

Trabajo de IC2

Daniel Rodríguez Alonso, Sofia Travieso García, Jaime Rivero Santana

04 abril 2024

Índice

1	Introducción	2
2	Declaración del vector grupo	2
3	Funciones	3
3.1	inicializa_grupo	3
3.2	matricula_alumno	3
3.3	desmatricula_alumno	4
3.4	testea_alumnos	4
3.5	plazas_libres	5
3.6	plazas_ocupadas	5
4	Programa de prueba	5
4.1	matricula	5
4.2	dematricula	6
4.3	matricula_multiple	7
5	Main	7
6	Compilaciones y pruebas	8
7	Conclusión	11

1 Introducción

La practica titulada “Caso práctico en el desarrollo en C”, constituye una oportunidad fundamental para profundizar en diversos aspectos del lenguaje C. A lo largo de este ejercicio, los paraticipantes explorarán conceptos esenciales como punteros, vectores, struct y funciones, todos ellos pilares fundamentales en el dominio de la programación en C.

Esta práctica se presenta como un desafío integral, donde los participantes tendrán la oportunidad de poner en práctica sus conocimientos teóricos y habilidades técnicas adquiridas hasta el momento.

El desarrollo de la practica se dividirá en las siguientes actividades:

2 Declaración del vector grupo

Para la delcaración del vector grupo que tiene que ser de MAX 50 persona hemos creado directamente el vector utilizando el struct de Persona

```
// Definición de la estructura
struct Persona
{
    int dni;
    char nombre[10];
    char apellidos[20];
    int numero_matricula;
} *ptr;
```

En la definicion de esta struct le definimos un dni, nombre, apellido, numero_matricula. Y al final de la definición de ese struct declaramos un puntero global que va a tener acceso todo el programa y que solo va a poder apuntar a un struct de ese tipo.

Luego en el main declaramos en vector de grupo con máximo 50 alumnos.

```
#define MAX 50

int main()
{
    struct Persona clase[MAX];
    ptr = clase;
    return 0;
}
```

Para definir el tamaño del vector utilizamos una constante que estara definida igualmente para todo el programa pero que no podrá cambiar de valor. En esta parte del desarrollo tambien le decimos al puntero *ptr* que va a apuntar a la primera posición del vector clase.

Este código se puede representar gráficamente a través de PythonTutor.

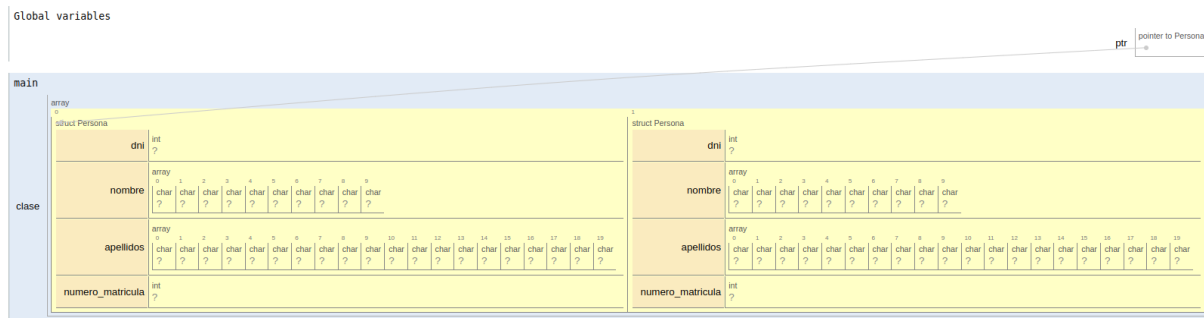


Figura 1: PythonTutor

Ahora pasamos al desarrollo de las funciones

3 Funciones

3.1 inicializa_grupo

La primera función que desarrollamos es la de inicializa_grupo:

```
int inicializa_grupo()
{
    for (int i = 0; i < MAX; i++)
    {
        ptr[i].dni = -1;
    }
    return 0;
};
```

Esta función se utilizará nada más ejecutar el programa para poder poner en todas las posiciones el dni = -1, esto nos indicará que no hay nadie matriculado en esa posición.

