PONTIFICIA UNIVERSIDAD JAVERIANA DE CALI

BRENDA DAYANA TORRES MURCIA

SEGUNDO SEMESTRE

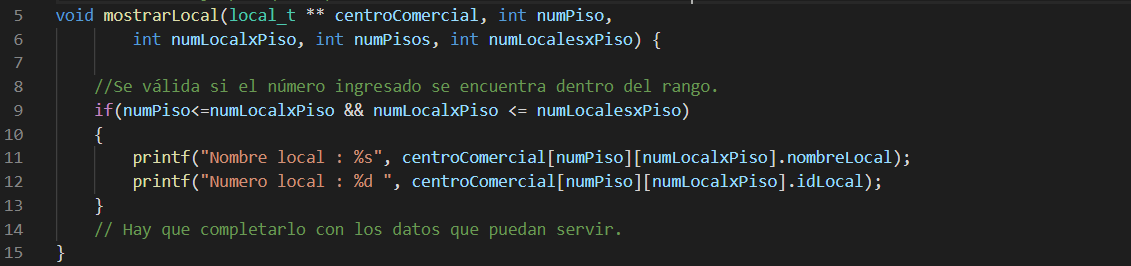
INGENIERÍA EN SISTEMAS Y COMPUTACIÓN

TÉCNICAS Y PRÁCTICAS DE PROGRAMACIÓN

LUISA FERNANDA

2020

**EXPLICACIÓN:**

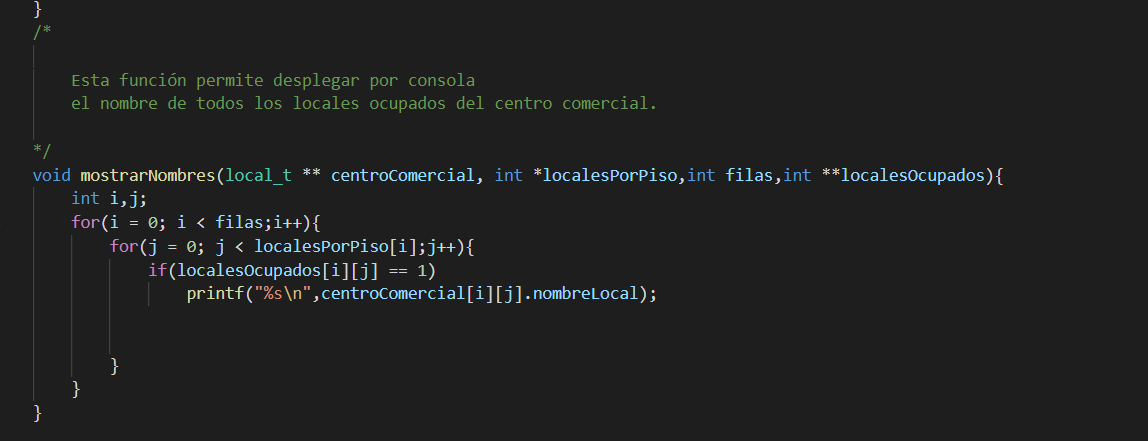


FUNCIÓN MOSTRARLOCAL:

Entradas: Se recibe el número de local por piso, número de pisos y números de locales por pisos para poder acomodarlos para luego mostrarse.

Salidad: Muestra el nombre y el número del local en el que se encuentra situado.

Conceptos usados: structs, matrices, ciclos, funciones, “punteros” y condicionales.

