux : cadena de caracteres[2];
    año : cadena de caracteres[4];

```

Inicio

```

    si longitud de dd = 1 hacer
        insertar un '0' antes de dd;
    fin_si;
    si longitud de mm = 1 hacer
        insertar un '0' antes de mm;
    fin_si;
    aux ← dd de fech;
    d ← aux, sino error1 = 1;
    aux ← mm de fech;

```

```

    m ← aux, sino error2 = 1;
    año ← aaaa de fech;
    a ← aux, sino error3 = 1;
    si (( error1 + error2 + error3) = 0) entonces
        inicio
            bisiesto ← ((a mod 400 = 0) y (a mod 4000 < > 0)) o ((a mod 4 = 0) y (a mod
100 < > 0));
            en caso de m
                1,3,5,7,8,10,12: hacer max ← 31;
                4,6,9,11: hacer max ← 30;
                2: hacer:
                    inicio
                        Si (bisiesto = verdadero) entonces
                            max ← 29
                        Sino
                            max ← 28;
                        fin_si;
                    fin_inicio;
                fin_caso;
            fechval ← (( m ≥ 1 ) y ( m ≤ 12 )) y (( d ≥ 1 ) y ( d ≤ max ));
            fin
        sino
            fechval ← falso;
        fin_si;
    fin;

```

procedimiento ingresopaciente; { *Esta es la función para ingresar un paciente* }

```

variables
    tec, pul : caracter;
    num, tt : entero largo;
    dto : paciente;
    ver : booleano;
    cont, cont1, error, t : entero;
    aux : cadena de caracteres;
    fechent : cadena de caracteres[12];
inicio
    limpiar pantalla;
    escribir('***** Ingreso Del Paciente *****');
    escribir("");
    inicio
        cont ← 0;
        activo ← verdadero;
    repetir

```

```

ver ← verdadero;
si cont > 0 entonces
  inicio
    escribir("");
    escribir('    << Por Favor Ingrese Nuevamente El Nombre >>');
    escribir(' << Asegúrese Que Solo Contenga Caracteres Alfabéticos >>');
    escribir("");
    fin_inicio;
  fin_si;
  escribir('  Digite el Nombre          : ');
  leer(nombre);
  para t ← 1 hasta (la longitud de "nombre") hacer
    si cada caracter de "nombre" pertenece a ['a'..'z','A'..'Z',' '] entonces
      inicio
        ver ← verdadero;
      fin
    sino
      ver ← falso;
    fin_si;
  fin_para
  cont ← cont + 1;
hasta que (ver = verdadero) y (t = longitud de "nombre");
cont ← 0;
repetir
  ver ← verdadero;
  si cont > 0 entonces
    inicio
      escribir("");
      escribir('    << Por Favor Ingrese Nuevamente El Apellido >>');
      escribir(' << Asegúrese Que Solo Contenga Caracteres Alfabéticos >>');
      escribir("");
      fin_inicio;
    fin_si;
    escribir('  Digite el Apellido          : ');
    leer(apell);
    para t ← 1 hasta (la longitud de "apell") hacer
      si cada caracter de "apell" pertenece a ['a'..'z','A'..'Z',' '] entonces
        inicio
          ver ← verdadero;
        fin
      sino
        ver ← falso;
      fin_si;

```



```

fin_para
cont ← cont + 1;
hasta que (ver = verdadero) y (t = longitud de "apell");
cont ← 0;
repetir
inicio
si cont > 0 entonces
escribir("");
escribir(' << El Nro. De Cédula Ya Está Registrado >>');
escribir(' << Por Favor Ingrese Una C.I. Diferente >>');
escribir("");
fin_si;
error ← 1;
cont1 ← 0;
mientras error <> 0 hacer
inicio
repetir
si cont1 > 0 entonces
escribir("");
escribir(' << Escriba Nuevamente El Nro De Cédula >>');
escribir(' << Que Solo Tenga Caracteres Numéricos >>');
escribir(' << Y Que Sea Un entero Positivo >>');
escribir("");
fin_si;
escribir(' Digite la C.I.      : ');
leer(aux);
asignar a num ← aux, sino error ← 1;
cont1 ← cont1 + 1;
hasta que num ≥ 1;
fin_inicio;
fin_mientras
inicio
ver ← falso;
tt ← 0;
asignar a f ← archi;
resetear f;
Si no existe el archivo "BDpac.dat" entonces
inicio
cerrar(f);
fin_si;
sino
para tt ← 0 hasta (longitud de f) - 1 hacer
buscar en el índice de f;

```

```

    leer datos de cada registro;
    si el paciente fue encontrado entonces
        ver ← verdadero;
    fin_si;
    fin_para
    cerrar(f);
    fin_si;
    fin_inicio;
    fin_inicio;
    cont ← cont + 1;
    hasta que ver = falso;
    numpac ← num;
    error ← 1;
    cont1 ← 0;
    mientras error <> 0 hacer
        inicio
            repetir
                si cont1 > 0 entonces
                    escribir("");
                    escribir(' << Escriba Nuevamente La Edad Entre 0 y 120, >>');
                    escribir(' << Que Solo Tenga Caracteres Numéricos >>');
                    escribir("");
                fin_si;
                escribir(' Digite la Edad      : ');
                leer(aux);
                asignar a num ← aux, sino error ← 1
                cont1 ← cont1 + 1;
            hasta que (0 ≤ num ≤ 120);
        fin_si;
    fin_mientras
    edad ← num;
    cont ← 0;
    rrepetir
        si cont > 0 entonces
            escribir("");
            escribir(' << Ingrese Nuevamente Una Sola Letra >>');
            escribir(' <<   Asegúrese Que Sea F o M   >>');
            escribir("");
        fin_si;
    inicio
        escribir(' Digite el Sexo [F/M]      : ');
        leer(sexo);
        si sexo = ['M','F','m','f'] entonces

```

```

    ver ← verdadero;
    fin_inicio;
    cont ← cont + 1;
hasta que ver = verdadero;
cont ← 0;
repetir
    si cont > 0 entonces
        escribir("");
        escribir(' << Ingrese Nuevamente Una Sola Letra >>');
        escribir(' << Asegúrese Que Sea A, B o C. >>');
        escribir("");
    fin_si;
inicio
    escribir(' Digite la Patología [A/B/C] : ');
    leer(patolg);
    si patolg = ['A','B','C','a','b','c'] entonces
        ver ← falso;
    fin_inicio;
    cont ← cont + 1;
hasta que ver = falso;
ingreso ← ingres;
inicio
    escribir(' Fecha de Ingreso: [H]=Hoy o [M]=Manual');
    repetir
        pul ← tecla presionada (leer tecla);
    hasta que pul = ['H','M'];
    si pul = 'H' entonces
        fechent ← diahoy;
    si pul = 'M' entonces
        cont ← 0;
    repetir
        si cont > 0 entonces
            escribir("");
            escribir(' << Fecha No Valida, Intente Nuevamente >>');
            escribir("");
        fin_si;
        escribir(' Digite la Fecha De Ingreso dd/mm/aaaa ');
        escribir("");
        escribir(' Fecha : ');
        leer(fechent);
        si en la posición 2 de fechent hay un '/' entonces
            insertar '0' antes de la posición 1;
        si en la posición 5 de fechent hay un '/' entonces

```

```

        insertar '0' antes de la posición 4;
        cont ← cont + 1;
        hasta que fechval (fechent) = verdadero;
        fechent ← fechent;
    fin_si;
fin_inicio;
escribir("");
escribir(' >>> Aceptar Datos [S/N] <<<');
repetir
    tec ← tecla presionada (leer tecla);
    hasta que tec = ['S','N'];
    si tec = 'S' entonces
        si guardardatos(datos) = verdadero entonces
            escribir("");
            escribir(' <<< Datos Del Paciente Guardados >>>');
            escribir("");
            escribir(' >>> Pulse Una Tecla Para Regresar <<<');
            leer tecla;
        sino
            escribir("");
            escribir(' Error, Datos Del Paciente No Guardados ');
            escribir("");
            escribir(' Pulse Una Tecla Para Regresar ');
            leer tecla;
        fin_si;
    fin_si;
fin_inicio;
end_inicio;

```

procedimiento consultar(num : entero largo); { *Esta es la función para consultar a un paciente por su C.I.* }