3.2 matricula_alumno

```
int matricular_alumno(int dni)
{
    // Comprobar si el alumno ya está matriculado
    for (int i = 0; i < MAX; i++)
    {
        if (ptr[i].dni == dni)
        {
            return -2;
        }
    }
    // Matriculando al alumno
    for (int i = 0; i < MAX; i++)
```

```

{
    if (ptr[i].dni == -1)
    {
        ptr[i].dni = dni;
        return i;
    }
}
return -1;
}

```

Esta función servirá para poder asignarle una plaza a un alumno, en donde comprueba primero que no está matriculado, que en caso de ser así devolverá un -2, si el alumno no estaba matriculado buscará una posición donde el valor dni = -1 lo que significa que esa posición está libre y pondrá el valor del dni que se quiere matricular en esa posición, devolviendo la posición del dni guardado. En caso de que el vector se encuentre lleno y no se matriculará el alumno y la función devolverá un -1.

3.3 desmatricula_alumno

```

int desmatricular_alumno(int dni)
{
    // Buscando al alumno e iterando para desmatricularlo
    for (int i = 0; i < MAX; i++)
    {
        if (ptr[i].dni == dni)
        {
            ptr[i].dni = -1;
            return i;
        }
    }
    return -1;
}

```

Al contrario de la función de matricular_alumno, esta función recibe por parámetro un dni que busca usando el puntero que apunta al vector de clase y cuando los encuentre pone el valor de dni de esa posición a -1 lo que significa que queda libre esa posición.

3.4 testea_alumnos

```

int testea_alumnos(int posicion)
{
    if (posicion < 0 || posicion >= MAX)
    {
        return -2;
    }
    return ptr[posicion].dni;
}

```

Esta función prueba la posición de un alumno, esa posición se pasa por parámetro que en caso de que esa posición sea menor que 0 o mayor que el tamaño del vector devolverá un -2 ya que se está intentando acceder a una posición que no está disponible, en caso de que no se cumpla lo anterior la función devolverá el dni de la persona que está matriculada en esa posición o un -1 si esa posición está libre.

3.5 plazas_libres

```
int plazas_libres()
{
    int count;
    count = 0;
    for (int i = 0; i < MAX; i++)
    {
        if (ptr[i].dni == -1)
        {
            count++;
        }
    }
    return count;
}
```

Esta función no recibe ningún argumento por parámetro, lo único que hace es definir un contador y recorrer el vector de clase y va incrementando el contador cada vez que encuentre una posición con el dni a -1.

3.6 plazas_ocupadas

```
int plazas_ocupadas()
{
    int count;
    count = 0;
    for (int i = 0; i < MAX; i++)
    {
        if (ptr[i].dni != -1)
        {
            count++;
        }
    }
    return count;
}
```

Esta función tiene la misma lógica que la función de plazas_libres lo único que en cada iteración va comprobando si el dni es distinto a -1 lo que significará que la plaza está ocupada y incrementará el contador, cuando termine de iterar devuelve el contador.

4 Programa de prueba

En esta sección del desarrollo crearemos funciones específicas para poder poner a prueba el funcionamiento de las funciones ya definidas y ejecutarlas con parámetros no esperados por las funciones observando su comportamiento.

4.1 matricula

```

void test_matricula(int dni)
{
    for (int i = 0; i < MAX + 2; i++)
    {
        int dni1;
        dni1 = dni + i;
        int resultado = matricular_alumno(dni1);
        if (resultado == -2)
        {
            printf("El alumno ya esta matriculado\n");
        }
        else if (resultado == -1)
        {
            printf("No hay plazas disponibles\n");
        }
        else
        {
            printf("El alumno con DNI %d ha sido matriculado en la posicion %d\n", dni1, resultado);
        }
    }
    // Comprobando que el alumno ya esta matriculado
    for (int i = 0; i < 5; i++)
    {
        int dni1;
        dni1 = dni + i;
        int resultado = matricular_alumno(dni1);
        if (resultado == -2)
        {
            printf("El alumno ya esta matriculado\n");
        }
    }
}

```

El uso de esta función es para poder testear la función de `matricula_alumno` en donde realizamos una iteración matriculando diversos dnis y dependiendo del resultado imprimimos por pantalla diferentes mensajes.

