



Facultad de Ingeniería y
Ciencias Aplicadas

ISWZ1102-3065_3068-PROGRAMACION I

Proyecto Integrador

07 JULIO

Udla: Universidad de las Américas
Docente: Nicolás Alejandro Rojas Cajamarca

Creado por:

Steven Alejandro Acosta Hidalgo
Michael Saul Tipantuña Ramírez



Contents

Creado por:	1
A. INTRODUCCIÓN	3
B. JUSTIFICACIÓN	3
C. OBJETIVOS	3
D. ANÁLISIS DEL PROBLEMA	4
E. ESTABLECIMIENTO DE PROPUESTAS DE SOLUCIÓN	5
Propuestas integrales de solución:	5
F. EVALUACIÓN DE PROPUESTAS DE SOLUCIÓN	7
G. DEFINICIÓN DEL PLAN DE TRABAJO Y METODOLOGÍA	7
H. Diseño de la propuesta de solución	8
I. EJECUCIÓN DE ESCENARIOS DE PRUEBA	22
ESCENARIO 1	22
ESCENARIO 2	22
ESCENARIO 3	23
ESCENARIO 4	23
ESCENARIO 5	23
ESCENARIO 6	23
ESCENARIO 7	23
ESCENARIO 8	23
J. CONCLUSIONES Y RECOMENDACIONES	23
Recomendaciones:	24
Conclusiones:	24
K. ANEXOS:	25
L. BIBLIOGRAFÍAS:	25

A. INTRODUCCIÓN

El proyecto integrador que presentamos pretende exponer una adecuada solución a la problemática, esto lo desarrollaremos en base a todos los conocimientos adquiridos de la materia de programación, donde se ejecutará una propuesta basada en un sistema de gestión de inventarios y despacho de recetas médicas, para dar fin a la problemática que enfrenta la Farmacéutica Fundación Salud para Todos.

B. JUSTIFICACIÓN

La Farmacéutica “Salud para Todos” se encuentra en una situación difícil debido a de ingresos causada por la pandemia de COVID-19. Para garantizar el control de recetas para los pacientes, es necesario contar con un sistema eficiente que permita llevar a cabo un control riguroso del inventario de medicamentos y el despacho de recetas médicas. Nuestro producto tecnológico busca satisfacer esta necesidad y maximizar los recursos financieros de la fundación.

C. OBJETIVOS

1. Crear el agendamiento de citas el cual es realizado manualmente por el personal de la fundación es deficiente, produciéndose una pérdida de información y confusión en las citas agendadas.

Facultad de Ingeniería y Ciencias

2. Llevar un sistema que les permita llevar un control estricto del inventario de medicinas, así como del despacho de recetas médicas a los pacientes.
3. Obtener un sistema contable que permita registrar los ingresos y egresos económicos de la unidad médica, dificultando el correcto seguimiento de los flujos de caja, lo cual a la larga puede causar el cierre de la unidad médica.
4. implementar a base de los objetivos generales realizar un sistema de inventario que lleve los ingresos de nuevos medicamentos o productos, alertas de cuando se vaya acabar su stock y su fecha de caducidad, así podremos actualizar diariamente y llevar un sistema de inventario autónomo y fácil

D. ANÁLISIS DEL PROBLEMA

En el análisis del problema voy a presentar 3 factores que llevan a una problemática siendo estos:

- Social: Al no tener un control de su inventario esto llegaría a tener un problema de desabastecimientos de medicinas esto afectaría a la empresa al no poder cumplir con una demanda de los pacientes y los pedidos de sus clientes
- Político: Al no tener un sistema de inventario podría no garantizar un sistema de transparencia y al no tener un registro de los ingresos y gastos, existe un de corrupción y malversación de fondos
- Económico: Hablando por el tema social en la parte económica afectaría a ganancias de la farmacéutica ya que perdería clientes ventas y asociaciones con hospitales o fundaciones al no poder tener las recetas para sus pacientes esto llevaría a una pérdida total

Facultad de Ingeniería y Ciencias

y en los peores de los casos estos productos al no ser controlados podrían los medicamentos tener caducidad y esto causaría demandas a la farmacéutica.

- Ambiental: Al no tener un sistema contable autónomo y llevar la contabilidad en un cuaderno o algún archivero, esto ocupa muchas hojas de papel y esto daña al medio ambiente ya que si muchos locales hicieran este tipo de inventario representa bastante a la tala de árboles para la elaboración de dichas hojas.