```

variables
    tt : entero largo;
    dto : paciente;
    si : booleano;
    patolg : caracter;
    d, m, an, d1, m1, an1 : entero sin signo;
    dar : cadena de caracteres[2];
    ay : cadena de caracteres[4];
    totaldias, error : entero;
    costdia, total: real;
inicio
    si ← falso;

```

```

tt ← 0;
asignar a f ← archi;
resetear f;
Si no existe el archivo "BDpac.dat" entonces
    inicio
        escribir(' Error Archivo No Encontrado Pulse Una Tecla');
        leer tecla;
        cerrar(f);
    fin_inicio
sino
    inicio
        para tt ← 0 hasta (longitud de f) - 1 hacer
            inicio
                buscar en el índice de f;
                leer datos de cada registro;
                si el paciente fue encontrado entonces
                    si ← verdadero;
                fin_si;
            fin_inicio;
        si si = verdadero entonces
            si el paciente está hospitalizado entonces
                inicio
                    escribir("");
                    escribir(' <<< El Paciente Esta Actualmente Hospitalizado >>> ');
                    escribir("");
                    escribir(' Nombre = ',nombre);
                    escribir(' Apellido = ',apell);
                    escribir(' Sexo = ',sexo);
                    escribir(' Edad = ',edad);
                    escribir(' Num. de Cédula = ',numpac);
                    en caso de edad
                        0..17 : escribir(' Categor ía = Menor De Edad');
                        18..64 : escribir(' Categor ía = Adulto');
                        65..120 : escribir(' Categor ía = Adulto Mayor');
                    fin_caso;
                    escribir(' Patolog ía = ',patolg);
                    si (patolg ía = 'A') o (patolg ía = 'a') entonces
                        costdia ← patoA;
                    fin_si;
                    si (patolg ía = 'B') o (patolg ía = 'b') entonces
                        costdia ← patoB;
                    fin_si;
                    si (patolg ía = 'C') o (patolg ía = 'c') entonces

```

```

    costdia ← patoC;
    fin_si;
    escribir(' Fecha Ingreso  = ',fechent);
    escribir(' Fecha Actual   = ',diahoy);
    totaldias ← (diahoy) - (fechent);
    si totaldias = 0 entonces
        totaldias ← 1
    sino
        totaldias ← totaldias;
    fin_si;
    escribir(' Días Hosp.    = ',totaldias);
    escribir(' Importe Ingreso = ',ingreso);
    escribir(' Importe Diario = ',costdia);
    total ← (costdia * totaldias);
    total ← (total + ingreso);
    escribir(' Total Importe  = ',total);
    escribir("");
    fin_inicio;
    fin_si;
    si si = verdadero entonces
        inicio
            si el paciente ya ha sido dado de alta entonces
                inicio
                    inicio
                        escribir("");
                        escribir(' <<< El Paciente Ya Ha Sido Dado De Alta >>> ');
                        escribir("");
                        escribir(' Nombre      = ',nombre);
                        escribir(' Apellido    = ',apell);
                        escribir(' Sexo       = ',sexo);
                        escribir(' Edad      = ',edad);
                        escribir(' Num. de Cédula = ',numpac);
                        en caso de edad
                            0..17 : escribir(' Categor ía    = Menor De Edad');
                            18..64 : escribir(' Categor ía    = Adulto');
                            65..120 : escribir(' Categor ía    = Adulto Mayor');
                        fin_caso;
                        escribir(' Patolog ía    = ',patolg);
                        si (patolg ía = 'A') o (patolg ía = 'a') entonces
                            inicio
                                costdia ← patoA;
                                fin_inicio;

```

```

fin_si;
si (patolg á = 'B') o (patolg á = 'b') entonces
    inicio
        costdia ← patoB;
        fin_inicio;
    fin_si;
si (patolg á = 'C') o (patolg á = 'c') entonces
    inicio
        costdia ← patoC;
        fin_inicio;
    fin_si;
escribir(' Fecha Ingreso  = ',fechent);
escribir(' Fecha Egreso   = ',fechalt);
totaldias ← (fechalt) - (fechent);
si totaldias = 0 entonces
    totaldias ← 1
sino
    totaldias ← totaldias;
fin_si;
escribir(' Dias Hosp.    = ',totaldias);
escribir(' Importe Ingreso = ',ingreso);
escribir(' Importe Diario = ',costdia);
fin_inicio;
total ← (costdia * totaldias);
total ← (total + ingreso);
escribir(' Total Importe = ',total);
escribir("");
fin_inicio;
fin_si;
fin_inicio
sino
    inicio
        escribir("");
        escribir(' >>> El Paciente No Esta Registrado <<<');
        fin_inicio;
    fin_si;
    cerrar(f);
    fin_inicio;
fin_si;
fin_inicio;

```

procedimiento modificadatos(num : entero largo); { *Función para modificar datos* }

variables

```

d, m, an, d1, m1, an1 : entero sin signo;
dar : cadena de caracteres[2];
ay : cadena de caracteres[4];
mo, mdi : paciente;
kk, jh, num1 : entero largo;
term, ver : booleano;
pul, aux, deci, tec : caracter;
totaldias, cont, cont1, t, error : entero;
fechent, fechalt, aux1 : cadena de caracteres;
inicio
term ← falso;
asignar a f ← archi;
resetear f;
Si no existe el archivo "BDpac.dat" entonces
    inicio
        escribir(' Error Archivo No Encontrado Pulse Una Tecla');
        leer tecla;
        fin_inicio
sino
    inicio
        para jh ← 0 hasta (longitud de f) - 1 hacer
            inicio
                buscar en el índice de f;
                leer datos de cada registro;
                si el paciente fue encontrado entonces
                    inicio
                        term ← verdadero;
                        kk ← jh;
                        mo ← mdi;
                        fin_inicio;
                    fin_si;
                fin_inicio;
            fin_para
            si term = verdadero entonces
                inicio
                    si el paciente está hospitalizado entonces
                        inicio
                            term ← falso;
                            repetir
                                limpiar pantalla;
                            inicio
                                escribir(' ***** Menu Modificaciones *****');
                                escribir("");

```



```

escribir(' <<< El Paciente Esta Actualmente Hospitalizado >>>');
escribir("");
escribir(' N = Nombre      = ',nombre);
escribir(' A = Apellido    = ',apell);
escribir(' C = C.I.        = ',numpac);
escribir(' E = Edad        = ',edad);
escribir(' S = Sexo        = ',sexo);
escribir(' P = Patología    = ',patolg);
escribir(' F = Fecha Ingreso = ',fechent);
escribir("");
escribir(' G = Guardar Cambios y Regresar');
escribir("");
escribir(' <<< Elija Opción >>>');
fin_inicio;
repetir
    deci ← tecla presionada (leer tecla);
hasta que deci = ['N','A','C','E','S','P','F','G'];
limpiar pantalla;
en caso de deci
    'N' : inicio
        cont ← 0;
        repetir
            ver ← verdadero;
            si cont > 0 entonces
                inicio
                    escribir("");
                    escribir(' << Por Favor Ingrese Nuevamente El Nombre >>');
                    escribir(' << Que Solo Contenga Caracteres Alfabéticos >>');
                    escribir("");
                    fin_inicio;
                fin_si;
                escribir(' Nombre : ');
                leer(nombre);
                para t ← 1 hasta (longitud de "nombre") hacer
                    si cada caracter de "nombre" pertenece a ['A'..'Z','a'..'z',' '] entonces
                        inicio
                            ver ← verdadero;
                        fin_inicio
                    sino
                        ver ← falso;
                    fin_si;
                fin_para
                cont ← cont + 1;

```