4.2 dematricula

```

void test_desmatricula(int dni)
{
    for (int i = 0; i < MAX - 10; i++)
    {
        int dni1;
        dni1 = dni + i;
        int resultado = desmatricular_alumno(dni1);
        if (resultado == -1)
        {
            printf("El alumno no existe\n");
        }
        else {
            printf("El alumno con DNI %d ha sido desmatriculado de la posicion %d\n",

```

```

        dni1, resultado);
    }
}
for (int i = 0; i < 3; i++)
{
    int dni1;
    dni1 = dni + i;
    int resultado = desmatricular_alumno(dni1);
    if (resultado == -1)
    {
        printf("El alumno no existe\n");
    }
}
}

```

Esta función testea la función de desmatricula_alumno llamándola un número MAX de veces y pansandole un dni diferente en cada iteración. Cuando se llama la función test dependiendo del resultado imprime un mensaje u otro

4.3 matricula_multiple

```

void matricula_multiple(int npersonas, int dnis[])
{
    int plazas = plazas_libres();
    if (plazas < npersonas)
    {
        printf("No hay suficientes plazas libres");
        return;
    }

    // Matricular a cada persona de la lista
    for (int i = 0; i < npersonas; i++)
    {
        matricular_alumno(dnis[i]);
    }
    printf("Se han matriculado todos los alumnos \n");
}

```

matricula_multiple es una función que trata de matricular a un conjunto de alumnos que se pasa por parametro y que solo los matriculara en caso de que hubiese le mismos de espacios libres que de alumnos que se quieren matricular en caso contrario no se matriculará a ninguno.

5 Main

Aqui mostraremos la parte del codigo que tiene el main()

```

int main()
{
    // Crear el vector de estructuras

```

```

int dnis[2] = {1234567, 1324567};
struct Persona clase[MAX];
ptr = clase;
inicializa_grupo();
test_matricula(79072763);
test_desmatricula(79072763);
matricula_multiple(2, dnis);
int resultado = plazas_libres();
printf("Hay %d plazas libres\n", resultado);
int resultado2 = plazas_ocupadas();
printf("Hay %d plazas ocupadas\n", resultado2);
printf("El alumno en la posición %d, tiene el dni = %d\n", 1, testea_alumnos(1));
printf("El alumno en la posición %d, tiene el dni = %d\n", 5, testea_alumnos(5));
printf("El alumno en la posición %d, tiene el dni = %d\n", 48, testea_alumnos(48));
return 0;
}

```

6 Compilaciones y pruebas

En la compilacion del código ejecutamos los siguientes comandos

```
gcc main.c -o programa
```

