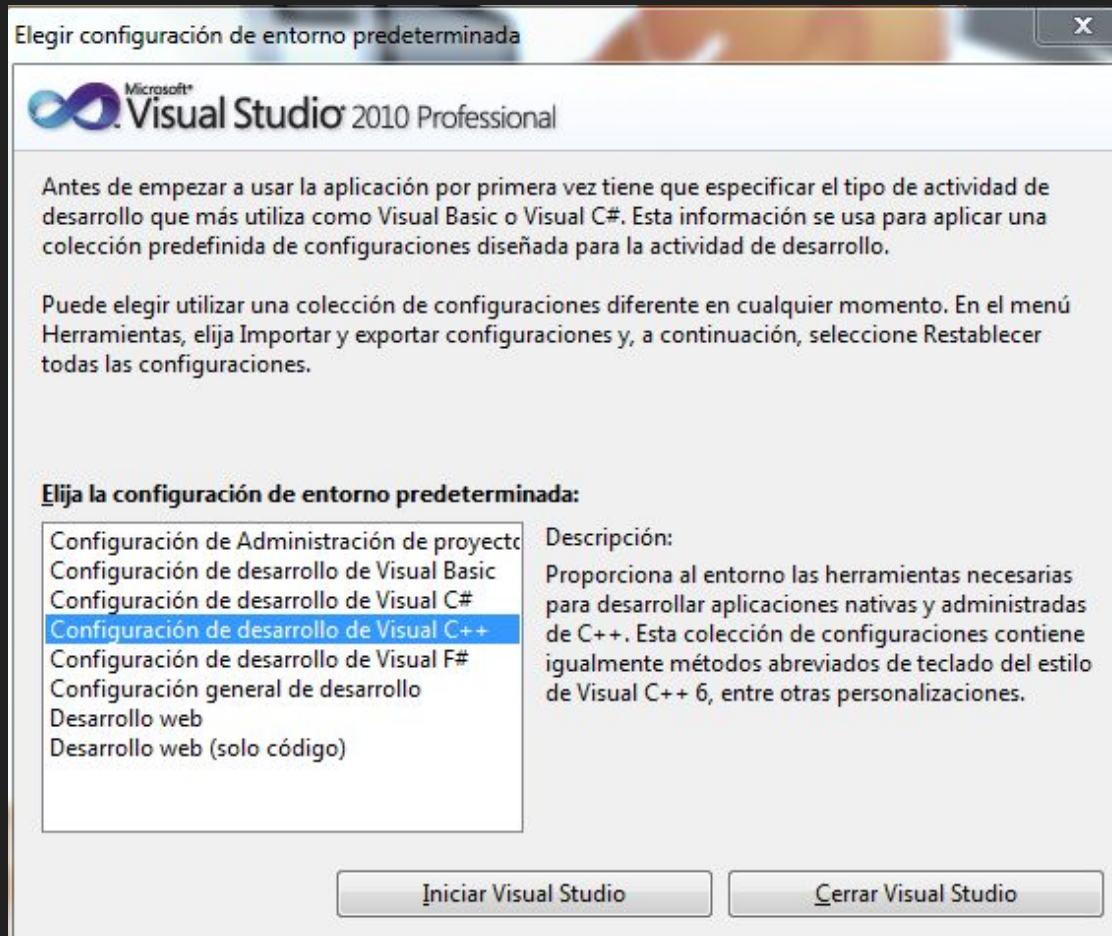


IINFRATEC 2017-10

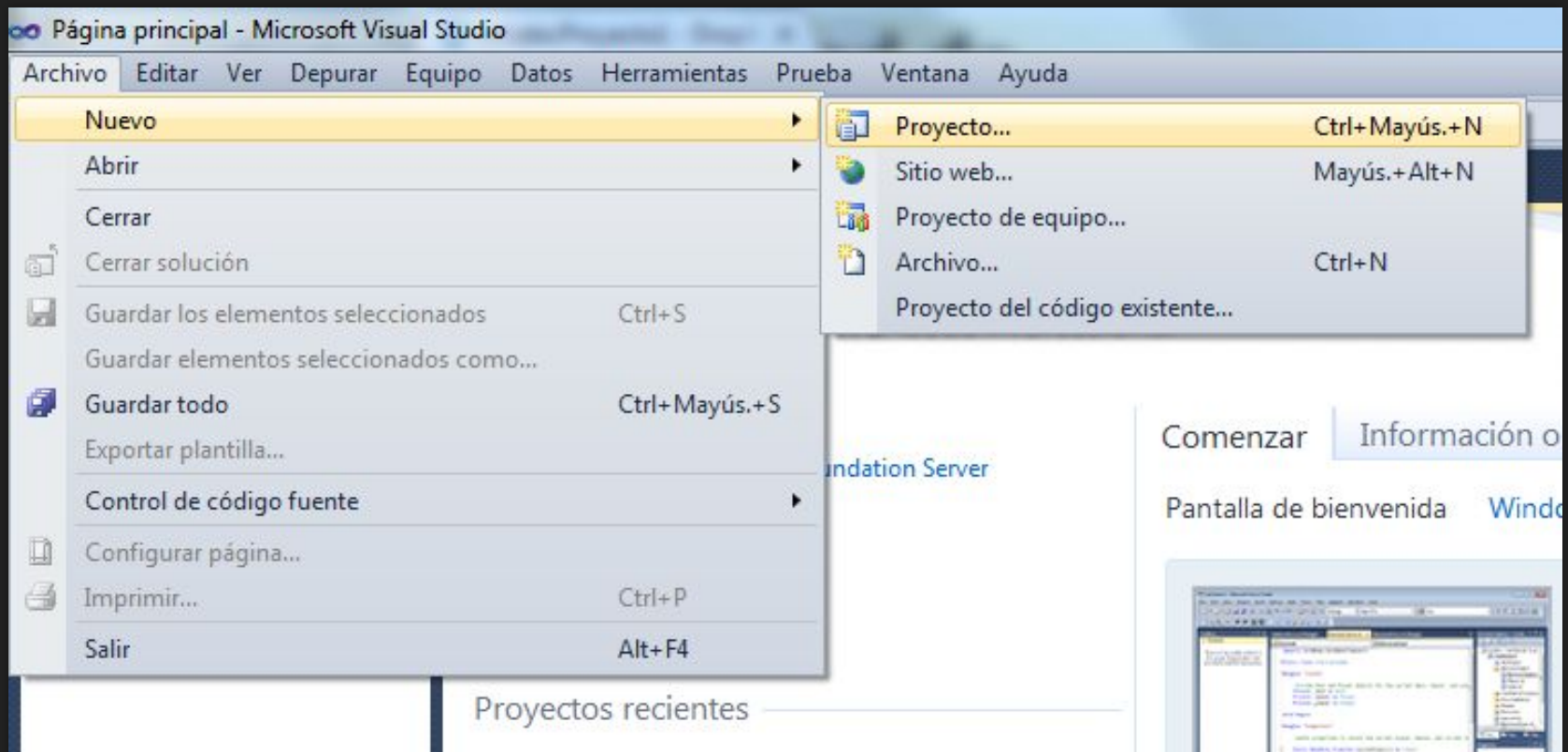
**Monitoria 1
Visual Studio y C**

Febrero 2017