E. ESTABLECIMIENTO DE PROPUESTAS DE SOLUCIÓN

Para esto tendremos control en el sistema de inventario en el que se llevará a cabo las siguientes funciones:

- ✓ Se podrá egresar medicamentos nuevos
- ✓ Se actualizará el inventario diariamente para no tener escasez de medicamentos
- ✓ Se registrará el despacho de cada medicamento
- ✓ Generará alertas para los medicamentos cuando ya queden pocos
- ✓ Generará alertas de los medicamentos por caducar
- ✓ Y por último mostrará el inventario



Propuestas integrales de solución:



A.	<p>Implementar un sistema tecnológico que permita llevar un control del inventario de medicamentos, se actualizará, se podrá ver el despacho de los medicamentos también podremos saber qué medicamentos están por acabarse y por caducar y por último mostrará el inventario actualizado.</p>
B.	<p>Crear un sistema de código para llevar el control respectivo del inventario de medicamentos y de las salidas de recetas médicas, cada medicamento tendrá registro en el sistema.</p> <p>El personal encargado de despachar o ingresar las medicinas, ingresarán los datos y los medicamentos como su fecha de caducidad el despacho de medicamentos y también podrán ingresar nuevos medicamentos para tener por último paso un inventario actualizado.</p>

F. EVALUACIÓN DE PROPUESTAS DE SOLUCIÓN

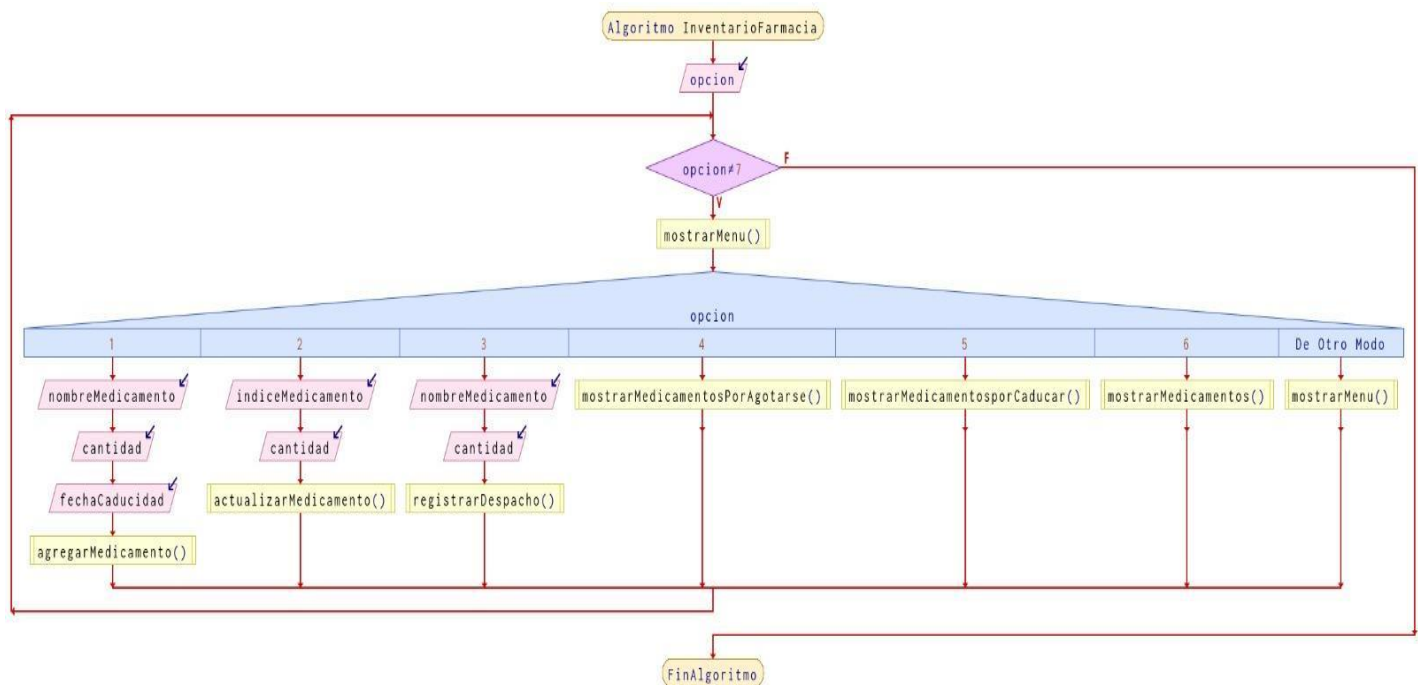
1. Tener un sistema de inventario manual y no tener un sistema automatizado. El cual tendríamos que detallar cada medicamento y su fecha de caducidad.
2. Tener un sistema de etiquetas de cada producto y una buena organización para saber el orden y donde se encuentra cada producto.
3. Tener un sistema de contabilidad automatizado el cual te permite tener un control de inventario, tener alertas de productos o medicinas escasas, tener el control del despacho de cada medicina, tener alertas por caducidad de cada producto y tener un inventario actualizado fácil y autónomo.