Y ejecutamos el programa

```
./programa
```

y visualizamos el resultado

```

daniel@daniel-nsi:~/Escritorio/IC2_Practicas/trabajos$ gcc main.c -o programa
daniel@daniel-nsi:~/Escritorio/IC2_Practicas/trabajos$ ./programa
El alumno con DNI 79972763 ha sido matriculado en la posición 0
El alumno con DNI 79972764 ha sido matriculado en la posición 1
El alumno con DNI 79972765 ha sido matriculado en la posición 2
El alumno con DNI 79972766 ha sido matriculado en la posición 3
El alumno con DNI 79972767 ha sido matriculado en la posición 4
El alumno con DNI 79972768 ha sido matriculado en la posición 5
El alumno con DNI 79972769 ha sido matriculado en la posición 6
El alumno con DNI 79972770 ha sido matriculado en la posición 7
El alumno con DNI 79972771 ha sido matriculado en la posición 8
El alumno con DNI 79972772 ha sido matriculado en la posición 9
El alumno con DNI 79972773 ha sido matriculado en la posición 10
El alumno con DNI 79972774 ha sido matriculado en la posición 11
El alumno con DNI 79972775 ha sido matriculado en la posición 12
El alumno con DNI 79972776 ha sido matriculado en la posición 13
El alumno con DNI 79972777 ha sido matriculado en la posición 14
El alumno con DNI 79972778 ha sido matriculado en la posición 15
El alumno con DNI 79972779 ha sido matriculado en la posición 16
El alumno con DNI 79972780 ha sido matriculado en la posición 17
El alumno con DNI 79972781 ha sido matriculado en la posición 18
El alumno con DNI 79972782 ha sido matriculado en la posición 19
El alumno con DNI 79972783 ha sido matriculado en la posición 20
El alumno con DNI 79972784 ha sido matriculado en la posición 21
El alumno con DNI 79972785 ha sido matriculado en la posición 22
El alumno con DNI 79972786 ha sido matriculado en la posición 23
El alumno con DNI 79972787 ha sido matriculado en la posición 24
El alumno con DNI 79972788 ha sido matriculado en la posición 25
El alumno con DNI 79972789 ha sido matriculado en la posición 26
El alumno con DNI 79972790 ha sido matriculado en la posición 27
El alumno con DNI 79972791 ha sido matriculado en la posición 28
El alumno con DNI 79972792 ha sido matriculado en la posición 29
El alumno con DNI 79972793 ha sido matriculado en la posición 30
El alumno con DNI 79972794 ha sido matriculado en la posición 31
El alumno con DNI 79972795 ha sido matriculado en la posición 32
El alumno con DNI 79972796 ha sido matriculado en la posición 33
El alumno con DNI 79972797 ha sido matriculado en la posición 34
El alumno con DNI 79972798 ha sido matriculado en la posición 35
El alumno con DNI 79972799 ha sido matriculado en la posición 36
El alumno con DNI 79972800 ha sido matriculado en la posición 37
El alumno con DNI 79972801 ha sido matriculado en la posición 38
El alumno con DNI 79972802 ha sido matriculado en la posición 39
El alumno con DNI 79972803 ha sido matriculado en la posición 40
El alumno con DNI 79972804 ha sido matriculado en la posición 41
El alumno con DNI 79972805 ha sido matriculado en la posición 42
El alumno con DNI 79972806 ha sido matriculado en la posición 43
El alumno con DNI 79972807 ha sido matriculado en la posición 44
El alumno con DNI 79972808 ha sido matriculado en la posición 45
El alumno con DNI 79972809 ha sido matriculado en la posición 46
El alumno con DNI 79972810 ha sido matriculado en la posición 47
El alumno con DNI 79972811 ha sido desmatriculado de la posición 0
El alumno con DNI 79972812 ha sido desmatriculado de la posición 1
El alumno con DNI 79972813 ha sido desmatriculado de la posición 2
El alumno con DNI 79972814 ha sido desmatriculado de la posición 3
El alumno con DNI 79972815 ha sido desmatriculado de la posición 4
El alumno con DNI 79972816 ha sido desmatriculado de la posición 5
El alumno con DNI 79972817 ha sido desmatriculado de la posición 6
El alumno con DNI 79972818 ha sido desmatriculado de la posición 7

```

Figura 2: Terminal

Vemos que al probar las funciones de testea matricula y testea desmatricula funciona perfectamente y rellena el vector clase y luego los desmatricula entero.

Ahora vamos a intentar compilar el trabajo intentando matricular mas de 50 alumnos y y no desmatriculandolos todos para ver si funciona el codigo y las funciones de plazas libres y plazas ocupadas.

```
daniel@daniel-msi:~/Escritorio/IC2_Practicas/trabajo$ gcc main.c -o programa
main.c: In function 'main':
main.c:160:9: error: redefinition of 'resultado'
160 |     int resultado = plazas_ocupadas();
    |     ^~~~~~
main.c:158:9: note: previous definition of 'resultado' with type 'int'
158 |     int resultado = plazas_libres();
    |     ^~~~~~
```