```

    hasta que (ver = verdadero) y (t = (longitud de "nombre"));
    fin_inicio;
'A' : inicio
    cont ← 0;
    repetir
        ver ← verdadero;
        si cont > 0 entonces
            inicio
                escribir("");
                escribir(' << Por Favor Ingrese Nuevamente El Nombre >>');
                escribir(' << Que Solo Contenga Caracteres Alfabéticos >>');
                escribir("");
            fin_inicio;
        fin_si;
        escribir(' Apellido : ');
        leer(apell);
        para t ← 1 hasta (longitud de "apell") hacer
            si cada caracter de "apell" pertenece a ['A'..'Z','a'..'z',' '] entonces
                inicio
                    ver ← verdadero;
                fin_inicio
            sino
                ver ← falso;
            fin_si;
        fin_para
        cont ← cont + 1;
    hasta que (ver = verdadero) y (t = (longitud de "apell"));
    fin_inicio;
'C' : inicio
    error ← 1;
    cont1 ← 0;
    mientras error <> 0 hacer
        inicio
            si cont1 > 0 entonces
                inicio
                    escribir("");
                    escribir(' << Escriba Nuevamente El Nro De Cédula >>');
                    escribir(' << Que Solo Tenga Caracteres Numéricos >>');
                    escribir("");
                fin_inicio;
            fin_si;
            escribir(' Digite la C.I.: ');
            leer(aux1);

```

```

    asignar a num1 ← aux1, sino error ← 1;
    cont1 ← cont1 + 1;
    fin_inicio;
    numpac ← num1;
    fin_mientras
    fin_inicio;
'E' : inicio
    error ← 1;
    cont ← 0;
    mientras error < > 0 hacer
        inicio
            repetir
                si cont > 0 entonces
                    inicio
                        escribir("");
                        escribir(' << Escriba Nuevamente La Edad Entre 0 y 120, >>');
                        escribir(' << Que Solo Tenga Caracteres Numéricos >>');
                        escribir("");
                    fin_inicio;
                fin_si;
                escribir(' Digite la Edad: ');
                leer(aux1);
                asignar a num1 ← aux1, sino error ← 1;
                cont ← cont + 1;
            hasta que  $0 \leq \text{num1} \leq 120$ ;
        fin_inicio;
        edad ← num1;
    fin_mientras
    fin_inicio;
'S' : inicio
    ver ← falso;
    cont ← 0;
    repetir
        si cont > 0 entonces
            inicio
                escribir("");
                escribir(' << Ingrese Nuevamente Una Sola Letra >>');
                escribir(' << Asegúrese Que Sea F o M >>');
                escribir("");
            fin_inicio;
        fin_si;
    inicio
        escribir(' Digite el Sexo [F/M]: ');

```

```

    leer(aux);
    si aux = ['M','F','m','f'] entonces
        ver ← verdadero;
    fin_si;
    fin_inicio;
    cont ← cont + 1;
    hasta que ver = verdadero;
    sexo ← aux;
    fin_inicio;
'P' : inicio
ver ← falso;
cont ← 0;
repetir
    si cont > 0 entonces
        inicio
            escribir("");
            escribir(' << Ingrese Nuevamente Una Sola Letra >>');
            escribir(' << Asegúrese Que Sea A, B o C. >>');
            escribir("");
        fin_inicio;
    inicio
        escribir(' Digite la Patología [A/B/C]: ');
        leer(aux);
        si aux = ['A','B','C','a','b','c'] entonces
            ver ← verdadero;
        fin_si;
        fin_inicio;
        cont ← cont + 1;
    hasta que ver = verdadero;
    patolg ← aux;
    fin_inicio;
'F' : inicio
cont ← 0;
repetir
    si cont > 0 entonces
        inicio
            escribir("");
            escribir(' << Fecha No Valida, Intente Nuevamente >>');
            escribir("");
        fin_inicio;
    fin_si;
    escribir(' Digite La Fecha De Ingreso: ');
    leer(fechent);

```

```

    si en la posición 2 de fechent hay un '/' entonces
        insertar '0' antes de la posición 1;
    fin_si;
    si en la posición 5 de fechent hay un '/' entonces
        insertar '0' antes de la posición 4;
    fin_si;
    cont ← cont + 1;
    hasta que fechval (fechent) = verdadero;
    fechent ← fechent;
    fin_inicio;
'G' : inicio
    escribir(' >>> Acepta Modificar Datos [S/N] <<<');
    repetir
        tec ← tecla presionada (leer tecla);
    hasta que tec = ['S','N'];
    si tec = 'S' entonces
        inicio
            term ← verdadero;
        fin_inicio;
    si tec = 'N' entonces
        inicio
            clrscr;
            mo ← mdi;
            term ← verdadero;
        fin_inicio;
    fin_si;
    fin_si;
    fin_inicio;
    fin_inicio;
    hasta que term = verdadero;
    mdi ← mo;
    buscar en el índice de f;
    escribir el registro mdi en f;
    fin_inicio;
    fin_si;
    si el paciente ya ha sido dado de alta entonces
        inicio
            term ← falso;
        repetir
            limpiar pantalla;
        inicio
            escribir(' ***** Menu Modificaciones *****');
            escribir("");

```

```

escribir(' <<< El Paciente Ya Ha Sido Dado de Alta >>>');
escribir("");
escribir(' N = Nombre      = ',nombre);
escribir(' A = Apellido    = ',apell);
escribir(' C = C.I.        = ',numpac);
escribir(' E = Edad        = ',edad);
escribir(' S = Sexo        = ',sexo);
escribir(' P = Patología    = ',patolg);
escribir(' F = Fecha Ingreso = ',fechent);
escribir(' X = Fecha Egreso  = ',fechalt);
escribir("");
escribir(' G = Guardar Cambios y Regresar');
escribir("");
escribir(' <<< Elija Opción >>>');
fin_inicio;
repetir
    deci ← tecla precionada (leer tecla);
hasta que deci = ['N','A','C','E','S','P','F','X','G'];
clrscr;
en caso de deci
    'N' : inicio
        cont ← 0;
        repetir
            ver ← verdadero;
            si cont > 0 entonces
                inicio
                    escribir("");
                    escribir('      << Por Favor Ingrese Nuevamente El Nombre >>');
                    escribir(' << Que Solo Contenga Caracteres Alfabéticos >>');
                    escribir("");
                    fin_inicio;
                fin_si;
            escribir(' Nombre : ');
            leer(nombre);
            para t ← 1 hasta (longitud de "nombre") hacer
                si cada caracter de "nombre" pertenece a ['A'..'Z','a'..'z',' '] entonces
                    inicio
                        ver ← verdadero;
                    fin_inicio
                sino
                    ver ← falso;
            fin_si;
        fin_para

```

```

    cont ← cont + 1;
    hasta que (ver = verdadero) y (t = (longitud de "nombre"));
    fin_inicio;
'A' : inicio
    cont ← 0;
    repetir
        ver ← verdadero;
        si cont > 0 entonces
            inicio
                escribir("");
                escribir('    << Por Favor Ingrese Nuevamente El Nombre >>');
                escribir(' << Que Solo Contenga Caracteres Alfabéticos >>');
                escribir("");
            fin_inicio;
        fin_si;
        escribir(' Apellido : ');
        leer(apell);
        para t ← 1 hasta (longitud de "apell") do
            si cada caracter de "apell" pertenece a ['A'..'Z','a'..'z',' '] entonces
                inicio
                    ver ← verdadero;
                    fin_inicio
                sino
                    ver ← falso;
                fin_si;
            fin_para
            cont ← cont + 1;
        hasta que (ver = verdadero) y (t = (longitud de "apell"));
        fin_inicio;
'C' : inicio
    error ← 1;
    cont1 ← 0;
    mientras error <> 0 hacer
        inicio
            si cont1 > 0 entonces
                inicio
                    escribir("");
                    escribir(' << Escriba Nuevamente El Nro De Cédula >>');
                    escribir(' << Que Solo Tenga Caracteres Numéricos >>');
                    escribir("");
                fin_inicio;
            fin_si;
            escribir(' Digite la C.I.: ');

```