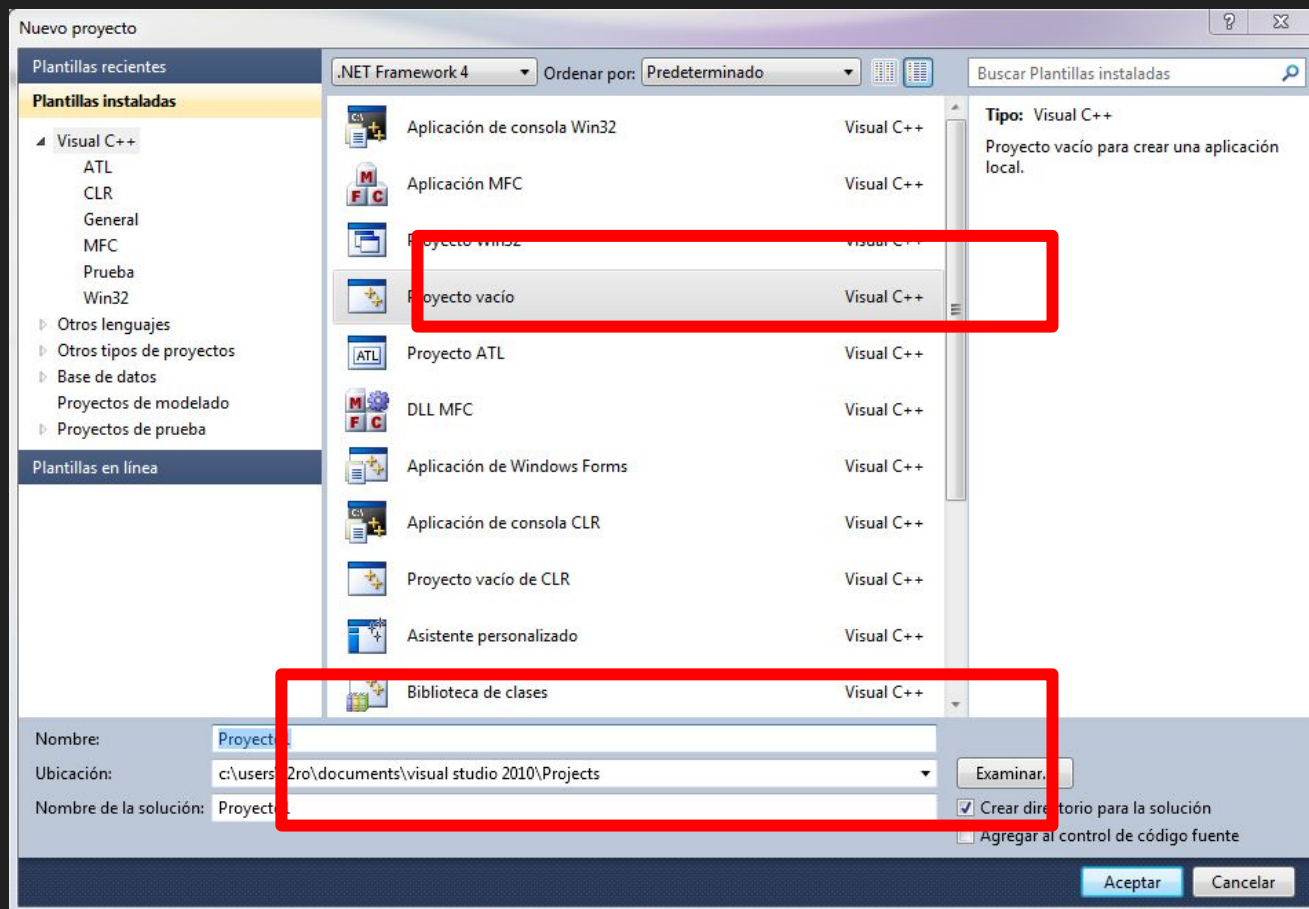
Creación proyecto Visual Studio



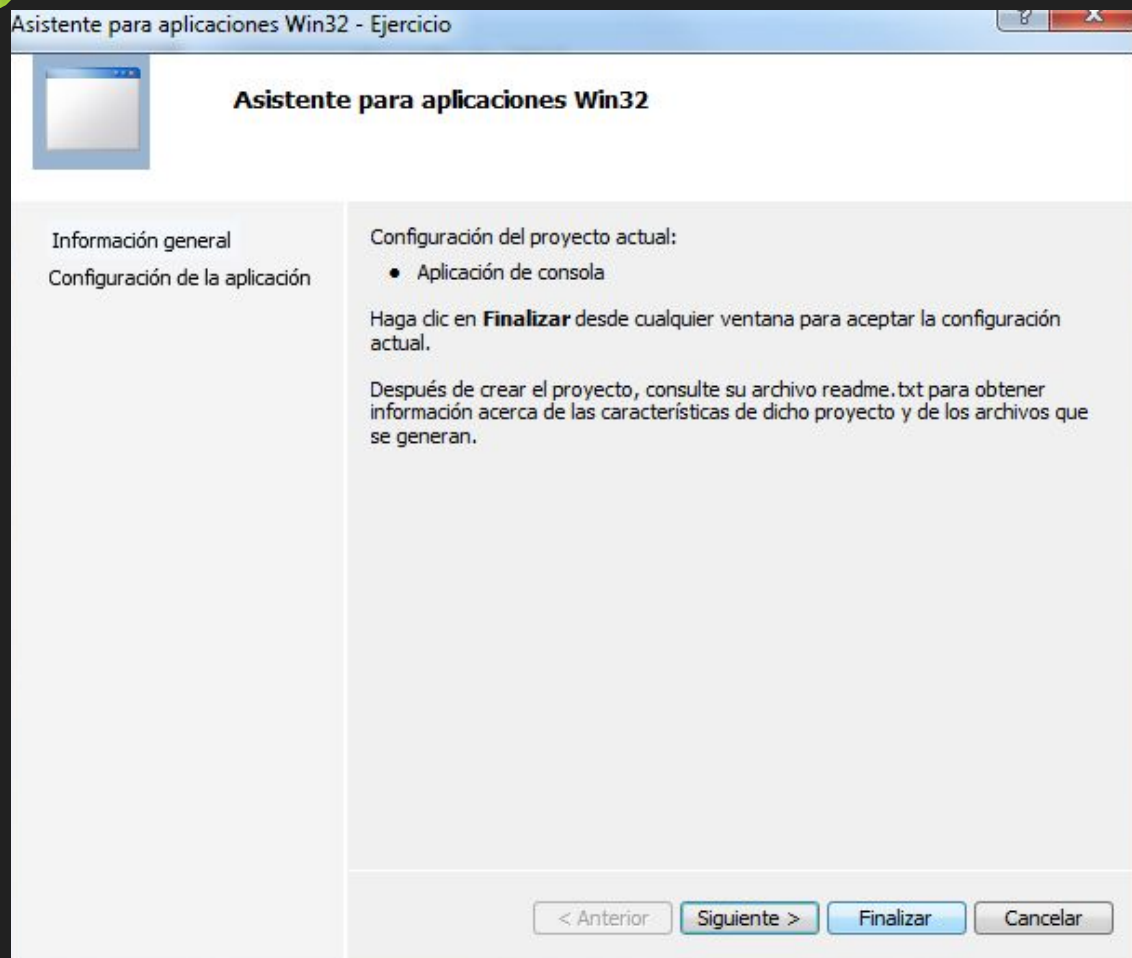
Creación proyecto Visual Studio



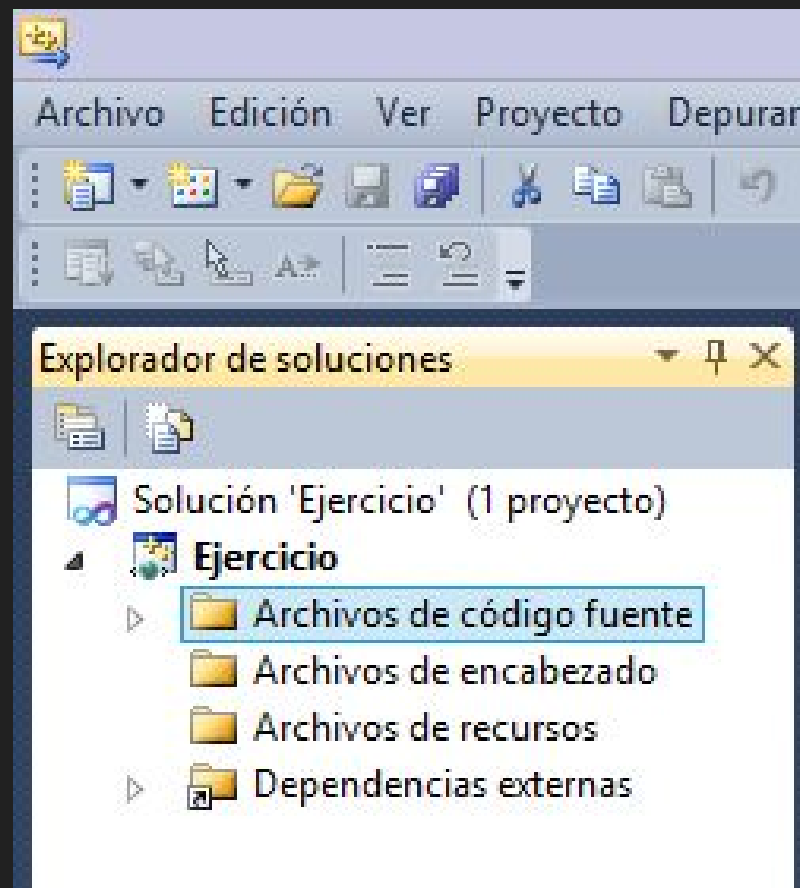