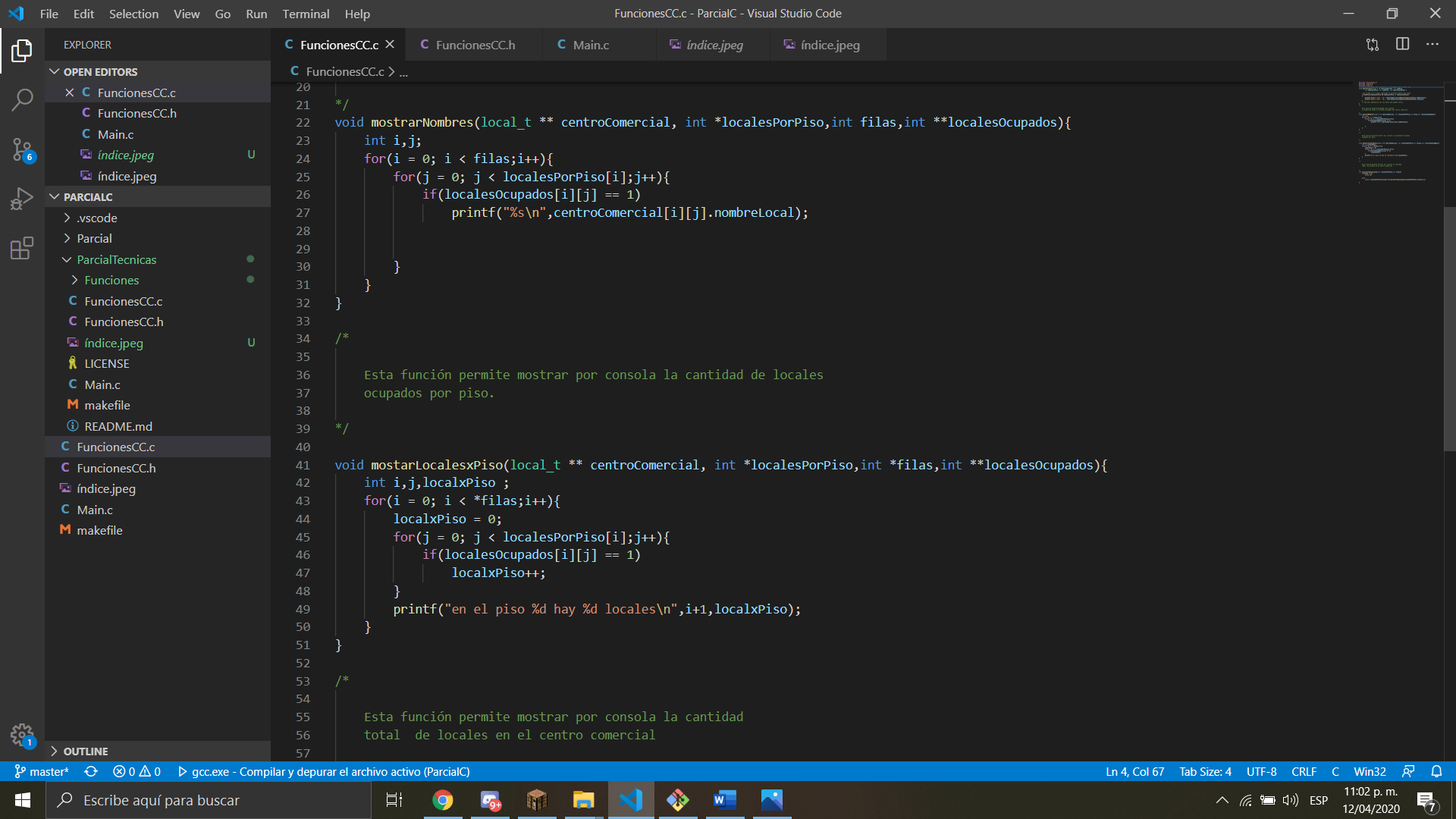


FUNCIÓN MOSTRARNOMBRES:

Entradas: Se recibe los datos de la struct [Matriz], los locales por piso, las filas y los locales ocupados, lo que realiza es un ciclo para recorrerlo y saber que locales hay dentro de la matriz del centro comercial.

Salidad: Muestra la posición del local y el nombre del local.

Conceptos usados: structs, matrices, ciclos, funciones, “punteros” y condicionales.