G. DEFINICIÓN DEL PLAN DE TRABAJO Y METODOLOGÍA

POR HACER	EN PROCESO	HECHO
Realización de main.c <ul style="list-style-type: none"> - Inclusión librerías - Declaración de variables - Función menú opciones:(Agregar, Actualizar, Registrar, Mostrar, Salir)) - Main(llamamos a otras funciones y realizamos una estructura con switch y depende de los casos imprimamos lo que indicamos y con algunos con la estructura while, indicamos si al introducir es menos se pausara y terminamos con return 0; - Cierre de main.c Alejandro Acosta 30/06/2023		 08/07/2023
Creación de inventario.c <ul style="list-style-type: none"> - Inclusión de librerías - Realizamos las funciones: - "Pausar": Detallamos lo que imprima y escanee - "limpiarPantalla" - Realización de estructuras de tipo if, else if, struct, de declaración de variables 		 08/07/2023

Michael Tipantuña 06/06/2023		
<p>Creación del menú</p> <ol style="list-style-type: none"> 1. Agregar medicina 2. Actualizar inventario 3. Registrar despacho 4. Generar alertas de pocos medicamentos 5. Generar alertas de medicamentos cerca de caducar 6. Mostrar el inventario 7. Salir <p>Alejandro Acosta 01/07/2023</p>		<div style="text-align: center;">  08/07/2023 </div>
<p>Realizar la presentación en canvas presentando partes del informe resumidas e importantes</p> <p>Michael Tipantuña 20/06/2023</p>		<div style="text-align: center;">  09/07/2023 </div>

H. Diseño de la propuesta de solución



Main.c

```
#include <stdlib.h>
#include <stdio.h>
#include "inventario.h"

void mostrarMenu() {
    printf("Menú:\n");
    printf("1. Agregar medicina\n");
    printf("2. Actualizar inventario\n");
    printf("3. Registrar despacho\n");
    printf("4. Mostrar medicamentos próximos a terminar\n");
    printf("5. Mostrar medicamentos próximos a caducar\n");
    printf("6. Mostrar inventario\n");
    printf("7. Salir\n");
    printf("Ingrese una opción: ");
}

int main() {
    int opcion;
    char nombre[50];
    int cantidad;
    char fecha_vencimiento[11];

    do {
        limpiarPantalla();
        mostrarMenu();
        scanf("%d", &opcion);
        switch (opcion) {
            case 1:
                printf("Ingrese el nombre de la medicina: ");
                scanf("%s", nombre);
                do {
                    printf("Ingrese la cantidad: ");
                    scanf("%d", &cantidad);
                    if(cantidad <= 0) {
                        printf("Ingrese un número válido.\n");
                    }
                } while (cantidad <= 0);
                do {
                    printf("Ingrese la fecha de vencimiento (dd/mm/yyyy): ");
                    scanf("%s", fecha_vencimiento);
                    if (!esFechaValida(fecha_vencimiento)) {
                        printf("Fecha de vencimiento inválida o caduca\n");
                    }
                } while (!esFechaValida(fecha_vencimiento));
                agregarMedicina(nombre, cantidad, fecha_vencimiento);
                pausar();
                break;
            case 2:
                mostrarInventario();
                printf("Ingrese el nombre de la medicina: ");
                scanf("%s", nombre);
                do {
                    printf("Ingrese la cantidad: ");
                    scanf("%d", &cantidad);
                    if(cantidad <= 0) {
                        printf("Ingrese un número válido.\n");
                    }
                } while (cantidad <= 0);
                actualizarInventario(nombre, cantidad);
                pausar();
                break;
            case 3:
                mostrarInventario();
                printf("Ingrese el nombre de la medicina: ");
                scanf("%s", nombre);
                do {
                    printf("Ingrese la cantidad: ");
                    scanf("%d", &cantidad);
                    if(cantidad <= 0) {
                        printf("Ingrese un número válido.\n");
                    }
                } while (cantidad <= 0);
                registrarDespacho(nombre, cantidad);
                pausar();
                break;
            case 4:
                generarAlertas();
                pausar();
                break;
            case 5:
                generarAlertasCaducidad();
                pausar();
                break;
            case 6:
                mostrarInventario();
                pausar();
                break;
            case 7:
                exit(0);
        }
    } while (opcion != 7);
}
```

```
default:
    printf("Opción inválida\n");
}
} while (opcion != 7);
return 0;
}
```

Este código nos permite que el usuario pueda gestionar el inventario de medicinas.