Creación proyecto Visual Studio



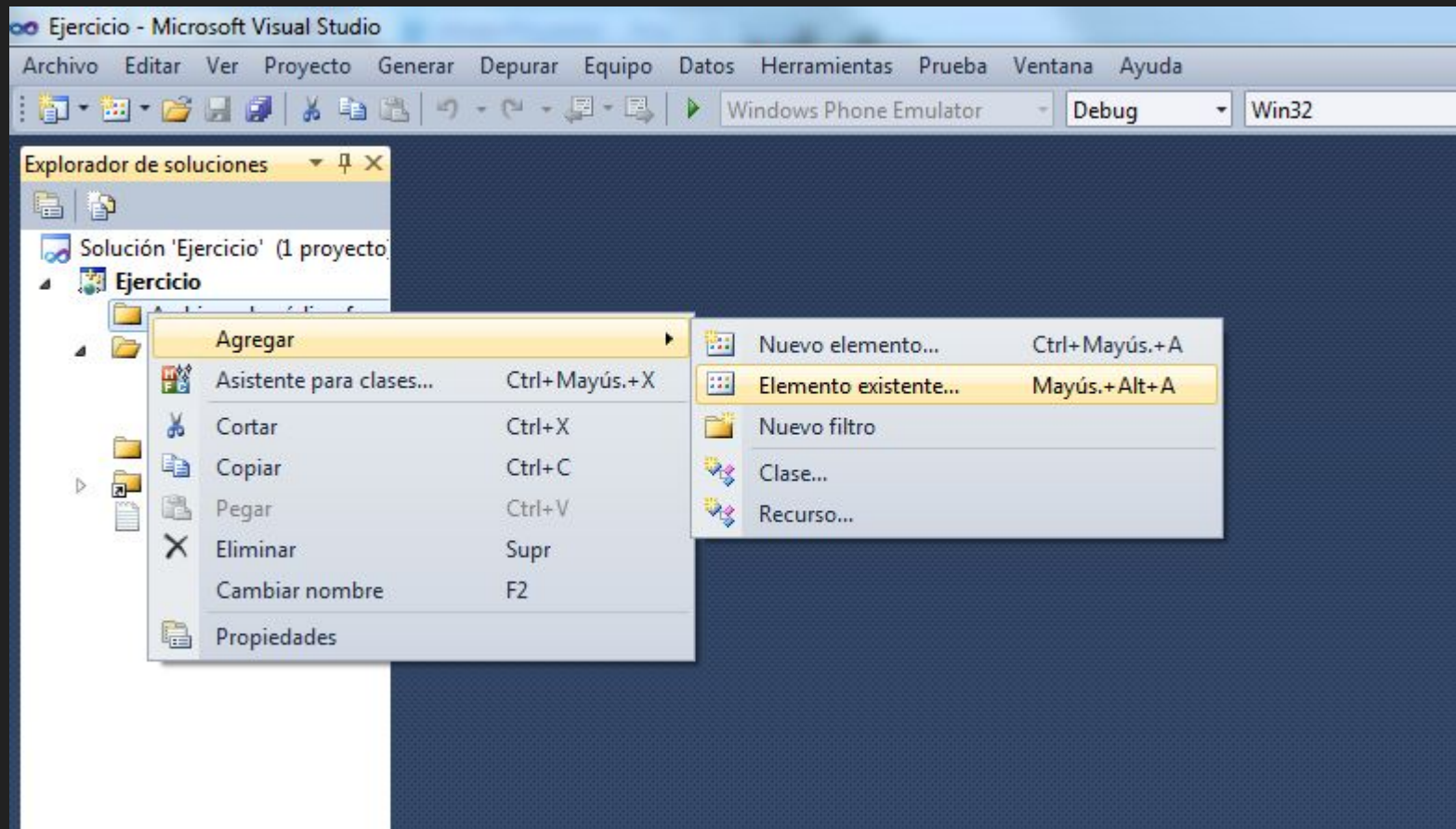
Creación proyecto Visual Studio



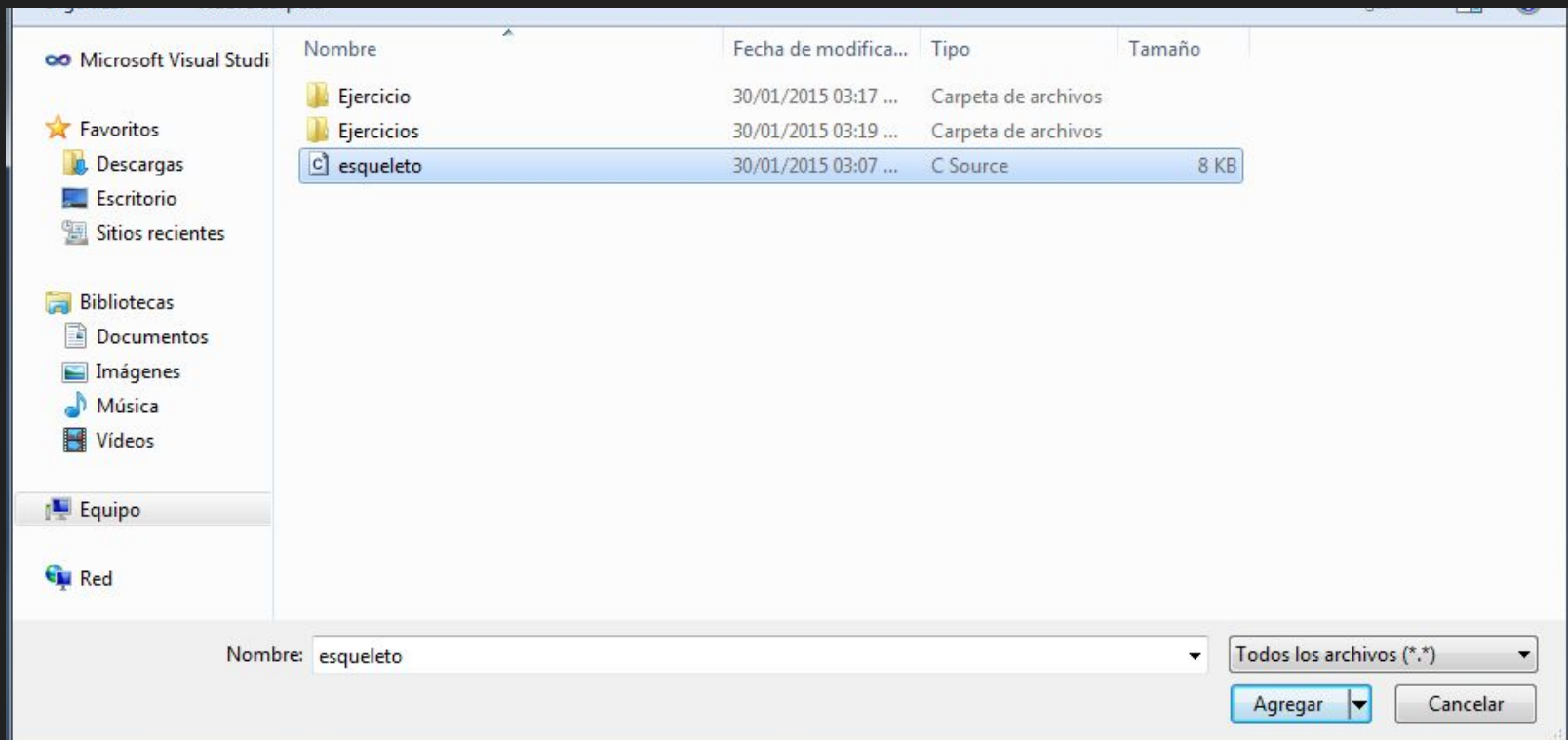
Creación proyecto Visual Studio