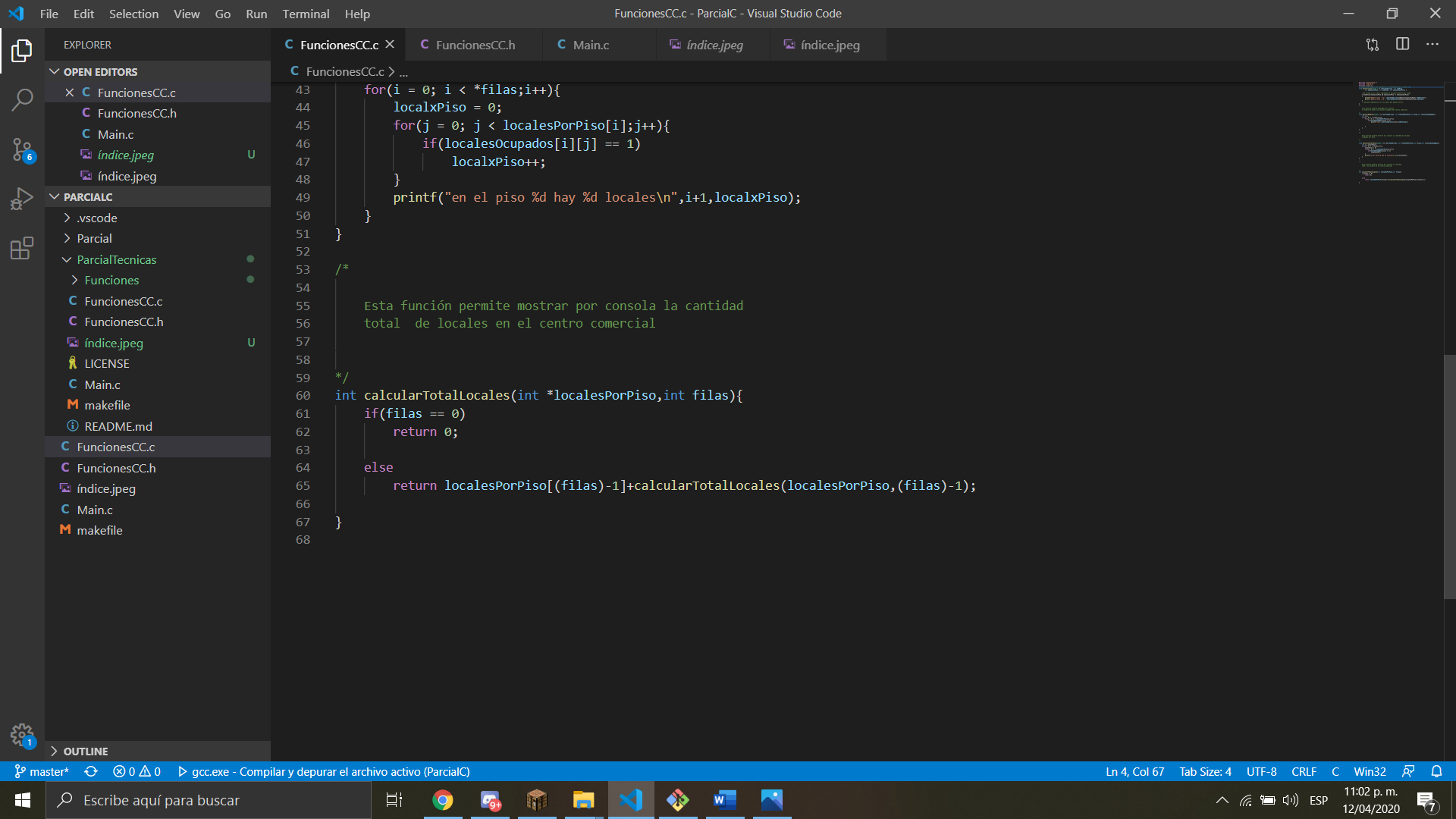


FUNCIÓN MOSTRARLOCALESXPISO:

Entradas: Se recibe los datos de la struct [Matriz], los locales por piso, las filas y los locales ocupados, lo que realiza es un ciclo para recorrerlo y saber que locales hay.