Figura 3: Error

Al intentar compilar este programa vemos que nos salta un error que es bastante sencillo de solucionar, se trata de que en el apartado de main() de nuestro trabajo duplicamos la variable de resultado 2 veces, cambiando el nombre a esta variable solucionamos el problema y podremos compilar el programa.

```
El alumno con DNI 79072800 ha sido matriculado en la posicion 37
El alumno con DNI 79072801 ha sido matriculado en la posicion 38
El alumno con DNI 79072802 ha sido matriculado en la posicion 39
El alumno con DNI 79072803 ha sido matriculado en la posicion 40
El alumno con DNI 79072804 ha sido matriculado en la posicion 41
El alumno con DNI 79072805 ha sido matriculado en la posicion 42
El alumno con DNI 79072806 ha sido matriculado en la posicion 43
El alumno con DNI 79072807 ha sido matriculado en la posicion 44
El alumno con DNI 79072808 ha sido matriculado en la posicion 45
El alumno con DNI 79072809 ha sido matriculado en la posicion 46
El alumno con DNI 79072810 ha sido matriculado en la posicion 47
El alumno con DNI 79072811 ha sido matriculado en la posicion 48
El alumno con DNI 79072812 ha sido matriculado en la posicion 49
No hay plazas disponibles
No hay plazas disponibles
El alumno con DNI 79072763 ha sido desmatriculado de la posicion 0
El alumno con DNI 79072764 ha sido desmatriculado de la posicion 1
El alumno con DNI 79072765 ha sido desmatriculado de la posicion 2
El alumno con DNI 79072766 ha sido desmatriculado de la posicion 3
El alumno con DNI 79072767 ha sido desmatriculado de la posicion 4
El alumno con DNI 79072768 ha sido desmatriculado de la posicion 5
El alumno con DNI 79072769 ha sido desmatriculado de la posicion 6
El alumno con DNI 79072770 ha sido desmatriculado de la posicion 7
El alumno con DNI 79072771 ha sido desmatriculado de la posicion 8
El alumno con DNI 79072772 ha sido desmatriculado de la posicion 9
El alumno con DNI 79072773 ha sido desmatriculado de la posicion 10
El alumno con DNI 79072774 ha sido desmatriculado de la posicion 11
El alumno con DNI 79072775 ha sido desmatriculado de la posicion 12
El alumno con DNI 79072776 ha sido desmatriculado de la posicion 13
El alumno con DNI 79072777 ha sido desmatriculado de la posicion 14
El alumno con DNI 79072778 ha sido desmatriculado de la posicion 15
El alumno con DNI 79072779 ha sido desmatriculado de la posicion 16
El alumno con DNI 79072780 ha sido desmatriculado de la posicion 17
El alumno con DNI 79072781 ha sido desmatriculado de la posicion 18
El alumno con DNI 79072782 ha sido desmatriculado de la posicion 19
El alumno con DNI 79072783 ha sido desmatriculado de la posicion 20
El alumno con DNI 79072784 ha sido desmatriculado de la posicion 21
El alumno con DNI 79072785 ha sido desmatriculado de la posicion 22
El alumno con DNI 79072786 ha sido desmatriculado de la posicion 23
El alumno con DNI 79072787 ha sido desmatriculado de la posicion 24
El alumno con DNI 79072788 ha sido desmatriculado de la posicion 25
El alumno con DNI 79072789 ha sido desmatriculado de la posicion 26
El alumno con DNI 79072790 ha sido desmatriculado de la posicion 27
El alumno con DNI 79072791 ha sido desmatriculado de la posicion 28
El alumno con DNI 79072792 ha sido desmatriculado de la posicion 29
El alumno con DNI 79072793 ha sido desmatriculado de la posicion 30
El alumno con DNI 79072794 ha sido desmatriculado de la posicion 31
El alumno con DNI 79072795 ha sido desmatriculado de la posicion 32
El alumno con DNI 79072796 ha sido desmatriculado de la posicion 33
El alumno con DNI 79072797 ha sido desmatriculado de la posicion 34
El alumno con DNI 79072798 ha sido desmatriculado de la posicion 35
El alumno con DNI 79072799 ha sido desmatriculado de la posicion 36
El alumno con DNI 79072800 ha sido desmatriculado de la posicion 37
El alumno con DNI 79072801 ha sido desmatriculado de la posicion 38
El alumno con DNI 79072802 ha sido desmatriculado de la posicion 39
Se han matriculado todos los alumnos
Hay 38 plazas libres
Hay 38 plazas ocupadas
daniel@daniel-msi:~/Escritorio/IC2_Practicas/trabajo$
```