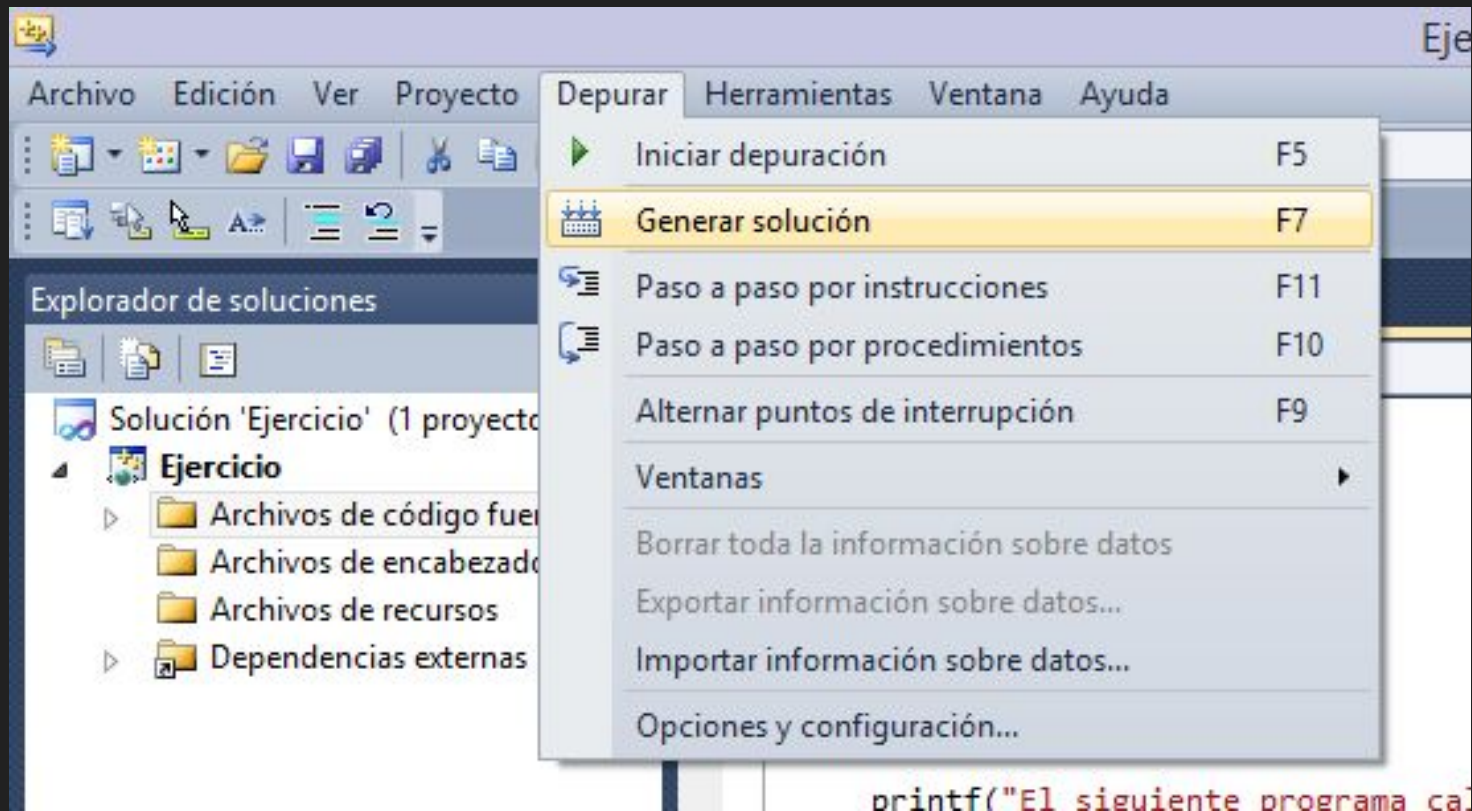
Creación proyecto Visual Studio



Creación proyecto Visual Studio



Compilación



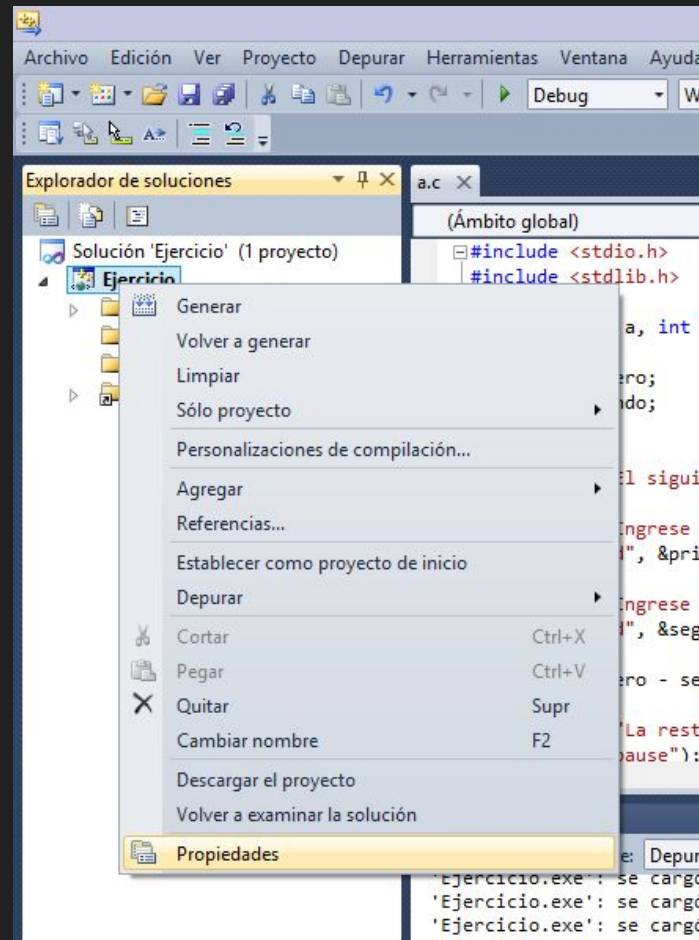