Muestra un menú con varias opciones para que el usuario pueda realizar diferentes acciones, como agregar medicinas al inventario, actualizar el inventario, registrar despachos, mostrar medicamentos próximos a terminar o caducar, y mostrar el inventario.

- El programa comienza con la inclusión de las bibliotecas necesarias y con el archivo llamado "inventario.h", en el que contiene las declaraciones de las funciones utilizadas para gestionar el inventario.
- Se define una función llamada "mostrarMenu" que nos muestra en pantalla y utilizando el printf para impresión de las opciones del menú.
- Realizamos el programa principal "main" en donde se guarda todos los archivos y se ejecuta.
- Definimos varias variables para almacenar información que va a ser ingresada y donde selecciona la opción, el nombre y cantidad de una medicina, y la fecha de vencimiento.
- Introducimos un bucle do-while en el que se debe ejecutar el mientras la opción seleccionada
- por el usuario no sea 7 (la opción para salir del programa). Y dentro de este bucle, se llama función "limpiarPantalla" (función para borrar el texto de la terminal), se muestra el menú utilizando la función "mostrarMenu" y se lee la opción seleccionada por el usuario utilizando la función scanf .
- Se utiliza una estructura switch para ejecutar diferentes acciones dependiendo de la

opción seleccionada.

- Si el usuario selecciona la opción 1: el programa pedirá que ingrese el nombre, cantidad y fecha de vencimiento de una medicina y luego llama a la función “agregarMedicina” para agregar esta información al inventario.
- Si el usuario selecciona la opción 2: el programa mostrará el inventario actual, utilizando la función “mostrarInventario” , y pedirá que ingrese el nombre y cantidad de una medicina y luego llama a la función actualizarInventario para actualizar la información en el inventario.
- Si el usuario selecciona la opción 3, el programa muestra el inventario actual, y pedira que ingrese el nombre y cantidad de una medicina y luego llama a la registrarDespacho para registrar un despacho de esta medicina.
- Si el usuario selecciona la opción 4, el programa llamará a la función generarAlertas para mostrar alertas sobre medicamentos que están por terminarse.
- Si el usuario selecciona la opción 5, el programa llamara a la función generarAlertasCaducidad para mostrar alertas sobre medicamentos próximos a llegar a su fecha de vencimiento(caduca).
- Si el usuario selecciona la opción 6, el programa mostrará el inventario actual utilizando la función mostrarinventario .
- Si el usuario selecciona la opción 7, el programa termina utilizando la función exit y si selecciona cualquier otra opción, se indicará como “opcion es inválida”.
- Después de ejecutar una acción, se llamará a la función pausar (para que el programa espere a que el usuario presione una tecla antes de continuar) y luego se vuelve al inicio del bucle para mostrar nuevamente el menú y espera otra opción.

Inventario.c

Agregar medicina

```
int esFechaValida(char *fecha) {
    int dia, mes, anio;
    if (sscanf(fecha, "%d/%d/%d", &dia, &mes, &anio) != 3) {
        return 0;
    }

    if (anio < 1900 || anio > 2099) {
        return 0;
    }

    if (mes < 1 || mes > 12) {
        return 0;
    }

    int dias_mes[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    if (anio % 400 == 0 || (anio % 100 != 0 && anio % 4 == 0)) {
        dias_mes[1] = 29;
    }

    if (dia < 1 || dia > dias_mes[mes - 1]) {
```

```
        return 0;
    }

    // Obtener la fecha actual del sistema
    time_t tiempo_actual = time(NULL);
    struct tm *fecha_actual = localtime(&tiempo_actual);
    int dia_actual = fecha_actual->tm_mday;
    int mes_actual = fecha_actual->tm_mon + 1;
    int anio_actual = fecha_actual->tm_year + 1900;

    if (anio < anio_actual) {
        return 0;
    } else if (anio == anio_actual) {
        if (mes < mes_actual) {
            return 0;
        } else if (mes == mes_actual) {
            if (dia <= dia_actual) {
                return 0;
            }
        }
    }

    return 1;
}

void agregarMedicina(char *nombre, int cantidad, char *fecha_vencimiento)
{
    FILE *archivo = fopen(ARCHIVO_INVENTARIO, "a");
    if (archivo == NULL) {
        printf("Error al abrir el archivo\n");
        return;
    }

    fprintf(archivo, "%s\t%d\t%s\n", nombre, cantidad, fecha_vencimiento);
    fclose(archivo);
}
```