Figura 4: Terminal

En esta compilación el programa funciona pero nos aparece que hay el mismo numero de plazas ocupadas que de plazas libres, y nos dimos cuenta del problema que se encontraba en el main():

```
int main()
{
    // Crear el vector de estructuras
    int dnis[2] = {1234567, 1324567};
    struct Persona clase[MAX];
```

```

ptr = clase;
inicializa_grupo();
test_matricula(79072763);
test_desmatricula(79072763);
matricula_multiple(2, dnis);
int resultado = plazas_libres();
printf("Hay %d plazas libres\n", resultado);
int resultado2 = plazas_ocupadas();
printf("Hay %d plazas ocupadas\n", resultado);
return 0;
}

```

El problema se encuentra en que en el segundo printf imprimimos la misma variable por eso el resultado es dos veces el mismo, cambiando eso se solucionaría el problema.

Volvemos a compilar el programa y vemos como esta vez el resultado si es correcto donde se matriculan 52 alumnos y cuando llega a su límite te salta el mensaje de “No hay plazas disponibles” y consecutivamente se desmatriculan 40 alumnos, cuando termina esos procesos se matriculan dos alumnos con la funcion de matricula_multiple() y salta el mensaje de “Se han matriculado todos los alumnos”, por ultimo salta los mensajes de “Hay 38 plazas libres” y “Hay 12 plazas ocupadas” teniendo todo el sentido ya que $38 + 12 = 50$ el número de alumnos máximo por clase.

```

El alumno con DNI 79072800 ha sido matriculado en la posicion 37
El alumno con DNI 79072801 ha sido matriculado en la posicion 38
El alumno con DNI 79072802 ha sido matriculado en la posicion 39
El alumno con DNI 79072803 ha sido matriculado en la posicion 40
El alumno con DNI 79072804 ha sido matriculado en la posicion 41
El alumno con DNI 79072805 ha sido matriculado en la posicion 42
El alumno con DNI 79072806 ha sido matriculado en la posicion 43
El alumno con DNI 79072807 ha sido matriculado en la posicion 44
El alumno con DNI 79072808 ha sido matriculado en la posicion 45
El alumno con DNI 79072809 ha sido matriculado en la posicion 46
El alumno con DNI 79072810 ha sido matriculado en la posicion 47
El alumno con DNI 79072811 ha sido matriculado en la posicion 48
El alumno con DNI 79072812 ha sido matriculado en la posicion 49
No hay plazas disponibles
No hay plazas disponibles
El alumno con DNI 79072763 ha sido desmatriculado de la posicion 0
El alumno con DNI 79072764 ha sido desmatriculado de la posicion 1
El alumno con DNI 79072765 ha sido desmatriculado de la posicion 2
El alumno con DNI 79072766 ha sido desmatriculado de la posicion 3
El alumno con DNI 79072767 ha sido desmatriculado de la posicion 4
El alumno con DNI 79072768 ha sido desmatriculado de la posicion 5
El alumno con DNI 79072769 ha sido desmatriculado de la posicion 6
El alumno con DNI 79072770 ha sido desmatriculado de la posicion 7
El alumno con DNI 79072771 ha sido desmatriculado de la posicion 8
El alumno con DNI 79072772 ha sido desmatriculado de la posicion 9
El alumno con DNI 79072773 ha sido desmatriculado de la posicion 10
El alumno con DNI 79072774 ha sido desmatriculado de la posicion 11
El alumno con DNI 79072775 ha sido desmatriculado de la posicion 12
El alumno con DNI 79072776 ha sido desmatriculado de la posicion 13
El alumno con DNI 79072777 ha sido desmatriculado de la posicion 14
El alumno con DNI 79072778 ha sido desmatriculado de la posicion 15
El alumno con DNI 79072779 ha sido desmatriculado de la posicion 16
El alumno con DNI 79072780 ha sido desmatriculado de la posicion 17
El alumno con DNI 79072781 ha sido desmatriculado de la posicion 18
El alumno con DNI 79072782 ha sido desmatriculado de la posicion 19
El alumno con DNI 79072783 ha sido desmatriculado de la posicion 20
El alumno con DNI 79072784 ha sido desmatriculado de la posicion 21
El alumno con DNI 79072785 ha sido desmatriculado de la posicion 22
El alumno con DNI 79072786 ha sido desmatriculado de la posicion 23
El alumno con DNI 79072787 ha sido desmatriculado de la posicion 24
El alumno con DNI 79072788 ha sido desmatriculado de la posicion 25
El alumno con DNI 79072789 ha sido desmatriculado de la posicion 26
El alumno con DNI 79072790 ha sido desmatriculado de la posicion 27
El alumno con DNI 79072791 ha sido desmatriculado de la posicion 28
El alumno con DNI 79072792 ha sido desmatriculado de la posicion 29
El alumno con DNI 79072793 ha sido desmatriculado de la posicion 30
El alumno con DNI 79072794 ha sido desmatriculado de la posicion 31
El alumno con DNI 79072795 ha sido desmatriculado de la posicion 32
El alumno con DNI 79072796 ha sido desmatriculado de la posicion 33
El alumno con DNI 79072797 ha sido desmatriculado de la posicion 34
El alumno con DNI 79072798 ha sido desmatriculado de la posicion 35
El alumno con DNI 79072799 ha sido desmatriculado de la posicion 36
El alumno con DNI 79072800 ha sido desmatriculado de la posicion 37
El alumno con DNI 79072801 ha sido desmatriculado de la posicion 38
El alumno con DNI 79072802 ha sido desmatriculado de la posicion 39
Se han matriculado todos los alumnos
Hay 38 plazas libres
Hay 12 plazas ocupadas
daniel@daniel-nsi:~/Escritorio/IC2_Practicas/trabajo5$