```

    leer(aux1);
    asignar a num1 ← aux1, sino error ← 1;
    cont1 ← cont1 + 1;
    fin_inicio;
    numpac ← num1;
fin_mientras
fin_inicio;
'E' : inicio
error ← 1;
cont1 ← 0;
mientras error < > 0 hacer
    inicio
    repetir
        si cont1 > 0 entonces
            inicio
                escribir("");
                escribir(' << Escriba Nuevamente La Edad Entre 0 y 120, >>');
                escribir(' << Que Solo Tenga Caracteres Numéricos >>');
                escribir("");
            fin_inicio;
            escribir(' Digite la Edad: ');
            leer(aux1);
            asignar a num1 ← aux1, sino error ← 1;
            cont1 ← cont1 + 1;
            hasta que  $0 \leq \text{num1} \leq 120$ ;
        fin_inicio;
        edad ← num1;
    fin_mientras
    fin_inicio;
'S' : inicio
ver ← falso;
cont ← 0;
repetir
    si cont > 0 entonces
        inicio
            escribir("");
            escribir(' << Ingrese Nuevamente Una Sola Letra >>');
            escribir(' << Asegúrese Que Sea F o M >>');
            escribir("");
        fin_inicio;
    inicio
        escribir(' Digite el Sexo [F/M]: ');
        leer(aux);

```



```

    si aux = ['M','F','m','f'] entonces
        ver ← verdadero;
    fin_si;
fin_inicio;
cont ← cont + 1;
hasta que ver = verdadero;
sexo ← aux;
fin_inicio;
'P' : inicio
ver ← falso;
cont ← 0;
repetir
    si cont > 0 entonces
        inicio
            escribir("");
            escribir(' << Ingrese Nuevamente Una Sola Letra >>');
            escribir(' << Asegúrese Que Sea A, B o C. >>');
            escribir("");
        fin_inicio;
    fin_si;
    inicio
        escribir(' Digite la Patología [A/B/C]: ');
        leer(aux);
        si aux = ['A','B','C','a','b','c'] entonces
            ver ← verdadero;
        fin_si;
    fin_inicio;
    cont ← cont + 1;
hasta que ver = verdadero;
patolg ← aux;
totaldias ← (fechalt) - (fechent);
si totaldias = 0 entonces
    inicio
        totaldias ← 1;
    fin_inicio;
    si (patolg == 'A') o (patolg == 'a') entonces
        inicio
            total ← (patoA * totaldias);
        fin_inicio;
    si (patolg == 'B') o (patolg == 'b') entonces
        inicio
            total ← (patoB * totaldias);
        fin_inicio;

```

```

si (patolg á = 'C') o (patolg á = 'c') entonces
inicio
    total ← (patoC * totaldias);
fin_inicio;
    total ← (total + ingres);
fin_inicio;
'F' : inicio
cont1 ← 0;
repetir
    si cont1 > 0 entonces
        inicio
            escribir("");
            escribir(' << La Fecha De Ingreso Debe Ser Anterior >>');
            escribir(' << A La Fecha De Egreso 'fechalt,' >>');
            escribir("");
            fin_inicio;
        fin_si;
        cont ← 0;
        repetir
            si cont > 0 entonces
                inicio
                    escribir("");
                    escribir(' << Fecha No Valida, Intente Nuevamente >>');
                    escribir("");
                    fin_inicio;
                fin_si;
                escribir(' Digite La Fecha De Ingreso: ');
                leer(fechent);
                si en la posición 2 de fechent hay un '/' entonces
                    insertar '0' antes de la posición 1;
                fin_si;
                si en la posición 5 de fechent hay un '/' entonces
                    insertar '0' antes de la posición 4;
                fin_si;
                cont ← cont + 1;
                hasta que fechval (fechent) = verdadero;
                fechent ← fechent;
                totaldias ← (fechalt) - (fechent);
                cont1 ← cont1 + 1;
            hasta que totaldias ≥ 0;
            si totaldias = 0 entonces
                inicio
                    totaldias ← 1;

```

```

fin_inicio;
si (patolg á = 'A') o (patolg á = 'a') entonces
inicio
    total ← (patoA * totaldias);
fin_inicio;
si (patolg á = 'B') o (patolg á = 'b') entonces
inicio
    total ← (patoB * totaldias);
fin_inicio;
si (patolg á = 'C') o (patolg á = 'c') entonces
inicio
    total ← (patoC * totaldias);
fin_inicio;
total ← (total + ingres);
fin_inicio;
'X' : inicio
cont1 ← 0;
repetir
    si cont1 > 0 entonces
        inicio
            escribir("");
            escribir(' << La Fecha De Egreso Debe Ser Posterior >>');
            escribir(' << A La Fecha De Ingreso ', fechent, ' >>');
            escribir("");
        fin_inicio;
    fin_si;
    cont ← 0;
    repetir
        si cont > 0 entonces
            inicio
                escribir("");
                escribir(' << Fecha No Valida, Intente Nuevamente >>');
                escribir("");
            fin_inicio;
        fin_si;
        escribir(' Digite La Fecha De Egreso: ');
        leer(fechalt);
        si en la posición 2 de fechalt hay un '/' entonces
            insertar '0' antes de la posición 1;
        fin_si;
        si en la posición 5 de fechalt hay un '/' entonces
            insertar '0' antes de la posición 4;
        fin_si;

```

```

    cont ← cont + 1;
    hasta que fechval (fechalt) = verdadero;
    fechalt ← fechalt;
    totaldias ← (fechalt) - (fechent);
    cont1 ← cont1 + 1;
    hasta que totaldias ≥ 0;
    si totaldias = 0 entonces
        inicio
            totaldias ← 1;
        fin_inicio;
    fin_si;
    si (patolg á = 'A') o (patolg á = 'a') entonces
        inicio
            total ← (patoA * totaldias);
        fin_inicio;
    fin_si;
    si (patolg á = 'B') o (patolg á = 'b') entonces
        inicio
            total ← (patoB * totaldias);
        fin_inicio;
    fin_si;
    si (patolg á = 'C') o (patolg á = 'c') entonces
        inicio
            total ← (patoC * totaldias);
        fin_inicio;
    fin_si;
    total ← (total + ingres);
    fin_inicio;
    'G' : inicio
    escribir(' >>> Acepta Modificar Datos [S/N] <<<');
    repetir
        tec ← tecla presionada (leer tecla);
        hasta que tec = ['S','N'];
        si tec = 'S' entonces
            inicio
                term ← verdadero;
            fin_inicio;
        fin_si;
        si tec = 'N' entonces
            inicio
                clrscr;
                mo ← mdi;
                term ← verdadero;

```

```

        fin_inicio;
        fin_si;
        fin_inicio;
        fin_inicio;
        hasta que term = verdadero;
        mdi ← mo;
        buscar en el índice de f;
        escribir el registro mdi en f;
        fin_inicio;
        fin_si;
        fin_inicio;
        fin_si;
        cerrar(f);
        consultar(num);
        escribir("");
        escribir(' >>> Pulse Una Tecla Para Continuar <<<');
        leer tecla;
        fin_inicio;
        fin_si;
        fin_inicio;

```

procedimiento eliminapaciente(num : entero largo); { *Función para eliminar un paciente* }

variables

```

    hh, índice : entero largo;
    hallado: booleano;
    faux: archivo de paciente;
    datosE: paciente;
    tec : caracter;

```

inicio

