

PROTOTIPOS DE CÓDIGOS

CODIGO CONTROL DE SEMAFOROS:

```
#include<stdio.h>
#include<windows.h>
#include<stdlib.h> //Biblioteca para asignación dinámica de memoria
#include<conio.h> //Biblioteca para el temporizador

void Estado1(int*,int*,int*);
void semaforo(void);
void semaforo2(void);
void salida(void);
void espera(void);
void espera2(void);

int main()
{
    int pulsadorvia = 0, * ppulsadorvia,pulsadorparking=0, * ppulsadorparking,
    dim = 1, contador = 0,salidacoche=0, * psalida;
    //Con asignacion dinamica de memoria reservo 4 bytes de memoria
    (equivalente del int)
    ppulsadorvia = (int*)malloc(dim * sizeof(int));
    ppulsadorparking = (int*)malloc(dim * sizeof(int));
    psalida= (int*)malloc(dim * sizeof(int));
    if (ppulsadorvia == NULL)
    {
        printf("No hay espacio");
        exit(1);
    }
    if (ppulsadorparking == NULL)
    {
        printf("No hay espacio");
        exit(1);
    }
    if (psalida == NULL)
    {
        printf("No hay espacio");
        exit(1);
    }

    ppulsadorvia = &pulsadorvia;
    ppulsadorparking = &pulsadorparking;
    psalida = &salidacoche;
    do
    {
        Estado1(ppulsadorvia,ppulsadorparking,psalida);
        if (*psalida == 0)
        {
            if (*ppulsadorvia != 0 || *ppulsadorparking != 0)
            {
                if (*ppulsadorvia != 0 && *ppulsadorparking == 0)
                {
                    espera(); //Llamada a las funciones espera y
semaforo
                    semaforo();
                    contador++;
                }
            }
        }
    }
}
```

```

        if (*ppulsadorparking != 0 && *ppulsadorvia == 0)
        {
            espera2(); //Llamada a las funciones espera y
semaforo
            semaforo2();
            contador++;
        }
        if (*ppulsadorparking != 0 && *ppulsadorvia != 0)
        {
            espera();
            semaforo();
            espera2();
            semaforo2();
            contador = contador + 2;
        }
    }
}
if (*psalida != 0)
{
    if (*ppulsadorparking != 0 || *ppulsadorvia != 0)
    {
        if (*ppulsadorvia != 0 && *ppulsadorparking == 0)
        {
            espera();
            salida();
            espera(); //Llamada a las funciones espera y
semaforo
            semaforo();
            contador++;
        }
        if (*ppulsadorparking != 0 && *ppulsadorvia == 0)
        {
            espera();
            salida();
            espera2(); //Llamada a las funciones espera y
semaforo
            semaforo2();
            contador++;
        }
        if (*ppulsadorparking != 0 && *ppulsadorvia != 0)
        {
            espera();
            salida();
            espera();
            semaforo();
            espera2();
            semaforo2();
            contador = contador + 2;
        }
    }
}
} while(*ppulsadorvia != 0 || *ppulsadorparking!=0 || *psalida!=0);
printf("\n\nSe han pulsado un total de %d veces los pulsadores de los
semaforos", contador);

free(ppulsadorvia); //Libero el espacio reservado
getchar();
printf("\n\n");
system("pause");
}
void Estado1(int* x,int* y,int* z)

```

```

{
    int pulsavia,pulsaparking,salida;
    printf("\nSemaforo Via\nVerde\n\nSemaforo Parking\nRojo\n\nSemaforo Peaton
Parking\nRojo\n\nSemaforo Peaton Via\nRojo");
    printf("\n\nPulse peaton de via\n");
    scanf_s("%d", &pulsavia);
    printf("\n\nPulse peaton de parking\n");
    scanf_s("%d", &pulsaparking);
    printf("\n\nPrevista salida de un coche del parking\n");
    scanf_s("%d", &salida);
    *x = pulsavia;
    *y = pulsaparking;
    *z = salida;
    system("cls"); //comando para borrar lo impreso en pantalla anteriormente
}

```

```

void espera(void)
{
    int seg;
    for (seg = 5; seg >= 0; seg--)
    {
        printf("\t\nSemaforo Via\nAmbar %d\t\n\nSemaforo Parking\nRojo
%d\t\n\nSemaforo Peaton Parking\nRojo %d\t\n\nSemaforo Peaton Via\n Rojo %d",
seg, seg, seg, seg);
        Sleep(999);
        system("cls");
    }
}

```

```

void semaforo(void)
{
    int color = 2, seg;
    while (color != 0)
    {
        if (color == 2) {
            for (seg = 10; seg >= 0; seg--) {
                printf("\t\nSemaforo Via\nRojo %d\t\n\nSemaforo
Parking\nVerde %d\t\n\nSemaforo Peaton Parking\nRojo %d\t\n\nSemaforo Peaton
Via\nVerde %d", seg, seg, seg, seg);
                Sleep(999); //999 milisegundos
                system("cls");
            }
        }
        else if(color == 1)
        {
            for (seg = 5; seg >= 0; seg--) {
                printf("\t\nSemaforo Via\nAmbar intermitente
%d\t\n\nSemaforo Parking\nAmbar %d\t\n\nSemaforo Peaton Parking\nRojo
%d\t\n\nSemaforo Peaton Via\nVerde Intermitente %d", seg, seg, seg, seg);
                Sleep(999);
                system("cls");
            }
        }
        color--;
    }
}