- La función esFechaValida toma como argumento una cadena de caracteres en el que se representa una fecha en el formato "dd/mm/yyyy" y devuelve 1 si la fecha es válida y 0 en caso contrario.
- Una fecha es válida si cumple con las siguientes condiciones:
 - La fecha puede ser dividida en día, mes y año utilizando la función sscanf .
 - El año se realizó en el rango [1900, 2099].
 - El mes se realizó en el rango [1, 12].

Facultad de Ingeniería y Ciencias

- El día se realizó en el rango válido para el mes y año dados.
- La función verifica estas condiciones utilizando una estructura if-else y devuelve el resultado que le corresponde, para verificar si el día está en el rango válido para el mes y año dados, la función utiliza un arreglo que contiene el número de días en cada mes y ajusta el número de días en febrero si el año es bisiesto.
- Y si la fecha es posterior a la fecha actual del sistema, la función utiliza las funciones time y localtime para obtener la fecha actual del sistema y luego compara el año, mes y día de la fecha dada con los valores correspondientes de la fecha actual.
- La función “agregarMedicina” toma el nombre, la cantidad y la fecha de vencimiento de una medicina y agregar información al inventario y se realiza con una función, abriendo el archivo de inventario en modo "a" (para agregar información al final) y escribiendo en el nombre, la cantidad y la fecha de vencimiento de la medicina utilizando la función fprintf para que se cierre el archivo utilizamos una función fclose.

Actualizar Inventario

```
void actualizarInventario(char *nombre, int cantidad) {
    FILE *archivo = fopen(ARCHIVO_INVENTARIO, "r");
    if (archivo == NULL) {
        printf("Error al abrir el archivo\n");
        return;
    }

    FILE *temporal = fopen("temporal.txt", "w");
    if (temporal == NULL) {
        printf("Error al crear el archivo temporal\n");
        fclose(archivo);
        return;
    }

    char linea[100];
    while (fgets(linea, 100, archivo) != NULL) {
        char nombre_medicina[50];
        int cantidad_medicina;
        char fecha_vencimiento[11];

        sscanf(linea, "%s %d %s", nombre_medicina, &cantidad_medicina,
            fecha_vencimiento);

        if (strcmp(nombre_medicina, nombre) == 0) {
            fprintf(temporal, "%s\t%d\t%s\n", nombre_medicina, cantidad,
                fecha_vencimiento);
        } else {
            fprintf(temporal, "%s", linea);
        }
    }

    fclose(archivo);
    fclose(temporal);

    remove(ARCHIVO_INVENTARIO);
    rename("temporal.txt", ARCHIVO_INVENTARIO);
}
```

Facultad de Ingeniería y Ciencias

- La función “actualizar Inventario” toma como argumentos el nombre y la cantidad de una medicina y actualiza esta información en el inventario, la función realiza el abrir el archivo del inventario en modo "r" (para leer su contenido) y creando un archivo temporal en modo "w" (para escribir en él). Luego, lee el contenido del archivo de inventario línea por línea utilizando la función fgets y para cada línea utiliza la función sscanf para extraer el nombre, la cantidad y la fecha de vencimiento de la medicina.
- Si el nombre de la medicina en la línea leída coincide con el nombre pasado como argumento a la función, se escribe en el archivo temporal el nombre, la nueva cantidad (pasada como argumento a la función) y la fecha de vencimiento de la medicina utilizando la función fprintf, si el nombre no coincide, se escribe en el archivo temporal la línea leída sin modificar.
- Después de procesar todas las líneas del archivo de inventario, se cierran los archivos utilizando la función fclose. Luego se elimina el archivo de inventario original con la función remove y se renombra el archivo temporal con el nombre del archivo de inventario utilizando la función rename y reemplaza al archivo de inventario original con la información actualizada.
- Si ocurre un error al abrir el archivo de inventario o al crear el archivo temporal (si el archivo no existe o no se tiene permiso para escribir en él), la función mostrará un mensaje de error utilizando la función printf y retorna sin realizar ninguna acción.