```

    consultar(num);
    escribir("");
    escribir(' <<< Pulse Una Tecla Para Continuar >>>');
    leer tecla;
    escribir("");
    hallado ← falso;
    asignar a f ← archi;
    resetear f;

```

Si no existe el archivo "BDpac.dat" entonces

inicio

```

    escribir(' Error Archivo No Encontrado Pulse Una Tecla');
    leer tecla;
    cerrar(f);

```

```

    fin_inicio
sino
    inicio
    para hh ← 0 hasta (longitud de f) - 1 hacer
    inicio
        buscar en el índice de f;
        leer datos de cada registro;
        si el paciente fue encontrado
        inicio
            hallado ← verdadero;
            índice ← hh;
            fin_inicio;
        fin_si;
    fin_inicio;
    si hallado = verdadero entonces
    inicio
        escribir("");
        escribir(' <<< Esta Seguro Que Desea Eliminar El Registro Del Paciente
[S/N] >>>');
        repetir
            tec ← tecla presionada (leer tecla);
        hasta que tec = ['S','N'];
        si tec = 'S' entonces
        inicio
            asignar a faux ← 'auxiliar.dat'
            reescribir(faux);
            para hh ← 0 hasta (longitud de f) - 1 hacer
            inicio
                buscar en el índice de f;
                si hh <> índice entonces
                inicio
                    leer datos de cada registro;
                    escribir datosE en faux;
                    fin_inicio;
                fin_si;
            fin_inicio;
        fin_para
        cerrar(f);
        cerrar(faux);
        eliminar(f);
        renombrar faux ← 'BDpac.dat';
        fin_inicio;
    fin_si;

```

```

    si tec = 'N' entonces
        inicio
            cerrar(f);
            fin_inicio;
        fin_si;
    fn:inicio
sino
    inicio
        cerrar(f);
        fin_inicio;
    fin_si;
fin_inicio;

```

procedimiento alta(num : entero largo); { *Esta es la función para dar de alta a los pacientes* }

```

variables
    totl : real;
    pul : caracter;
    pos, tt : entero largo;
    sil : booleano;
    d, m, an, d1, m1, an1 : entero sin signo;
    dar : cadena de caracteres[2];
    ay : cadena de caracteres[4];
    cont, cont1, totaldias, error : entero;
    fechalt : cadena de caracteres;

```

```

inicio
    asignar a f ← archi;
    resetear f;
    Si no existe el archivo "BDpac.dat" entonces
        inicio
            escribir(' Error Archivo No Encontrado Pulse Una Tecla');
            leer tecla;
        fin_inicio
    sino
        inicio
            sil ← falso;
            para tt ← 0 hasta (longitud de f) - 1 hacer
                inicio
                    buscar en el índice de f;
                    leer datos de cada registro;
                    si numpac = num entonces

```

```

    inicio
    sil ← verdadero;
    pos ← tt;
    fin_inicio;
    fin_si;
    fin_inicio;
fin_para
si sil = verdadero entonces
    inicio
    si el paciente está hospitalizado entonces
        inicio
        escribir("");
        escribir(' <<< Dar De Alta Al Paciente ',nombre,',',apell,',>>>');
        escribir("");
        escribir(' >>> Presione [S] Para Si, o [N] Para No <<<');
        repetir
            pul ← tecla presionada (leer tecla);
        hasta que pul = ['S','N'];
        si pul = 'S' entonces
            inicio
            cont1 ← 0;
            repetir
                si cont1 > 0 entonces
                    inicio
                    escribir("");
                    escribir(' << La Fecha De Alta Debe Ser Posterior >>');
                    escribir(' << A La Fecha De Ingreso ',fechent,',>>');
                    escribir("");
                    fin_inicio;
                fin_si;
                escribir("");
                escribir(' Fecha de Alta: [H]=Hoy o [M]=Manual');
                repetir
                    pul ← tecla presionada (leer tecla);
                hasta que pul = ['H','M'];
                si pul = 'H' entonces
                    fechalt ← diahoy;
                fin_si;
                si pul = 'M' entonces
                    inicio
                    cont ← 0;
                    repetir
                        si cont > 0 entonces

```



```

    inicio
    escribir("");
    escribir(' << Fecha No Valida, Intente Nuevamente >>');
    escribir("");
    fin_inicio;
fin_si;
    escribir(' Digite la Fecha De Alta dd/mm/aaaa ');
    escribir("");
    escribir(' Fecha : ');
    leer(fechalt);
    si en la posición 2 de fechalt hay un '/' entonces
        insertar '0' antes de la posición 1;
    fin_si;
    si en la posición 5 de fechalt hay un '/' entonces
        insertar '0' antes de la posición 4;
    fin_si;
    cont ← cont + 1;
    hasta que fechval (fechalt) = verdadero;
    fechalt ← fechalt;
    fin_inicio;
fin_si;
    totaldias ← (fechalt) - (fechent);
    cont1 ← cont1 + 1;
    hasta que totaldias ≥ 0;
    si totaldias = 0 entonces
        inicio
        totaldias ← 1;
        fin_inicio;
    fin_si;
    si (patolg á = 'A') o (patolg á = 'a') entonces
        inicio
        total ← (patoA * totaldias);
        fin_inicio;
    fin_si;
    si (patolg á = 'B') o (patolg á = 'b') entonces
        inicio
        total ← (patoB * totaldias);
        fin_inicio;
    fin_si;
    si (patolg á = 'C') o (patolg á = 'c') entonces
        inicio
        total ← (patoC * totaldias);
        fin_inicio;

```

```

fin_si;
total ← (total + ingres);
activo ← falso;
limpiar pantalla;
escribir(' ***** Informe de Salida *****');
escribir("");
escribir(' Fecha De Ingreso = ', fechent);
escribir(' Fecha De Alta   = ', fechalt);
escribir(' Dias Hosp.     = ', totaldias);
escribir(' Patología      = ', patolg);
escribir(' Total Importe   = ', total);
escribir("");
escribir(' <<< Pulse Una Tecla Para Regresar >>>');
leer tecla;
buscar en el índice de f;
escribir registro datos en f;
cerrar(f);
fin_inicio;
fin_si;
si pul = 'N' entonces
    inicio
        cerrar(f);
        fin_inicio;
    fin_si;
end_inicio
sino
    inicio
        escribir("");
        escribir(' El Paciente Ya Ha Sido Dado De Alta ');
        escribir("");
        escribir(' <<< Pulse Una Tecla Para Regresar >>>');
        leer tecla;
        cerrar(f);
        fin_inicio;
    fin_si;
    fin_inicio;
    fin_si;
    fin_inicio;
fin_si;
si numpac < > num entonces
    inicio
        escribir("");
        escribir(' >>> El Paciente No Esta Registrado <<<');

```

```

    escribir("");
    escribir(' <<< Pulse Una Tecla Para Regresar >>>');
    leer tecla;
    cerrar(f);
    fin_inicio;
fin_inicio;

```

procedimiento reporte(mes : cadena de caracteres); { *Función generar reporte* }

variables

```

    busca : paciente;
    hs, mins, segs, segs100 : entero sin signo;
    patA, patB, patC, men, mayM, mayF : real;
    tec : caracter;

```

inicio

```

    patA ← 0;
    patB ← 0;
    patC ← 0;
    men ← 0;
    mayM ← 0;
    mayF ← 0;
    asignar a f ← archi;
    resetear f;
    mientras no eof(f) hacer

```

 inicio

 leer datos de cada registro;

 si ((patolog ía = 'A') o (patolog ía = 'a')) y (el paciente ya ha sido dado de alta) y
 (mes = mes de la fecha de alta) entonces

 inicio

 patA ← patA + busca.total;

 fin_inicio;

 fin_si;

 si ((patolog ía = 'B') o (patolog ía = 'b')) y (el paciente ya ha sido dado de alta) y
 (mes = mes de la fecha de alta) entonces

 inicio

 patB ← patB + busca.total;

 fin_inicio;

 fin_si;

 si ((patolog ía = 'C') o (patolog ía = 'c')) y (el paciente ya ha sido dado de alta) y
 (mes = mes de la fecha de alta) entonces

 inicio

 patC ← patC + busca.total;

 fin_inicio;

 fin_si;