Salidad: Muestra el piso y el local que hay en estos y se le suma uno para

Conceptos usados: structs, matrices, ciclos, funciones, “punteros”.



FUNCIÓN CALCULARTOTALLOCALES:

Entradas: Se recibe los locales por piso y las filas, si hay cero retornará que no hay pero en caso de haber con locales por piso que se le pide al usuario, se le irá restando hasta no tener más.

Salidad: Arroja la cantidad total de locales.

Conceptos usados: structs, matrices, condicionales, funciones.

enum Bool {False,True};  // Lo uso en el arreglo locales ocupados para marcar como True o False.

Coloqué un enum bool, que sería un booleano de verdadero y falso, para luego utilizarlo en locales ocupados, que marqué si está o no ocupado.

case 1:

                printf("Ingrese la cantidad de pisos del centro comercial\n");

                localesPorPiso = malloc(filas\*sizeof(int));

                centroComercial = malloc(filas\* sizeof(local\_t \*));

                localesOcupados = malloc(filas\*sizeof(int \*));

                scanf("%d",&filas);

                for(i = 0; i < filas;i++){

                    printf("Ingrese los locales disponibles en el piso %d\n",i+1);

                    scanf("%d",&columna);

                    localesPorPiso[i] = columna;

                    centroComercial[i] = malloc(columna\*sizeof(local\_t));

                    localesOcupados[i] = malloc(columna\*sizeof(int));

                    for(j = 0; j < columna;j++)

                        localesOcupados[i][j] = False;

                }

                break;

Caso 1: Asignar la cantidad de pisos y locales dentro del centro comercial.

Entradas: Se pide al usuario que ingrese la cantidad de pisos del centro comercial para luego asignarles la memoria con el malloc y junto al sizeof devolver el tamaño, acomoda dentro de la struct este valor que introdujo, luego se le pide que ingrese la cantidad de locales disponibles en el piso que anteriormente agregó y se le agrega a las columnas, dependiendo de esas columnas lo que hace es acomodarlo en la matriz de structs que se realizó anteriormente.

Salidas: Cantidad de pisos del centro comercial y la cantidad de los locales disponibles en esos pisos.

Temas Aplicados: Switch, ciclos, matrices, funciones y structs.

case 2:

                printf("Ingrese el piso donde quiere colocar el local\n");

                scanf("%d",&fila);

                while(fila < 0 || fila >= filas){

                    printf("Ingrese una fila entre 0 y %d\n",filas);

                    scanf("%d",&fila);

                }

                printf("Ingrese el local donde quiere poner el local comercial\n");

                scanf("%d",&columna);

                while(columna < 0 || columna > localesPorPiso[fila]){

                    printf("Ingrese una columna entre 0 y %d\n",localesPorPiso[fila]);

                    scanf("%d",&columna);

                }

                while(localesOcupados[fila][columna] == 1){

                    printf("Ingrese otro local ya que este esta ocupado\n");

                    printf("Ingrese el piso donde quiere colocar el local\n");

                    scanf("%d",&fila);

                    while(fila < 0 || fila >= filas){

                        printf("Ingrese una fila entre 0 y %d\n",filas);

                        scanf("%d",&fila);

                    }

                    printf("Ingrese el local donde quiere poner el local comercial\n");

                    scanf("%d",&columna);

                    while(columna < 0 || columna > localesPorPiso[fila]){

                        printf("Ingrese una columna entre 0 y %d\n",localesPorPiso[fila]);

                        scanf("%d",&columna);

                    }

                }

                localesOcupados[fila][columna] = True;

                printf("Ingrese el nombre del local\n");

                scanf("%s",centroComercial[fila][columna].nombreLocal);

                printf("%s\n",centroComercial[fila][columna].nombreLocal);

                centroComercial[fila][columna].idLocal = ids++;

                centroComercial[fila][columna].pisoLocal = fila;

                centroComercial[fila][columna].numLocalxPiso = columna;

                break;

Caso 2: Ingresar un nuevo local al centro comercial.