Registrar Despacho

[illegible]

- La función "registrarDespacho" toma como argumentos el nombre y cantidad de una medicina y registra un despacho de esta medicina en el inventario, la función realiza el abrir el archivo de inventario en modo "r" (para leer su contenido) y creando un archivo temporal en modo "w" (para escribir en él).Luego, lee el contenido del archivo de inventario línea por línea utilizando la función fgets y para cada línea utiliza la función sscanf para extraer el nombre, la cantidad y la fecha de vencimiento de la medicina.
- Si el nombre de la medicina en la línea leída corresponde con el nombre pasado como argumento a la función, se verificará si hay suficiente cantidad en el inventario para realizar el despacho, si no hay suficiente cantidad, se muestra un mensaje de error utilizando la función printf y se cierran ambos archivos utilizando la función fclose , se elimina el archivo temporal utilizando la función remove y se retorna sin realizar ninguna acción o si hay suficiente cantidad, se escribe en el archivo temporal el nombre, la nueva cantidad (calculada restando la cantidad despachada a la cantidad original) y la fecha de vencimiento de la medicina utilizando la función fprintf . Si el nombre no coincide, se

escribe en el archivo temporal la línea leída sin modificar.

- Todas las líneas del archivo de inventario, se verifica si se encontró la medicina en el inventario, en el caso que no se encuentre, se mostrará un mensaje indicando que no se encontró la medicina utilizando la función printf y se cierran ambos archivos utilizando la función fclose , eliminando el archivo de inventario original utilizando la función remove y se renombra el archivo temporal con el nombre del archivo de inventario utilizando la función rename.
- Si ocurre algún error al abrir el archivo de inventario o al crear el archivo temporal (el archivo no existe o no se tiene permiso para escribir en él), la función muestra un mensaje de error utilizando la función printf y retorna sin realizar ninguna acción.

Generar Alertas

```
void generarAlertas() {
    FILE *archivo = fopen(ARCHIVO_INVENTARIO, "r");
    if (archivo == NULL) {
        printf("Error al abrir el archivo\n");
        return;
    }

    printf("Alertas:\n");
    char linea[100];
    while (fgets(linea, 100, archivo) != NULL) {
        char nombre_medicina[50];
        int cantidad_medicina;

        char fecha_vencimiento[11];

        sscanf(linea, "%s %d %s", nombre_medicina, &cantidad_medicina,
            fecha_vencimiento);

        if (cantidad_medicina < NIVEL_MINIMO) {
            printf("La medicina %s tiene una cantidad por debajo del nivel mínimo\n", nombre_medicina);
        }

        fclose(archivo);
    }
}
```

- La función “generar Alertas” no toma argumentos y muestra alertas sobre medicamentos cuya cantidad en el inventario está por debajo del nivel mínimo definido

Facultad de Ingeniería y Ciencias

por la constante NIVEL_MINIMO, la función hace esto abriendo el archivo de inventario en modo "r" (para leer) y leyendo su contenido línea por línea utilizando la función fgets, y cada línea utiliza la función sscanf para extraer el nombre, la cantidad y la fecha de vencimiento de la medicina.

- La cantidad de la medicina es menor que el nivel mínimo definido por NIVEL_MINIMO, se mostrará una alerta utilizando la función printf indicando que la medicina tiene una cantidad por debajo del nivel mínimo. Y tendrá que procesar todas las líneas del archivo de inventario y cerrar el archivo utilizando la función fclose .
- Si ocurre un error al abrir el archivo de inventario (el archivo no existe o no tiene permiso para leerlo), la función muestra un mensaje de error utilizando la función printf y retorna sin realizar ninguna acción.