```

    si (edad < 18) y (mes = mes de la fecha de entrada) entonces
        inicio
            men ← men + 1;
        fin_inicio;
    fin_si;
    si (edad ≥ 65) y (mes = mes de la fecha de entrada) y ((sexo = 'F') o (sexo = 'f'))
    entonces
        inicio
            mayF ← mayF + 1;
        fin_inicio;
    fin_si;
    si (edad ≥ 65) y (mes = mes de la fecha de entrada) y ((sexo = 'M') o (sexo =
'm')) entonces
        inicio
            mayM ← mayM + 1;
        fin_inicio;
    fin_si;
    fin_inicio;
fin_mientras
si mes = '01' entonces
mes ← 'Enero';
si mes = '02' entonces
mes ← 'Febrero';
si mes = '03' entonces
mes ← 'Marzo';
si mes = '04' entonces
mes ← 'Abril';
si mes = '05' entonces
mes ← 'Mayo';
si mes = '06' entonces
mes ← 'Junio';
si mes = '07' entonces
mes ← 'Julio';
si mes = '08' entonces
mes ← 'Agosto';
si mes = '09' entonces
mes ← 'Septiembre';
si mes = '10' entonces
mes ← 'Octubre';
si mes = '11' entonces
mes ← 'Noviembre';
si mes = '12' entonces
mes ← 'Diciembre';

```

```

escribir("");
escribir(' En El Mes De ',mes,' Se Ha Recaudado:');
escribir(' Por Atención A Pacientes Con Patología Tipo A: ',patA,' Bs');
escribir(' Por Atención A Pacientes Con Patología Tipo B: ',patB,' Bs');
escribir(' Por Atención A Pacientes Con Patología Tipo C: ',patC,' Bs');
escribir(' Fueron Atendidos ',men,' Pacientes Menores De Edad');
escribir(' Fueron Atendidas ',mayF,' Damas Adultas Mayores');
escribir(' Fueron Atendidos ',mayM,' Caballeros Adultos Mayores');
escribir(' Reporte Estadístico Generado El ',diahoy,' A Las ',hs,':',mins,':',secs);
escribir("");
escribir(' > Pulse [I] Para Imprimir o [R] Para Regresar Al Menu Principal <');
repetir
    tec ← tecla presionada (leer tecla);
hasta que tec = ['I','R'];
si tec = 'I' entonces
    inicio
        imprimir, ' En El Mes De ',mes,' Se Ha Recaudado:');
        imprimir, ' Por Atención A Pacientes Con Patología Tipo A: ',patA,' Bs');
        imprimir, ' Por Atención A Pacientes Con Patología Tipo B: ',patB,' Bs');
        imprimir, ' Por Atención A Pacientes Con Patología Tipo C: ',patC,' Bs');
        imprimir, ' Fueron Atendidos ',men,' Pacientes Menores De Edad');
        imprimir, ' Fueron Atendidas ',mayF,' Damas Adultas Mayores');
        imprimir, ' Fueron Atendidos ',mayM,' Caballeros Adultos Mayores');
        imprimir, ' Reporte Estadístico Generado El ',diahoy,' A Las
        ',hs,':',mins,':',secs);
    fin_inicio;
    fin_si;
    leer tecla;
    cerrar(f);
    fin_inicio;

procedimiento ordenar; { *Función que ordena los datos dentro del registro* }
variables
    Dat, E1, A : paciente;
    ar, ar2 : entero largo;
inicio
    asignar a f ← archi;
    resetear f;
    Si no existe el archivo "BDpac.dat" entonces
        inicio
            escribir(' Error');
            readln;
            exit;

```

```

    fin_inicio
sino
    para ar ← 0 hasta (longitud de f) - 1 hacer
        buscar en el índice de f;
        leer datos de cada registro;
    inicio
        para ar2 ← ar + 1 hasta (longitud del archivo f) - 1 hacer
            buscar en el índice de f;
            leer datos de cada registro;
            si numpac > E1.numpac entonces
                A ← Dat;
                Dat ← E1;
                E1 ← A;
            buscar en el índice de f;
            escribir el registro Dat en f;
            buscar en el índice de f;
            escribir el registro E1 en f;
        fin_si;
    fin_para
fin_inicio;
fin_para
fin_si;
cerrar(f);
fin_inicio;

procedimiento mostrar; { *Función que muestra la lista de los pacientes* }
variable
    z : entero largo;
inicio
    asignar a f ← archi;
    resetear f;
    Si no existe el archivo "BDpac.dat" entonces
        inicio
            escribir("");
            escribir(' Error');
            leer tecla;
            exit;
        fin_inicio
    sino
        inicio
            escribir("");
            escribir(' >>> Registro de Pacientes <<<');
            escribir("");

```

```

para z ← 0 hasta (longitud de f) - 1 hacer
    inicio
        buscar en el índice de f;
        leer datos de cada registro;
        escribir(' ',numpac,' ',nombre,' ',apell);
        fin_inicio;
    fin_para
    cerrar(f);
    escribir("");
    escribir(' <<< Pulse Una Tecla Para Regresar >>>');
    leer tecla;
    fin_inicio;
fin_si;
fin_inicio;

procedimiento menu; { *Menú principal o General que llama a las funciones* }
variables
    tecla, tec : caracter;
    ver, sal : booleano;
    nrm, num : entero largo;
    mes : cadena de caracteres[2];
    aux1 : cadena de caracteres;
    cont1, cont, error : entero;
inicio
    sal ← falso;
    repetir
        clrscr;
        escribir(' ***** Menu Principal *****');
        escribir("");
        escribir(' I = Incluir');
        escribir(' C = Consultar');
        escribir(' M = Modificar');
        escribir(' E = Eliminar');
        escribir(' A = Alta');
        escribir(' O = Ordenar Registro');
        escribir(' R = Reporte Mensual');
        escribir(' S = Salir');
        escribir("");
        escribir(' <<< Elija Una Opción >>>');
        repetir
            tecla ← tecla presionada (leer tecla);
        hasta que tecla = ['I','C','M','E','A','S','R','O'];
        limpiar pantalla;

```

```

en caso de
'T' : ingresopaciente;
'C' : inicio
    error ← 1;
    cont ← 0;
    mientras error <> 0 hacer
        inicio
            si cont > 0 entonces
                inicio
                    escribir("");
                    escribir(' << Escriba Nuevamente El Nro De Cédula >>');
                    escribir(' << Que Solo Tenga Caracteres Numéricos >>');
                    escribir("");
                fin_inicio;
                escribir(' Digite la C.I. del Paciente : ');
                leer(aux1);
                asignar a num ← aux1, sino error ← 1;
                cont ← cont + 1;
            fin_inicio;
            nrm ← num;
            consultar(nrm);
            escribir("");
            escribir(' >>> Pulse Una Tecla Para Continuar <<<');
            leer tecla;
            fin_inicio;
'M' : inicio
    error ← 1;
    cont ← 0;
    mientras error <> 0 hacer
        inicio
            si cont > 0 entonces
                inicio
                    escribir("");
                    escribir(' << Escriba Nuevamente El Nro De Cédula >>');
                    escribir(' << Que Solo Tenga Caracteres Numéricos >>');
                    escribir("");
                fin_inicio;
            fin_si;
            escribir(' Digite la C.I. del Paciente : ');
            leer(aux1);
            asignar a num ← aux1, sino error ← 1;
            cont ← cont + 1;
        fin_inicio;

```



