

# Programación I

## Práctico de Máquina Integrador

**Fecha de control: 12/11/2019.**

**Fecha de entrega: 19/11/2019**

**Grupo: 2 (dos) alumnos**

Se desea realizar un programa en lenguaje C que permita administrar la atención de urgencias de un hospital público.

La atención es realizada por diferentes agentes de salud de la siguiente manera

1- El paciente por orden de llegada es anotado por la secretaria

2- Luego de tomar los datos es revisado por una enfermera quien determina el grado de urgencia del paciente (ROJO: MUY URGENTE, AMARILLO: URGENCIA MEDIA; VERDE: NO URGENTE). Luego de determinar el nivel de urgencia el paciente vuelve a la sala de espera para ser atendido por el médico.

3- El paciente es atendido por el médico según orden de llegada y nivel de urgencia, es decir primero las urgencias de color rojo, luego las amarillas y por ultimo las verdes.

Nota: No se consideran casos de ingreso en ambulancia, accidente.

### 1- Para administrar los pacientes se pide:

a) Implementar el **TDA PACIENTE (paciente .h)** el cual define el tipo Paciente (mantiene los datos de cada uno de ellos) según se detalla en el punto 2.) y sus respectivas operaciones.

b) Modificar la lista vinculada dinámica de manera tal que permita almacenar los Pacientes que ingresan según orden de llegada esto debe realizarse definiendo el **TDA lista\_paciente (lista\_paciente .h)**. La lista debe contener las operaciones básicas (ver punto 3).

b) Implementar el **TDA lista\_prioridad (lista\_prioridad.h)** la cual almacene los paciente luego de ser revisados por la enfermera, **ordenados** por nivel de urgencia y orden de llegada.

c) realizar un programa en C que utilice los TDA definidos y que permita cumplir con las funcionalidades según se describe en el punto 4.

### 2- Definir el TDA PACIENTE el cual contiene.

Definición de la estructura de datos de manera tal que pueda almacenar por cada paciente los siguientes datos:

**Nombre:** string

**Apellido:** string

**Dni:** solo almacena el número, asuma que todos los pacientes tienen DNI

**Edad:** dato no obligatorio

**Hora ingreso** (que a su vez es una estructura con hora y minuto) Rango Horario de 00 a 23

**Hora atención:** (ídem anterior, cargado cuando el paciente es atendido por el médico)

**Nivel de urgencia:** entero (1- ROJO,2- AMARILLO,3 - VERDE)

**Atendido:** verdadero o falso

Los tipos no especificados son determinados por el alumno.

### Definición de las operaciones:

Defina las funciones necesarias de manera tal que se permita:

Inicializar el TDA (1) inicializa el PACIENTE colocando la hora de ingreso, introducida por el usuario, no se le pide al paciente.

Cargar cada uno de los campos restantes por separado. (7)

Modificar por separado los campos: edad, hora atención, nivel urgencia, atendido (4)

Mostrar cada uno de los campos por separado. (8)

### 3- definir el TDA lista\_paciente

Utilice la implementación de lista brindada por la cátedra de manera que pueda almacenar pacientes. Modificar las estructuras de datos y/o operaciones de la lista (insert, suppress, isempty, isfull, forward, init, reset) que sea necesario.

El programa tendrá **dos listas**; una lista de ingreso que funciona como FILA UTILIZANDO **lista\_paciente** y una lista ordenada de pacientes según la prioridad y hora de llegada del paciente con **lista\_prioridad**

**Lista prioridad debe ser modificada de manera que inserte ordenado.**

### 4- Definir un programa que permita realizar las siguientes funcionalidades:

- a) cargar una cantidad n de pacientes que ingresan a urgencia por teclado, se solicita nombre, apellido, dni, mutual y edad (opcional). La cantidad es ingresada por el usuario. Considere que la hora de ingreso es cargada por la secretaria antes de solicitar los datos al paciente.
- b) mostrar siguiente; muestra los datos del paciente siguiente para que sea atendido por la enfermera
- c) mostrar siguiente; muestra los datos del paciente siguiente para que sea atendido por el médico.
- d) nivel de urgencia; asigna el nivel de urgencia y se pasa a la lista con prioridad eliminándolo de la lista de ingreso.
- e) atendido por el medico; carga la hora en el que el medico atendió al paciente
- f) modificar edad
- g) no fue atendido; en este caso el paciente se retiró sin ser atendido por el médico, colocando además la hora de egreso como 24 (hora invalida)
- h) mostrar el listado de los pacientes atendido por nivel el cual es solicitado al usuario.
- i) mostrar nombre, apellido, dni, nivel de urgencia y tiempo de espera de todos los pacientes atendidos.
- j) mostrar el paciente que esperó más tiempo (sin considerar los no atendidos) esta función debe ser recursiva; en caso de haber más de un paciente mostrarlos.
- k) cargar en el archivo noatendidos.txt todos los pacientes que no fueron atendidos
- l) borrar de la lista\_prioridad todos los pacientes que fueron atendidos o se retiraron sin ser atendidos; debe indicar cuantos se borraron
- ll) cargar del archivo precarga.txt a la lista de ingreso un listado de pacientes; los pacientes se encuentran en el archivo ordenados por hora. Los datos que tiene cada paciente son los mismos que en el punto a)
- m) dado el dni de un paciente mostrar cuantos pacientes antes hay ya sea en la lista de ingreso o en la lista de prioridad.
- n) mostrar cuantos pacientes hay en espera en ingreso y en lista de prioridad.
- ñ) mostrar cuantos pacientes hay de cada nivel de urgencia en lista de prioridad sin ser **atendidos (SOLO ALUMNOS LICENCIATURA)**, para esto buscar al paciente.
- o) almacenar en el archivo únicamente los datos de los pacientes que fueron atendidos; en archivos separados por nivel de urgencia. **(SOLO ALUMNOS LICENCIATURA).**

Realice un menú de opciones que permita realizar las funcionalidades solicitadas. Considere que se realiza un programa que luego será utilizado por un usuario, es por esto **NO OLVIDAR los carteles!!**

**Todas las funcionalidades se realizan en forma aleatoria.**

**En el caso que un dato u operación no pueda realizarse debe informarlo por pantalla.**

**Los archivos deben ser legibles desde un editor de texto.**

**Nota.** Recuerde que en todos los casos debe manejar los TDA con las funciones propias de los mismos sin violar la semántica correspondiente a la estructura de almacenamiento. Considere el uso de tipo y estructuras auxiliares en caso de ser necesarias.