Compilación

Resultados

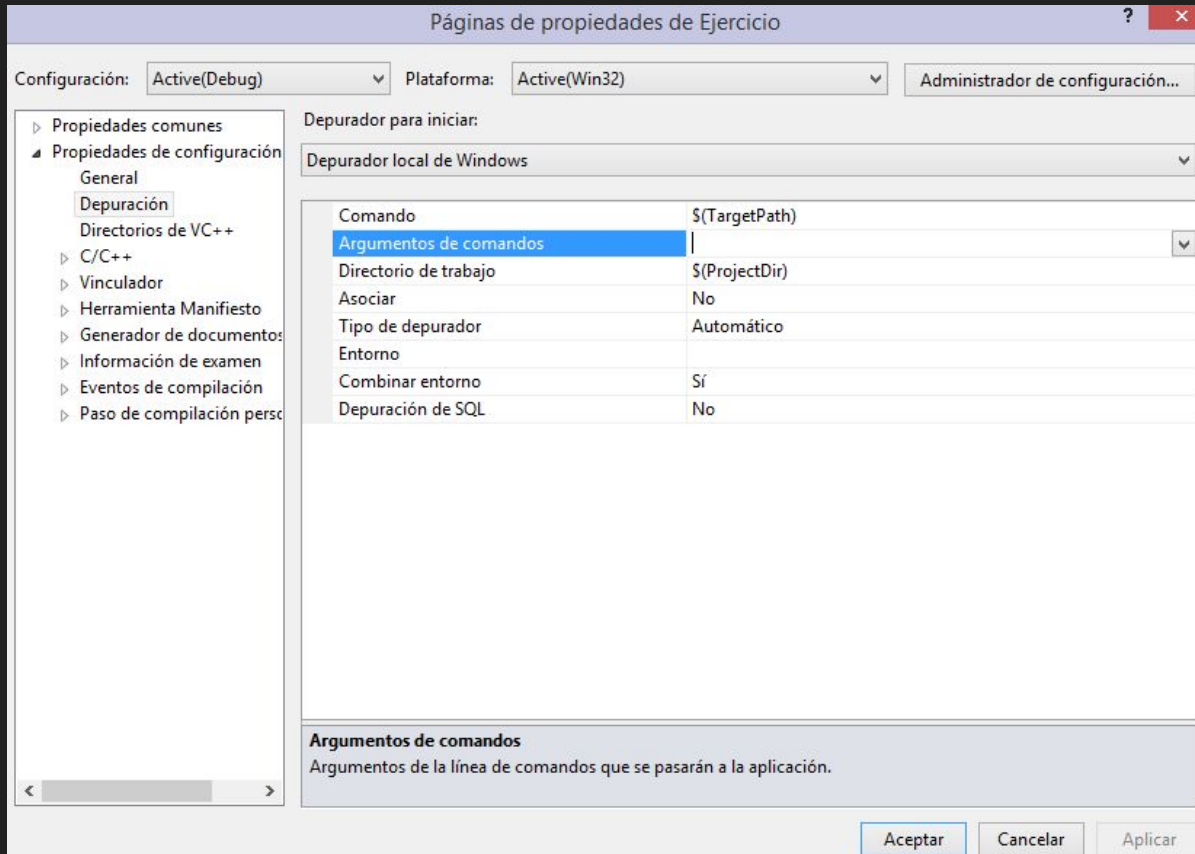
Mostrar resultados desde:

```
1>----- Operación Generar iniciada: proyecto: Ejercicio, configuración: Debug Win32 -----
1> aaa.vcxproj -> C:\Users\Meil\Downloads\aaa\Debug\Ejercicio.exe
===== Generar: 1 correctos, 0 incorrectos, 0 actualizados, 0 omitidos =====
|
```