```

    nnm ← num;
    modificadatos(nnm);
fin_mientras
fin_inicio;
'E' : inicio
error ← 1;
cont ← 0;
mientras error <> 0 hacer
    inicio
        si cont > 0 entonces
            inicio
                escribir("");
                escribir(' << Escriba Nuevamente El Nro De Cédula >>');
                escribir(' << Que Solo Tenga Caracteres Numéricos >>');
                escribir("");
            fin_inicio;
            escribir(' Digite la C.I. del Paciente : ');
            leer(aux1);
            asignar a num ← aux1, sino error ← 1;
            cont ← cont + 1;
        fin_inicio;
    fin_mientras
    nnm ← num;
    eliminapaciente(nnm);
fin_inicio;
'A' : inicio
error ← 1;
cont ← 0;
mientras error <> 0 hacer
    inicio
        si cont > 0 entonces
            inicio
                escribir("");
                escribir(' << Escriba Nuevamente El Nro De Cédula >>');
                escribir(' << Que Solo Tenga Caracteres Numéricos >>');
                escribir("");
            fin_inicio;
            escribir(' Digite la C.I. del Paciente : ');
            leer(aux1);
            asignar a num ← aux1, sino error ← 1;
            cont ← cont + 1;
        fin_inicio;
    fin_mientras

```

```

    nnm ← num;
    alta(nnm);
    fin_inicio;
'O' : inicio
    ordenar;
    mostrar;
    fin_inicio;
'R' : inicio
    cont ← 0;
    repetir
        error ← 1;
        mientras error <> 0 hacer
            inicio
                si cont > 0 entonces
                    inicio
                        escribir("");
                        escribir(' Escriba Nuevamente El Numero del Mes');
                        escribir(' >>> En Un Rango Del 1 al 12 <<<');
                        escribir("");
                    fin_inicio;
                fin_si;
                escribir(' Digite El Mes Del Cual Desea Generar El Reporte: ');
                leer(aux1);
                asignar a num ← aux1, sino error ← 1;
                cont ← cont + 1;
            fin_inicio;
        fin_mientras
    hasta que num = [1..12];
    asignar a mes ← num;
    si longitud de mes = 1 entonces
        insertar '0' antes;
        reporte(mes);
    fin_inicio;
'S' : sal ← verdadero;
    fin_inicio;
    hasta que sal = verdadero;
fin_inicio;

inicio
    limpiar pantalla;
    menu;
fin.

```

Prueba del Algoritmo (Unidad 7)

Prueba de Robustez del Algoritmo

El usuario puede, en la introducción de datos, cometer errores por comisión u omisión de las precondiciones de entrada, es ahí donde entra en juego el concepto de robustez del algoritmo, que no es otra cosa que la validación de los datos de entrada y sus precondiciones no solo del valor de la variable, sino también del formato; es decir, al momento de ingresar cualquier dato se indica el formato en el que debe ingresarlo y en caso que no se cumpla con los parámetros preestablecidos en el algoritmo debe haber un rechazo del dato que no cumple con las condiciones y repetir la petición de entrada.

Por Ejemplos:

Ingreso de nombre y apellido:

Las variables nombre y apellido (nombre, apell) se han definido como cadena de 20 caracteres, sin embargo un nombre no debe incluir en su cadena, caracteres especiales ni números, sólo letras y espacios, éste último porque puede usarse un apellido de casada (caso de una dama) o bien pueda tenerse un apellido o nombre de dos palabras, como por ejemplo Del Monte, Del Valle, De Leon, entre otros. La fracción del algoritmo par el ingreso de nombre es similar al de apellido, solo se difieren en la definición de la variable y a que cada cual se almacena en un lugar del registro para tal fin; aquí analizaremos su robustez y amigabilidad:

```
cont ← 0; {se inicializa un contador}
repetir
    ver ← verdadero; {se asigna verdadero a la variable de verificación}
    si cont > 0 entonces {Si el ciclo ya ha pasado la primera vez muestra el mensaje}
        inicio
            escribir("");
```

```

    escribir('    << Por Favor Ingrese Nuevamente El Nombre >>');
    escribir(' << Asegúrese Que Solo Contenga Caracteres Alfabéticos >>');
    escribir("");
    fin_inicio;
fin_si;
escribir(' Digite el Nombre      : ');
leer(nombre);
para t ← 1 hasta (la longitud de "nombre") hacer { verifica que cada carácter del
string “nombre” pertenezca a los caracteres alfabéticos más espacio}
    si cada caracter de "nombre" pertenece a ['a'..'z','A'..'Z',' '] entonces
        inicio
            ver ← verdadero; {de ser así, ratifica verdadero a “ver”}
        fin
    sino
        ver ← falso; {sino, lo cambia a falso}
    fin_si;
fin_para
cont ← cont + 1;
hasta que (ver = verdadero) y (t = longitud de "nombre"); {esto se repite hasta que
ver = verdadero y se hayan revisado todos los caracteres de “nombre”}

```

De modo que no sólo hace robusta la entrada de éste dato, sino que lo hace amigable al usuario advirtiéndole sobre su error.

Ingreso de C.I.:

La variable C.I. (numpac) se ha definido como un entero largo, de modo que no debe incluir caracteres especiales ni alfabéticos, sólo números. La fracción del algoritmo par el ingreso de C.I. es similar a la del ingreso de la edad, con la diferencia que la edad está acotada de 0 a 120 años. Aquí analizaremos su robustez y amigabilidad para el caso de la función ingresopaciente:

```

cont ← 0; {se inicializa el contador}
repetir
    inicio
        si cont > 0 entonces {si ya se ha terminado una vez el ciclo repetir, es porque el
paciente ya está registrado}
            inicio {y se mostrará el mensaje siguiente}

```

```

    escribir("");
    escribir(' << El Nro. De Cédula Ya Está Registrado >>');
    escribir(' << Por Favor Ingrese Una C.I. Diferente >>');
    escribir("");
    fin_inicio;
fin_si;
error ← 1;
cont1 ← 0;
mientras error <> 0 hacer {esto se repetirá hasta que error = 0}
    inicio
        repetir
            si cont1 > 0 entonces {si el segundo contador es mayor que 0, es porque el
ciclo mientras ya se ha cumplido al menos una vez, esto indica no se ha introducido
sólo números}
                inicio {y arrojará el siguiente mensaje}
                    escribir("");
                    escribir(' << Escriba Nuevamente El Nro De Cédula >>');
                    escribir(' << Que Solo Tenga Caracteres Numéricos >>');
                    escribir(' << Y Que Sea Un entero Positivo >>');
                    escribir("");
                fin_inicio;
            fin_si;
            escribir(' Digite la C.I.      : ');
            leer(aux);
            asignar a num ← aux, sino error ← 1; {Al convertir la lectura del string
“aux” en entero “num”, si no se puede hacer, es debido a que “aux” contiene más que
sólo números, y asignará a error el valor de 1}
            cont1 ← cont1 + 1; {contador del ciclo repetir}
            hasta que num ≥ 1; {hasta que sea un entero positivo}
        fin_inicio;
    fin_mientras
    inicio
        ver ← falso; {se le asigna “falso” a “ver”}
        tt ← 0;
        asignar a f ← archi;
        resetear f; {se busca el archivo "BDpac.dat"}
        Si no existe el archivo "BDpac.dat" entonces
            inicio {si no existe se cierra el auxiliar f}
                cerrar(f);
            fin_si;
        sino {y si existe}
            inicio
                para tt ← 0 hasta (longitud de f) - 1 hacer

```