Generar Alertas Caducidad

```
int esBisiesto(int anio) {
    if (anio % 400 == 0) {
        return 1;
    } else if (anio % 100 == 0) {
        return 0;
    } else if (anio % 4 == 0) {
        return 1;
    } else {
        return 0;
    }
}

int diasEntreFechas(int dia1, int mes1, int anio1, int dia2, int mes2, int
anio2) {
```

```
    int dias_mes[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

    if (dia1 > dias_mes[mes1 - 1]) {
        dia1 = dias_mes[mes1 - 1];
    }

    if (dia2 > dias_mes[mes2 - 1]) {
        dia2 = dias_mes[mes2 - 1];
    }

    long int n1 = anio1 * 365 + dia1;
    for (int i = 0; i < mes1 - 1; i++) {
        n1 += dias_mes[i];
    }
    n1 += esBisiesto(anio1) && mes1 > 2;

    long int n2 = anio2 * 365 + dia2;
    for (int i = 0; i < mes2 - 1; i++) {
        n2 += dias_mes[i];
    }
    n2 += esBisiesto(anio2) && mes2 > 2;

    return abs(n2 - n1);
}

void generarAlertasCaducidad() {
    FILE *archivo = fopen(ARCHIVO_INVENTARIO, "r");
    if (archivo == NULL) {
        printf("Error al abrir el archivo\n");
        return;
    }

    time_t t = time(NULL);
    struct tm tm = *localtime(&t);
    int dia_actual = tm.tm_mday;
    int mes_actual = tm.tm_mon + 1;
    int anio_actual = tm.tm_year + 1900;

    printf("Alertas de caducidad:\n");
    char linea[100];
    while (fgets(linea, 100, archivo) != NULL) {
        char nombre_medicina[50];
        int cantidad_medicina;
        char fecha_vencimiento[11];

        sscanf(linea, "%s %d %s", nombre_medicina, &cantidad_medicina,
fecha_vencimiento);

        int dia_vencimiento, mes_vencimiento, anio_vencimiento;
```

```
        sscanf(fecha_vencimiento, "%d/%d/%d", &dia_vencimiento,
&mes_vencimiento, &anio_vencimiento);

        int dias_restantes = diasEntreFechas(dia_actual, mes_actual,
anio_actual, dia_vencimiento, mes_vencimiento, anio_vencimiento);
        if (dias_restantes <= DIAS_PARA_CADUCAR) {
            printf("La medicina %s está a punto de caducar (%d días
restantes)\n", nombre_medicina, dias_restantes);
        }
    }

    fclose(archivo);
}
```

- Estas funciones permiten verificar si un año es bisiesto, el como calcular el número de días entre dos fechas y mostrar alertas sobre medicamentos cuya fecha de vencimiento está próxima, con la función es Bisiesto toma como argumento un año y devuelve 1 si el año es bisiesto y 0 en caso contrario.
- En este caso se observa si el año es bisiesto, si es divisible entre 400 o si es divisible entre 4 pero no entre 100.

Facultad de Ingeniería y Ciencias

- La función utiliza una estructura if-else para verificar estas condiciones y devolver el resultado que le indicamos.
- La función días “EntreFechas” toma las dos fechas como argumento (representada por un día, mes y año) y nos devuelve el número de días entre estas dos fechas y la función realizará el calcular el número total de días transcurridos desde una fecha de referencia (por ejemplo, el 1 de enero del año 1) hasta cada una de las dos fechas y luego posteriormente restando estos valores para obtener el número de días entre las dos fechas.
- Para realizar el cálculo de un número total de días transcurridos desde la fecha de referencia hasta
- una fecha dada:
 - La función multiplica el año por 365 (para obtención del número de días en los años completos) y sumar el día del mes.
 - Después se realiza la suma del número de días en los meses completos utilizando un arreglo que contiene el número de días en cada mes.
 - Si el mes es mayor que 2 (es decir, si febrero ya pasó) y el año es bisiesto, se suma un día adicional para tener en cuenta el día extra en febrero.
 - Finalmente se restará el valor calculado para la primera fecha del valor calculado para la segunda fecha y se devuelve el valor absoluto del resultado.
- La función “generarAlertasCaducidad” no toma argumentos y muestra alertas sobre medicamentos cuya fecha de vencimiento está próxima y la función hace el abrir el archivo de inventario en modo "r" (para leer), obteniendo la fecha actual del sistema utilizando las funciones time y localtime , y leyendo el contenido del archivo de

inventario línea por línea utilizando la función `fgets` .y con la utilización de función `sscanf` para extraer el nombre, la cantidad y la fecha de vencimiento de la medicina.

- Utiliza nuevamente la función `sscanf` para extraer el día, mes y año de la fecha de vencimiento y llama a la función “`diasEntreFechas`” para calcular el número de días entre la fecha actual y la fecha de vencimiento y con la condición si este número es menor o igual que el valor definido por la constante `DIAS_PARA_CADUCAR` se mostrará una alerta utilizando la función `printf` indicando que la medicina está a punto de caducar.
- Después de procesar todas las líneas del archivo de inventario con las funciones dichas anteriormente, se cierra el archivo utilizando la función `fclose` .
- Si ocurre un error al abrir el archivo de inventario (el archivo no existe o no se tiene permiso para leerlo), la función muestra un mensaje de error utilizando la función `printf` y retorna sin realizar ninguna acción.