Entradas: Se pide al usuario que ingrese la cantidad de pisos del centro comercial para luego asignarles la memoria con el malloc y junto al sizeof devolver el tamaño, acomoda dentro de la struct este valor que introdujo, luego se le pide que ingrese la cantidad de locales disponibles en el piso que anteriormente agregó y se le agrega a las columnas, dependiendo de esas columnas lo que hace es acomodarlo en la matriz de structs que se realizó anteriormente, lo que se realiza es ir acomodando cada valor de los que ha ingresado en la matriz ya hecha.

Salidas: Nuevo local con su ubicación.

Conceptos usados: structs, matrices, ciclos, funciones, “punteros”.

  case 3:

                mostrarNombres(centroComercial, localesPorPiso, filas,localesOcupados);

                break;

Caso tres se utiliza la función que anteriormente expliqué.

case 4:

                mostarLocalesxPiso(centroComercial, localesPorPiso, &filas,localesOcupados);

                break;

Caso cuatro se utiliza la función que anteriormente expliqué.

case 5:

                printf("Ingrese el piso del local que quiere cambiar el nombre\n");

                scanf("%d",&fila);

                while(fila < 0 || fila >= filas){

                    printf("Ingrese una fila entre 0 y %d\n",filas);

                    scanf("%d",&fila);

                }

                printf("Ingrese la columna del local que quiere cambiar el nombre\n");

                scanf("%d",&columna);

                while(columna < 0 || columna > localesPorPiso[fila]){

                    printf("Ingrese una columna entre 0 y %d\n",localesPorPiso[fila]);

                    scanf("%d",&columna);

                }

                while(localesOcupados[fila][columna] == 0){

                    printf("Ingrese otro local que este ocupado\n");

                    printf("Ingrese el piso del local que quiere cambiar el nombre\n");

                    scanf("%d",&fila);

                    while(fila < 0 || fila >= filas){

                        printf("Ingrese una fila entre 0 y %d\n",filas);

                        scanf("%d",&fila);

                    }

                    printf("Ingrese la columna del local que quiere cambiar el nombre\n");

                    scanf("%d",&columna);

                    while(columna < 0 || columna > localesPorPiso[fila]){

                        printf("Ingrese una columna entre 0 y %d\n",localesPorPiso[fila]);

                        scanf("%d",&columna);

                    }

                }

                printf("nombre viejo %s\n",centroComercial[fila][columna].nombreLocal);

                printf("Ingrese nuevo nombre\n");

                scanf("%s",centroComercial[fila][columna].nombreLocal);

                break;

Entradas: Recibe el piso y la columna del local que se va a cambiar y con estos datos verificar si existe tal local para luego mostrar el nombre viejo y pedir el nombre nuevo que agregará.

Salidas: Nombre nuevo del local.

Conceptos usados: switch, structs, matrices, ciclos y funciones.

case 6:

                printf("Ingrese el piso del local que quiere eliminar\n");

                scanf("%d",&fila);

                while(fila < 0 || fila >= filas){

                    printf("Ingrese una fila entre 0 y %d\n",filas);

                    scanf("%d",&fila);

                    }

                printf("Ingrese la columna del local que quiere eliminar\n");

                scanf("%d",&columna);

                while(columna < 0 || columna > localesPorPiso[fila]){

                    printf("Ingrese una columna entre 0 y %d\n",localesPorPiso[fila]);

                    scanf("%d",&columna);

                    }

                localesOcupados[fila][columna] = False;

Entradas: Recibe el piso y columna del local a eliminar y va a verificar que estos existan para luego eliminar.

Salidas: Locales.

Conceptos usados: switch, structs, matrices, ciclos y funciones.

  case 7:

                printf("Cantidad de locales totales %d\n",calcularTotalLocales(localesPorPiso,filas) );

                break;

Caso 7 utiliza una función que anteriormente expliqué.