**Funcionalidades**

1. Añade un local al piso que indique el usuario.

• Entradas: piso que indique el usuario y local que indique el usuario que no esté ocupado.

• Salidas: devuelve los datos ingresados por el usuario para el local y estos se asignan a la matriz que es dinámica, se agrega al archivo el nuevo usuario.

• Conceptos usados: interacción con la matriz dinámica, enums, archivos.

1. Busca un local por el nombre

• Entradas: Escribe el nombre del local

• Salidas: imprime el local que busca el usuario y los datos de este.

• Conceptos: interacción con la matriz dinámica

1. Busca un local por su respectivo id

• Entradas: Escribe el id del local a buscar

• Salidas: imprime el local que busca el usuario y los datos de este.

• Conceptos usados: interacción con la matriz dinámica

1. Elimina los locales de un piso

• Entradas: El usuario digita el piso al cual se eliminarán los locales o ponerlos en disponibles.

• Salidas: retorna 0 o NULL en los espacios donde había locales estando eliminados o disponibles, se elimina el usuario en el archivo de centro comercial.

• Conceptos usados: Recursión en la eliminación de cada local, archivos.

1. Actualiza los datos de un local

• Entradas: Pide al usuario digitar el piso y el número de local a actualizar.

• Salidas: Actualiza los datos anteriores con los datos nuevos ingresados por el usuario, se actualiza el usuario en el archivo de centro comercial.

• Conceptos usados: interacción con la matriz dinámica, archivos

1. Información centro comercial

• Entradas: Pide al usuario que digite una opción del menú que se le muestra información básica como de ¿qué es un centro comercial?

• Salidas: un mensaje dependiendo de la opción elegida por el usuario.

• Conceptos usados: Enums

1. Organizar por id de local
   * Entradas: El usuario digita el piso al cual se le organizaran los locales por número de id, la función principal recibe el número de pisos, número de locales por piso y la matriz del centro comercial.
   * Salidas: Organizar la matriz imprimiendo así el nombre del local y los id de los locales organizados escribe en archivo de centro comercial el array de struct organizado.
   * Conceptos usados: Método de ordenamiento por selección y arrays de structs, archivos.
2. Organizar por número de local
   * Entradas: El usuario digita el piso al cual se le organizaran los locales por número de numero de local, la función principal recibe el número de pisos, número de locales por piso y la matriz del centro comercial.
   * Salidas: Organizar la matriz imprimiendo así el nombre del local y los números de local organizados, escribe en archivo de centro comercial el array de struct organizado.
   * Conceptos usados: Método de ordenamiento por inserción y arrays de structs, archivos.
3. Organizar por cantidad de productos por local
   * Entradas: El usuario digita el piso al cual se le organizaran los locales por número de numero de local, la función principal recibe el número de pisos, número de locales por piso y la matriz del centro comercial.
   * Salidas: Organizar la matriz imprimiendo así el nombre del local y la cantidad de productos por local organizados, escribe en el archivo de centro comercial el array de struct organizado.
   * Conceptos usados: Método de ordenamiento por quick sort y arrays de structs, archivos.
4. Organizar por cantidad de productos por local
   * Entradas: El usuario digita el piso al cual se le organizaran los locales por número de numero de local, la función principal recibe el número de pisos, número de locales por piso y la matriz del centro comercial.
   * Salidas: Organizar la matriz imprimiendo así el nombre del local y la cantidad de productos por local organizados, escribe en el archivo de centro comercial el array de struct organizado.
   * Conceptos usados: Método de ordenamiento por merge sort y arrays de structs, archivos.
5. Salir es aquella opción donde te permite salir del do-while es decir del ciclo.