Mostrar Inventario

```
void mostrarInventario() {
    FILE *archivo = fopen(ARCHIVO_INVENTARIO, "r");
    if (archivo == NULL) {
        printf("Error al abrir el archivo\n");
        return;
    }
    printf("Inventario:\n");
    char linea[100];
    while (fgets(linea, 100, archivo) != NULL) {
        printf("%s", linea);
    }
    fclose(archivo);
}
```

- La función “`mostrarInventario`” no toma argumentos y muestra el contenido del inventario y la función hace esto abriendo el archivo de inventario en modo “`r`” (para

Facultad de Ingeniería y Ciencias

leer su contenido) y leyendo su contenido línea por línea utilizando la función fgets y

para cada línea vamos a utilizar la función printf para mostrar la línea en pantalla

- Después de procesar todas las líneas del archivo de inventario, se cierra el archivo utilizando la función fclose .
- Si ocurre un error al abrir el archivo de inventario (el archivo no existe o no se tiene permiso para leerlo), la función muestra un mensaje de error utilizando la función printf y retorna sin realizar ninguna acción.

I. EJECUCIÓN DE ESCENARIOS DE PRUEBA

ESCENARIO 1

1. En el escenario uno ejecutar el sistema de inventario y encontrar con un menú que obtenga:

Agregar medicina

Actualizar inventario

Registrar despacho

Generar alertas de pocos medicamentos

Generar alertas de medicamentos cerca de caducar

Mostrar el inventario

Salir

ESCENARIO 2

2. Este escenario vamos a poder ingresar nuevos medicamentos más los que ya están inventariados y si en un agregamos el mismo producto se actualiza el valor en lugar de

crear un nuevo medicamento, salvo que las fechas de caducidad sean diferentes

ESCENARIO 3

3. En este escenario podremos ver la actualización del inventario y nos enseñara las medicinas que están en stock con su fecha de caducidad.

ESCENARIO 4

4. Aquí podremos registrar el despacho de cada medicamento y se del stock inventariado automáticamente.

ESCENARIO 5

5. Aplicando la opción 4 del menú nos saltará una alerta de que medicamentos están por acabarse así podremos llamar a un proveedor para no tener escasez de productos

ESCENARIO 6

6. Gracias al introducir su medicamento o producto con su fecha de caducidad al presionar la opción 5 nos indicara otro aviso informándonos que productos están por caducar

ESCENARIO 7

7. Antes de finalizar en la opción 6 nos indicará el inventario de todos los productos y medicamentos actualizados.

ESCENARIO 8

8. Acabando todo el proceso presionaremos opción 7 esto nos dará una salida del sistema de inventario

J. CONCLUSIONES Y RECOMENDACIONES

Recomendaciones:

- Mantener una supervisión constante del sistema de inventario y realizar actualizaciones para asegurar su funcionalidad de manera eficiente y adaptación a las necesidades cambiantes de la Farmacéutica “Salud para Todos”.
- Implementar cursos para capacitar al personal encargado con el fin de dar un buen manejo del programa, de tal manera poder aprovechar al máximo este sistema.

Conclusiones:

- Finalmente hemos podido concluir que la implementación de un sistema tecnológico de control de inventario y salida de recetarios médicos ha probado ser una solución segura para dar solución a la problemática que enfrenta la Farmacéutica “Salud para Todos” en cuanto a la escasez de ingresos.
- En conclusión, desarrollar el sistema de gestión de inventario, brindará y será una herramienta efectiva. Sin embargo, es fundamental seguir las asegurar el éxito del sistema.

K. ANEXOS:

<https://view.genial.ly/64ab91260605bd001273672f/presentation-proyecto-c-abet>

<https://github.com/Maik2101/Abet>

L. BIBLIOGRAFÍAS:

Cómo utilizar genially:

<https://genial.ly/es/primeros-pasos/>

La importancia de tener un inventario actualizado:

<https://www.holded.com/es/blog/la-importancia-de-tener-un-inventario-actualizado>

La autonomía en los emprendimientos:

<https://www.revistalideres.ec/lideres/autonomia-organizacion-empresa-trabajadores-productividad.html#:~:text=La%20autonomía%20es%20un%20factor%20clave%20del%20éxito%20de%20las%20empresas&text=La%20autonomía%20es%20una%20cualidad,productividad%20y%20al%20desarrollo%20organizacional>.

Prácticas para aprender programación en C desde 0 hasta un nivel avanzado:

<https://udla.udemy.com/course/aprende-c-con-ejercicios-practicos/learn/lecture/30903798#overview>