Pasos a seguir cuando hay archivos de entrada y/o salida



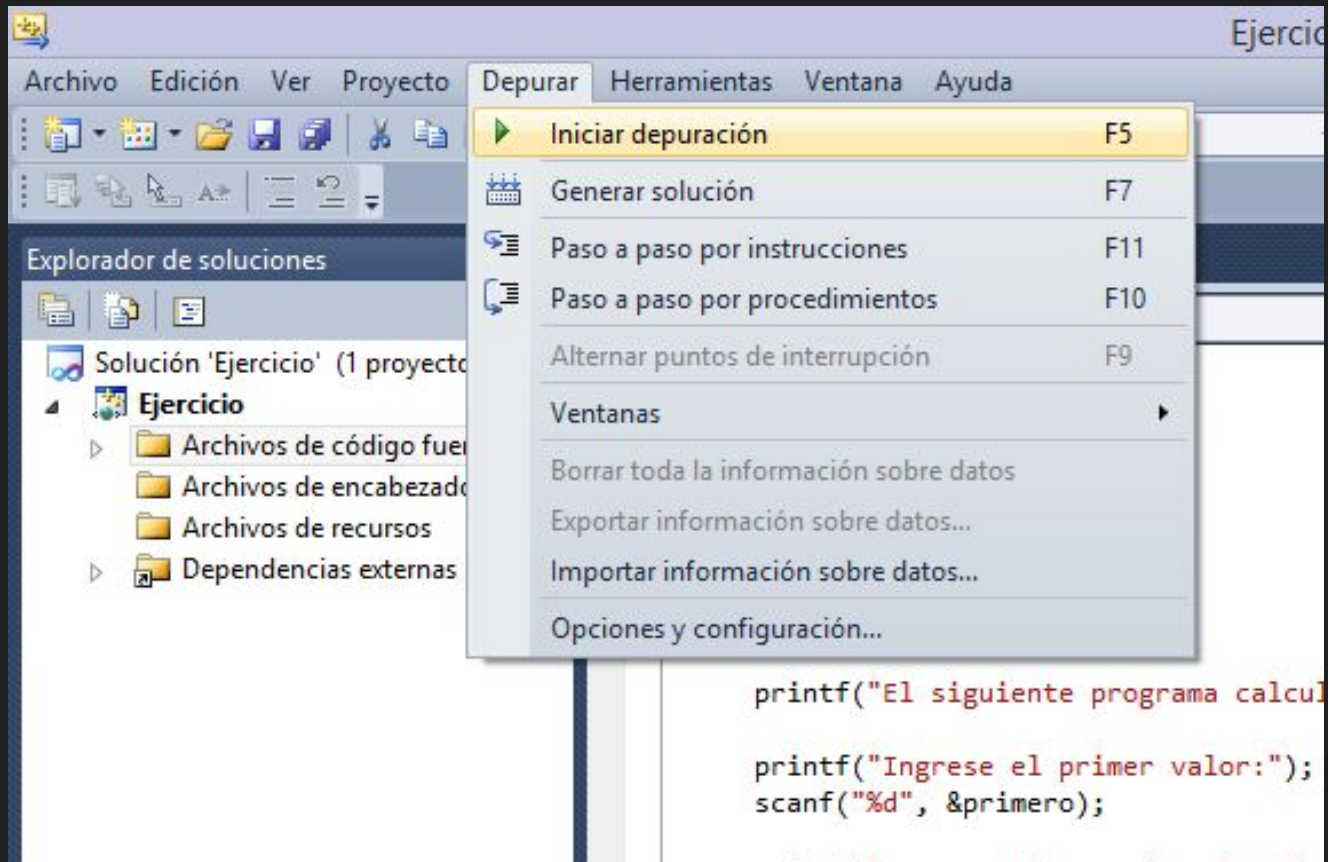
Archivos de entrada y salida



En “Argumentos de comandos” se debe ingresar la ruta de entrada y la ruta de salida en orden.

Cada ruta debe estar entre comillas y las rutas deben estar separadas por un espacio.

Compilación



Introducción al Lenguaje C

Estructura básica

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int x;
5 int andBit(int a , int b);
6 int orBit(int a, int b);
7 int xor(int a, int b);
8 double complemento(int a, int b);
9
10 int andBit(int a, int b)
11 {
12     //TODO
13     return a & b;
14 }
15
16 int orBit(int a, int b)
17 {
18     //TODO
19     return a | b;
20 }
```

Librerías

Declaración de
funciones y
de variables globales

Funciones

Estructura básica

Método main()

- Es el método principal que inicia la ejecución del programa.

TIPOS DE DATOS

TIPO	DESCRIPCION	LONGITUD	RANGO	Especificadores
char	Este tipo de dato es utilizado para representar un código ASCII, y siempre los char van encerrados en comillas simples. (')	1 BYTE	0 a 255 -128 a 128	c
Int	Es un numero sin punto decimal.	2 BYTE	-32678 A 32678	d
float	Es un numero con punto decimal.	4 Bytes	$-3,40 * 10^{38}$ a $3,40 * 10^{38}$	f
double	Es un tipo de dato real de doble precisión de 15 dígitos.	8 Bytes	$-1,79 * 10^{324}$ a $1,79 * 10^{324}$	f/e/E/g
void	Se utiliza para declarar funciones que no retornan un valor, o para declarar un puntero a un tipo de datos no especifico.			

Modificador

NOMBRE	DESCRIPCION
<code>unsigned</code>	Entero en el rango 0 a 65535
<code>short</code>	Entero corto -32768 a 32767
<code>long</code>	Entero en formato largo en el rango ± 214783648

Cadenas de caracteres

Las cadenas son arreglos unidimensionales de tipo char por convención, una cadena en C termina con el carácter nulo, `'\0'` (`\cero`).

Las constantes de cadena se escriben entre comillas dobles, por ejemplo `"malo"` es una cadena de 5 elementos con el último elemento el carácter nulo.

Especificador: %s

