Informe trabajo final integrador

Grupo integrado por:

- Juan Ignacio García
- Julián García
- Diego Gómez

Metodología de trabajo

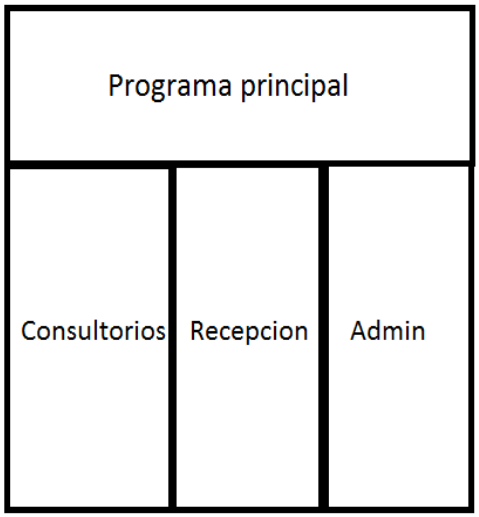
Durante las fechas anteriores a la defensa del trabajo realizamos una serie de video llamadas por la plataforma Google Meet. Realizando una coordinación de horarios con nuestro compañero que se encuentra de viaje en el exterior. Trabajamos de manera conjunta cada uno de los módulos y nos tocó decidir entre 2 implementaciones de diseños distintos para la realización correcta del trabajo.

Las cuales eran:

1. Un programa principal con todas las funciones haciendo llamadas a 3 librerías distintas.
2. Un programa principal conectado con 3 archivos .exe que generaran los documentos correspondientes y la interacción con el usuario.

Después de un análisis concreto decidimos optar por la 2da usando también el diseño top down y la simplificación de funciones haciendo uso de librerías externas para evitar las fallas en el programa y optimizar la velocidad del mismo. Hemos diseñado una aplicación integral dedicada a la administración de turnos, clientes y profesionales de un hospital. El trabajo cuenta con un menú que facilita el acceso a los 3 módulos principales, encargados de gestionar todos los aspectos de la aplicación mediante registros y archivos.

Para este proyecto, hemos optado por un diseño de interfaz minimalista, aprovechando un prototipo de menú que ya hemos implementado en ejercicios previos realizados en los trabajos prácticos proporcionados por la cátedra.




```
printf("\n1. Modulo Administracion.");
printf("\n2. Modulo Recepcion.");
printf("\n3. Modulo Consulta");
printf("\n4. SALIR.");
```

Modulo administración

El módulo de administración se encarga de gestionar los usuarios encargados de la administración del sanatorio. Permite registrar nuevos usuarios recepcionistas, validando sus datos de usuario y contraseña según criterios predefinidos. Además, ofrece funcionalidades para el registro de profesionales médicos, atenciones por parte de dichos profesionales y generación de un ranking de profesionales según la carga horaria.

Utiliza estructuras de datos para almacenar la información de los usuarios y médicos, y funciones para validar nombres de usuario y contraseñas. El menú principal ofrece opciones para realizar acciones específicas, como registrar usuarios y médicos, programar citas y generar informes.

Modulo recepción

El código es un programa en C para gestionar un sistema de registro de pacientes en un centro médico. A continuación detallamos las funciones:

Inicio de Sesión:

- Solicita un nombre de usuario y una contraseña.
- Verifica si cumplen ciertos requisitos de validez.

Menú Principal:

- Permite registrar pacientes, programar consultas y listar pacientes.

Funciones Principales:

- registrarPacientes(): Guarda los datos de un nuevo paciente.
- listarPacientes(): Muestra la lista de pacientes registrados.
- registrarConsulta(): Programa una consulta médica para un paciente.

Almacenamiento:

- Los datos de pacientes se guardan en un archivo llamado "pacientes.dat".
- Los datos de médicos se guardan en un archivo llamado "medicos.dat".

Modulo consultorios

Utiliza conceptos como estructuras, funciones y manipulación de archivos para permitir a los usuarios administrar datos relacionados con pacientes y sus historias clínicas.

En términos de estructura, el programa se divide en varias secciones claramente definidas. La primera sección contiene las declaraciones de las estructuras fecha, pacientes y historia_clinica, que definen la información relevante sobre las fechas, los pacientes y sus historias clínicas respectivamente. Estas estructuras proporcionan un marco organizado para almacenar y manipular los datos del consultorio médico de manera eficiente.

La siguiente sección del programa incluye la definición de funciones que operan en los datos de los pacientes y sus historias clínicas. Por ejemplo, la función listarPacientes permite visualizar la información de los pacientes almacenada en un archivo, mientras que la función registrarHistoriaClinica facilita la creación de nuevas entradas de historias clínicas para los pacientes. Estas funciones están diseñadas para ser modulares y reutilizables, lo que promueve un diseño de código limpio y mantenible.

Además, el programa incorpora funciones para validar nombres de usuario y contraseñas al iniciar sesión. Estas funciones, validarNombreUsuario y validarContrasena, aplican criterios específicos para garantizar que las credenciales proporcionadas cumplan con ciertos estándares de seguridad. Por ejemplo, se verifica que el nombre de usuario tenga una longitud adecuada y comience con una letra minúscula, mientras que la contraseña debe contener al menos una letra mayúscula, una letra minúscula, un número y cumplir con otras condiciones específicas.

La parte principal del programa se encuentra en la función main, que sirve como punto de entrada y coordinador central de las operaciones del sistema. Aquí, se presenta un menú interactivo que permite a los usuarios seleccionar diferentes acciones, como listar pacientes, registrar nuevas historias clínicas o volver al menú principal. Esta estructura de menú proporciona una interfaz intuitiva para que los usuarios interactúen con el sistema de manera efectiva y eficiente.

Validación

Para la función de validación usamos una estructura bifuncional, donde tenemos la función principal que llama a las demás para hacer la validación de manera correcta.

Función validarUsuario:

- *Requisitos de longitud:* Verifica si el nombre de usuario tiene una longitud mínima de 6 caracteres y una longitud máxima de 10 caracteres. Si no cumple con estos requisitos, devuelve un código de error correspondiente.
- *Unicidad:* Comprueba si el nombre de usuario es único entre una lista de usuarios. Si encuentra una coincidencia, indica que el nombre de usuario no es único y devuelve un código de error.
- *Formato de la primera letra:* Asegura que la primera letra del nombre de usuario sea minúscula. Si no lo es, devuelve un código de error.
- *Número de mayúsculas y dígitos:* Verifica que el nombre de usuario tenga al menos 2 letras mayúsculas y no más de 3 dígitos. Si no cumple con estas condiciones, devuelve un código de error correspondiente.

Función validarContrasena:

- *Requisitos de longitud*: Valida que la contraseña tenga una longitud mínima de 6 caracteres y una longitud máxima de 10 caracteres. Si no cumple con estos requisitos, devuelve un código de error.

- *Composición de la contraseña*: Verifica que la contraseña contenga al menos una letra mayúscula, una letra minúscula y un dígito. Si no cumple con estos criterios, devuelve un código de error.