```

Figura 5: Terminal

La última compilación vamos a probar la funcion de testea_alumno en el que le pasas una posicion y te devuelve el dni de la persona en esa posición.

```

El alumno con DNI 79072786 ha sido desmatriculado de la posicion 23
El alumno con DNI 79072787 ha sido desmatriculado de la posicion 24
El alumno con DNI 79072788 ha sido desmatriculado de la posicion 25
El alumno con DNI 79072789 ha sido desmatriculado de la posicion 26
El alumno con DNI 79072790 ha sido desmatriculado de la posicion 27
El alumno con DNI 79072791 ha sido desmatriculado de la posicion 28
El alumno con DNI 79072792 ha sido desmatriculado de la posicion 29
El alumno con DNI 79072793 ha sido desmatriculado de la posicion 30
El alumno con DNI 79072794 ha sido desmatriculado de la posicion 31
El alumno con DNI 79072795 ha sido desmatriculado de la posicion 32
El alumno con DNI 79072796 ha sido desmatriculado de la posicion 33
El alumno con DNI 79072797 ha sido desmatriculado de la posicion 34
El alumno con DNI 79072798 ha sido desmatriculado de la posicion 35
El alumno con DNI 79072799 ha sido desmatriculado de la posicion 36
El alumno con DNI 79072800 ha sido desmatriculado de la posicion 37
El alumno con DNI 79072801 ha sido desmatriculado de la posicion 38
El alumno con DNI 79072802 ha sido desmatriculado de la posicion 39
Se han matriculado todos los alumnos
Hay 38 plazas libres. "Se han matriculado todos los alumnos \n");
Hay 12 plazas ocupadas
El alumno en la posición 1, tiene el dni = 1324567
El alumno en la posición 5, tiene el dni = -1
El alumno en la posición 48, tiene el dni = 79072811

```

Figura 6: Terminal

Como vemos el resultado tiene sentido y la función muestra el dni de la posición pasada por parámetro.

7 Conclusión

El desarrollo de este trabajo nos ha dado la oportunidad de profundizar en diversos aspectos del lenguaje de C. En el que hemos visto elementos esenciales como punteros, vectores, struct y funciones, siendo concientes de la posibilidad de mejora de este trabajo dejamos por escrito las ganas que tenemos de mejorar y aladir mejores funcionalidades como memoria dinámica para siguientes entregas adjuntamos un link de [GitHub](#) donde esta el repositorio del trabajo con todas las versiones.