```

inicio
  buscar en el índice de f; {se busca el número de cédula}
  leer datos de cada registro;
  si el paciente fue encontrado entonces {si se encuentra}
    inicio
      ver ← verdadero; {se le asigna “verdadero” a “ver”}
      fin_inicio;
    fin_si;
  fin_inicio;
fin_para
  cerrar(f); {se cierra f}
fin_inicio;
fin_si;
fin_inicio;
fin_inicio;
cont ← cont + 1;
hasta que ver = falso; {se repite hasta que ver sea falso (es decir, que el paciente
no esté registrado)}
numpac ← num; {se asigna el número de cédula}

```

Esta subrutina en la entrada de la C.I. del ingreso de un paciente permite, verificar si el número introducido contiene sólo números, que éste sea un entero positivo, caso contrario arrojará el mensaje correspondiente, luego verificar si está registrado en la base de datos, y si lo está arrojar el mensaje que ingrese un número de cédula diferente pues el ingresado ya está registrado.

Para el resto de caso en que deberá ingresarse un número de cédula, el procedimiento es similar, sólo que, en vez de verificar que no esté registrado para continuar, deberá verificar que esté registrado para seguir, como por ejemplo en los submenús consultar y modificar, y para el caso del ingreso a menú alta, deberá verificar que esté registrado y activo (hospitalizado).

Ingreso de la edad

En el caso del ingreso de la edad, se debe verificar que sólo se escriba un número entero positivo entre el 0 y 120 inclusive, acá la subrutina:

```

error ← 1; {Se le asigna el valor 1 a error}
cont1 ← 0; {se inicializa el contador}
mientras error <> 0 hacer {mientras exista el error}
    inicio
    repetir
        si cont1 > 0 entonces {si el contador es mayor que uno es porque el ciclo se ha
completado al menos una vez}
            inicio {y arroja el siguiente mensaje}
                escribir("");
                escribir(' << Escriba Nuevamente La Edad Entre 0 y 120, >>');
                escribir(' << Que Solo Tenga Caracteres Numéricos >>');
                escribir("");
                fin_inicio;
            fin_si;
            escribir(' Digite la Edad          : ');
            leer(aux);
            asignar a num ← aux, sino error ← 1 {se convierte el string “aux” a entero
“num”, sino se logra es porque contiene más que números}
            cont1 ← cont1 + 1;
            hasta que (0 ≤ num ≤ 120); {se repite hasta que esté dentro del rango}
            fin_si;
        fin_mientras {cierre del ciclo mientras}
        edad ← num; {se asigna la edad}

```

Acá se evidencia con claridad el mismo protocolo para los casos anteriores: se verifica que la entrada sea únicamente numérica, se verifica que esté dentro de un rango, y de no cumplirse se arroja el mensaje correspondiente.

Ingreso de fechas

El caso de ingreso de fechas ya fue analizado en el apartado “Definición de Abstracciones” (Resolución del problema usando la metodología MAPS). Sin embargo, existe otra cuestión relacionada con las fechas que debe ser analizada, y es en el caso del submenú modificar, el usuario puede, por omisión, modificar las fechas de alta o entrada de un paciente, o ambas, y por error dejarlas invertidas, es decir,

dejar la fecha de ingreso posterior a la de egreso, lo cual arrojará un valor negativo a la variable totaldías, para ello se realizó la siguiente subrutina:

```
'F' : inicio
      cont1 ← 0; {inicializar contador}
      repetir
        si cont1 > 0 entonces {sin contador es mayor a 0 es porque el ciclo repetir ya
se completó al menos una vez}
          inicio {si es así se arrojará el siguiente mensaje}
            escribir("");
            escribir(' << La Fecha De Ingreso Debe Ser Anterior >>');
            escribir(' << A La Fecha De Egreso 'fechalt,' >>');
            escribir("");
          fin_inicio;
        fin_si;
      cont ← 0; {se inicializa otro contador}
      repetir
        si cont > 0 entonces {sin el contador es mayor a cero es porque éste
ciclo repetir se ha completado al menos una vez}
          inicio {si es así arrojar el siguiente mensaje}
            escribir("");
            escribir(' << Fecha No Valida, Intente Nuevamente >>');
            escribir("");
          fin_inicio;
        fin_si;
      escribir(' Digite La Fecha De Ingreso: ');
      leer(fechent);
      si en la posición 2 de fechent hay un '/' entonces
        insertar '0' antes de la posición 1;
      fin_si;
      si en la posición 5 de fechent hay un '/' entonces
        insertar '0' antes de la posición 4;
      fin_si;
      cont ← cont + 1;
      hasta que fechval (fechent) = verdadero; {hasta que la fecha sea válida}
      fechent ← fechent;
      totaldias ← (fechalt) - (fechent);
      cont1 ← cont1 + 1;
      hasta que totaldias ≥ 0; {hasta que la fecha de egreso sea posterior a la fecha
de ingreso}
```


si totaldias = 0 entonces {a partir de aquí recalcula el monto total para guardarlo en su respectiva variable, ya que de no modificarse, se arrojarán datos estadísticos errados}

```
inicio
totaldias ← 1;
fin_inicio;
si (patología = 'A') o (patología = 'a') entonces
inicio
total ← (patoA * totaldias);
fin_inicio;
si (patología = 'B') o (patología = 'b') entonces
inicio
total ← (patoB * totaldias);
fin_inicio;
si (patología = 'C') o (patología = 'c') entonces
inicio
total ← (patoC * totaldias);
fin_inicio;
total ← (total + ingres);
fin_inicio;
```

De manera similar se hizo con el caso de la modificación de la fecha de egreso y con el caso de la entrada de fecha de egreso en el menú alta.

A lo largo de todo el sistema se hacen verificaciones de éste tipo, como lo es en el ingreso del género sexual y el tipo de patología, guiando al usuario a sólo introducir las opciones correctas, de la misma manera se comprobó que a la modificación del tipo de patología se recalcula el monto final pagado y se almacena en su respectiva variable.

De modo que el sistema realizado es amigable, confiable y robusto, capaz de guiar al usuario a las opciones correctas e impide en su totalidad la entrada de datos que no cumplan con las pre y post condiciones de entrada y salida.

Codificación del Algoritmo

(Unidad 8)

La codificación del algoritmo en lenguaje Turbo Pascal 7.0 fue probado y compilado en Turbo Pascal 7.0 y en Free Pascal, se utilizó Geany y Notepad++ para la edición y organización del código. Se entrega en CD contentivo de los archivos digitales del informe del trabajo practico en formato *.pdf, el archivo con el código del programa en formato *.pas y el ejecutable “clinica.exe” con su respectiva base de datos “BDpac.dat” con datos pre-cargados para su evaluación.

Conclusión

Como se dijo al inicio, aprender a programar no es cosa de pocas horas, pero existe en su haber una gran satisfacción por el logro alcanzado que es imposible de ocultar. La programación estructurada en Pascal, aunque es poco práctica para fines productivos, indica el camino a seguir para quienes quieran hacer de la programación un medio de vida profesional o para quienes quieran tenerla como herramienta de uso práctico para resolver problemas de baja, media y gran magnitud.

La resolución de un problema de codificación de algoritmos usando la tecnología MAPS no sólo lo hace más fácil, sino que su propia forma o “algoritmo” a seguir, enseña que el seguimiento de pasos lógicos, formales, estructuralmente ordenados, facilita la comprensión del problema, su análisis y posterior resolución.

Un software o sistema no robusto, es un sistema lleno de errores, bucles interminables y cierres inesperados, debido a que aunque se conozcan las pre y post condiciones de entrada y salida, se errará constantemente por no hacer el sistema las verificaciones necesarias. Lograr que el sistema sea robusto es un gran avance, pero lo será mayor aún si aparte tiene un acabado amigable.

El sistema realizado para la ficticia Clínica La Mejor, se enfoca en cumplir las expectativas pautadas para el trabajo práctico, en guardar y respetar las pre y post condiciones de entrada y salida, en ser robusto y amigable. Se agradecerá a la profesora cualquier corrección u opinión acerca del mismo, ya que éstas ayudarán a hacerlo cada vez mejor.