```
char M[5] = "malo"
```

M	A	L	O	\0
---	---	---	---	----

E/S de datos

FUNCION	DESCRIPCION
scanf ()	<p>Funcion que nos permite ingresar un dato al computador:</p> <pre>scanf ("%d", &var) ;</pre> <p>Donde %d es el especificador de tipo de dato. &var indica en que variable se guardara.</p>
printf ()	<p>Nos permite imprimir resultados y mensajes en la pantalla:</p> <pre>printf(" su edad es : %d \n", var);</pre> <p>Donde %d es el especificador de tipo de dato.</p>

Ejemplo 1 – Tipos de datos

```
#include <stdio.h>

int main()
{
    char    character = 'c';
    char    cadena[20] = "Hello World";
    float    floatP = 10.234;
    int    entero = 10;
    double    doubleP = 20;

    printf("Caracter: %c \n", character);
    printf("Cadena: %s \n" , cadena);
    printf("Float: %f \n", floatP);
    printf("Entero: %d \n" , entero);
    printf("Double: %f \n", doubleP);
    printf("Valor del entero en hexa: %x \n", entero);

    return 0;
}
```

Instrucciones de control

- Las instrucciones de control (if, else, else if, while, for, etc...) se escriben con la misma sintaxis y lógica que en Java.
- No existe tipo Booleano, se usan enteros
 - = 1 → TRUE
 - = 0 → FALSE

Funciones

- Las funciones son bloques de código que pueden ser llamados desde diferentes partes del programa.
- Su estructura general es:

```
tipo Nombre(Lista de parámetros)
{
    estructura interna

    return;
}
```


Funciones

EJEMPLO:

```
int suma(int a, int b)
{
    int z;
    z=a+b;
    return z;
}
```

```
.
.
int c=34;
int e = 4;

int r = suma(c,e);

printf("%d",r);
.
.
```

Estructuras

Una estructura la podemos definir como un nuevo tipo de datos el cual reúne a su vez variables de diferentes tipos de datos, bajo un mismo nombre.

El formato general de una estructura es:

```
struct nombre {  
    tipo variable1;  
    tipo variable2;  
    :  
    :  
    tipo variablen;  
} Variables;
```

Estructuras

EJEMPLO:

```
struct alumno{  
    char nombre[40];  
    char apellido[40];  
    unsigned int codigo;  
    int curso;  
    char carrera[40];  
}persona;
```

En este caso se crea una nueva variable de tipo alumno llamada persona.

Acceso a estructuras

Para referenciar a un elemento de la estructura alumno lo podemos hacer así:

```
persona.nombre (directo)  
personaptr -> nombre (indirecto)
```

El acceso indirecto se usa cuando se tienen apuntadores a estructuras.

Ejemplo 2 – Mayúsculas a Minúsculas

- Programa que reciba un nombre y cambie las mayúsculas a minúsculas y viceversa.

M = m -
32

Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
64	40	100	@	@	96	60	140	`	`
65	41	101	A	A	97	61	141	a	a
66	42	102	B	B	98	62	142	b	b
67	43	103	C	C	99	63	143	c	c
68	44	104	D	D	100	64	144	d	d
69	45	105	E	E	101	65	145	e	e
70	46	106	F	F	102	66	146	f	f
71	47	107	G	G	103	67	147	g	g
72	48	110	H	H	104	68	150	h	h
73	49	111	I	I	105	69	151	i	i
74	4A	112	J	J	106	6A	152	j	j
75	4B	113	K	K	107	6B	153	k	k
76	4C	114	L	L	108	6C	154	l	l
77	4D	115	M	M	109	6D	155	m	m
78	4E	116	N	N	110	6E	156	n	n
79	4F	117	O	O	111	6F	157	o	o
80	50	120	P	P	112	70	160	p	p
81	51	121	Q	Q	113	71	161	q	q
82	52	122	R	R	114	72	162	r	r
83	53	123	S	S	115	73	163	s	s
84	54	124	T	T	116	74	164	t	t
85	55	125	U	U	117	75	165	u	u
86	56	126	V	V	118	76	166	v	v
87	57	127	W	W	119	77	167	w	w
88	58	130	X	X	120	78	170	x	x
89	59	131	Y	Y	121	79	171	y	y
90	5A	132	Z	Z	122	7A	172	z	z
91	5B	133	[[123	7B	173	{	{
92	5C	134	\	\	124	7C	174	|	
93	5D	135]]	125	7D	175	}	}
94	5E	136	^	^	126	7E	176	~	~
95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

Ejemplo 2 – Invertir capitalización de caracteres

```
#include <stdio.h> #include <stdlib.h>

unsigned char var[10];
int main(void)
{
    printf("Por favor ingrese su nombre:");
    scanf("%s", &var);
    for(int i =0; i< 10; i++)
    {
        if(var[i]!=0)
        {
            if (var[i] < 91) //Mayuscula
                var[i]= var[i]+32;
            else //Minuscula
                var[i]= var[i]-32;
        }
    }
    printf("%s",var);
    scanf("%s", &var);
}
```

Ejercicio

- Programa que reciba por pantalla nombre y edad de una persona y responda si puede tomar o no.

Ejercicio - Solución

```
#include <stdio.h>

int edad;
unsigned char nombre[20];
int telefono;
char puedeTomar;

#define edadMin 18

int main(void)
{
    printf("Por favor ingrese su nombre: ");
    scanf("%s", &nombre);
    printf("Por favor ingrese su edad: ");
    scanf("%d", &edad);
    printf("Por favor ingrese su numero telefonico:");
    scanf("%d", &telefono);
    if (edad < edadMin)
    {
        printf("No puede tomar (edad es %d)", edad);
    }
    else
    {
        printf("Puede tomar (edad es %d)", edad);
    }
    return 0;
}
```