```

```

void espera2(void)
{
    int seg;
    for (seg = 5; seg >= 0; seg--)
    {

```

```

        printf("\t\nSemaforo Via\nVerde %d\t\n\nSemaforo Parking\nRojo
%d\t\n\nSemaforo Peaton Parking\nRojo %d\t\n\nSemaforo Peaton Via\n Rojo %d",
seg, seg, seg, seg);
        Sleep(999);
        system("cls");
    }
}

void semaforo2(void)
{
    int color = 2, seg;
    while (color != 0)
    {
        if (color == 2) {
            for (seg = 10; seg >= 0; seg--) {
                printf("\t\nSemaforo Via\nVerde %d\t\n\nSemaforo
Parking\nRojo %d\t\n\nSemaforo Peaton Parking\nVerde %d\t\n\nSemaforo Peaton
Via\nRojo %d", seg, seg, seg, seg);
                Sleep(999); //999 milisegundos
                system("cls");
            }
        }
        else if (color == 1)
        {
            for (seg = 5; seg >= 0; seg--) {
                printf("\t\nSemaforo Via\nVerde %d\t\n\nSemaforo
Parking\nAmbar intermitente %d\t\n\nSemaforo Peaton Parking\nVerde intermitente
%d\t\n\nSemaforo Peaton Via\nRojo %d", seg, seg, seg, seg);
                Sleep(999);
                system("cls");
            }
        }
        color--;
    }
}

void salida(void)
{
    int color = 2, seg;
    while (color != 0)
    {
        if (color == 2) {
            for (seg = 10; seg >= 0; seg--) {
                printf("\t\nSemaforo Via\nRojo %d\t\n\nSemaforo
Parking\nVerde %d\t\n\nSemaforo Peaton Parking\nRojo %d\t\n\nSemaforo Peaton
Via\nRojo %d", seg, seg, seg, seg);
                Sleep(999); //999 milisegundos
                system("cls");
            }
        }
        else if (color == 1)
        {
            for (seg = 5; seg >= 0; seg--) {
                printf("\t\nSemaforo Via\nAmbar intermitente
%d\t\n\nSemaforo Parking\nAmbar %d\t\n\nSemaforo Peaton Parking\nRojo
%d\t\n\nSemaforo Peaton Via\nRojo %d", seg, seg, seg, seg);
                Sleep(999);
                system("cls");
            }
        }
        color--;
    }
}

```

```

    }
}

```

CODIGO GUIADO DE CLIENTES:

```

#include<stdio.h>
#include<windows.h>
#include<malloc.h>
#define TAM 5

void situacion(int*);
void direccion(int*);
void lleno(int*,int*);

int main()
{
    int *psensor,cocheentrada=0,cierre=0,completo=0,*pcompleto;
    psensor = (int*)malloc(TAM * sizeof(int));
    if (psensor == NULL)
    {
        printf("No hay memoria suficiente\n\n");
    }
    else
    {
        printf("Memoria asignada correctamente\n\n");
    }
    pcompleto = &completo;
    situacion(psensor);
    lleno(psensor,pcompleto);
    if (*pcompleto == 0)
    {
        do
        {
            printf("Hay coche a la entrada\n1-Si\n0-No\n");
            scanf_s("%d", &cocheentrada);
            lleno(psensor, pcompleto);
            if (*pcompleto == 0)
            {
                if (cocheentrada == 1)
                {
                    direccion(psensor);
                }
            }
            else
            {
                printf("\nLo sentimos nuestro parking esta
completo\n");
            }
            printf("\n%cEs la hora del cierre del parking%c:", 168, 63);
            scanf_s("%d", &cierre);
            printf("\n\n");
        } while (cierre != 1 && *pcompleto != 1);
    }
    else
    {

```

```

        printf("El parking esta completo\n");
    }
}

void situacion(int* psensor)
{
    int i;
    printf("\nIntroduzca el estado de cada sensor situado en cada plaza\n");
    for (i = 0; i < TAM; i++)
    {
        printf("Plaza %d:",i+1);
        scanf_s("%d", psensor+i);
        printf("\n");
    }
}

void direccion(int* psensor)
{
    int i=0,flag=0;

    do
    {
        if (*(psensor + i) == 0)
        {
            printf("\nDirijase a la plaza numero %d\n", i + 1);
            *(psensor + i) = 1;
            flag = 1;
            i++;
        }
        i++;
    } while (flag != 1);
}

void lleno(int* psensor, int* pcompleto)
{
    int i=0,flag=0;
    do
    {
        if (*(psensor + i) == 1)
        {
            *pcompleto = 1;
            i++;
        }
        else if (*(psensor + i) == 0)
        {
            *pcompleto = 0;
            i++;
            flag = 1;
        }
    } while (flag != 1 && i<TAM);
}

```

- Observación: En este código falta implementar la opción de que salga un coche del parking y por lo tanto el puntero psensor tuviera que modificar el estado de esa plaza que queda libre pasando dicha posición a valer "0".