



Gestión de Estudiantes  
IPEC Santo Domingo de Heredia  
Por Jesús Alberto Moscoso Agüero

---

# Manual de Usuario Técnico

Versión 1

El programa es para gestionar, el código que cada estudiante tiene en el IPEC de Santo Domingo de Heredia.

El código es un identificador único de tipo alfanumérico, que gestiona la institución educativa para identificar a un determinado estudiante con respecto a trámites administrativos en el Ministerio de Educación Pública.

El programa permite ingresar registros de código para estudiantes nuevos, buscar un estudiante por código exacto, eliminar del registro un determinado estudiante y listar todos los códigos por estudiante presentes en la base de datos.

Se encuentra escrito en el lenguaje C++ y compilado con MinGW x64 bits. Se hace uso de la estructura de datos tipo listas doblemente enlazadas, para llevar el control de los registros de estudiantes.

## Tabla de contenido

---

Acerca del manual.....	2
Tabla de contenido .....	3
Tabla de imágenes .....	4
Requisitos del sistema operativo .....	5
Instalación .....	5
Descripción general de herramientas .....	7
Librerías del software.....	7
EstudiantesIPEC.cpp.....	8
Método “Menu()” .....	10
Método “EsValidoCodigoEstudiante()” .....	10
Método “Loop()” .....	10
Ingresar estudiante.....	11
Buscar estudiante .....	12
Eliminar estudiante .....	14
Listar todos los estudiantes .....	16
Salir del programa.....	17
Método “main()” .....	18
Clase “CuentaEstudiante” .....	18
ListaDoble.hpp .....	20
Clase “Nodo” .....	21
Clase “ListaDoble” .....	22

## Tabla de imágenes

---

Imagen 1. Ejecutable principal.....	5
Imagen 2. Mensaje de Error. Librería libgcc_s_seh-1.dll.....	6
Imagen 3. Mensaje de Error. Librería libstdc++-6.dll.....	6
Imagen 4. Pantalla de bienvenida.....	7
Imagen 5. Opción 1. Opciones del menú. ....	11
Imagen 6. Opción 1. Ingresar estudiante. ....	12
Imagen 7. Opción 2. Opciones del menú. ....	13
Imagen 8. Opción 2. Buscar estudiante. Valor encontrado. ....	13
Imagen 9. Opción 2. Buscar estudiante. Valor no encontrado. ....	14
Imagen 10. Opción 3. Opciones del menú. ....	14
Imagen 11. Opción 3. Eliminar estudiante. Valor encontrado. ....	15
Imagen 12. Opción 3. Eliminar estudiante. Valor no encontrado. ....	16
Imagen 13. Opción 4. Opciones del menú. ....	16
Imagen 14. Opción 4. Listar todos los estudiantes. ....	17
Imagen 15. Opción 5. Opciones del menú. ....	17

## Requisitos del sistema operativo

---

Windows x64 bits.

Requiere un procesador de 64 bits.

Al menos 50 MB de memoria dedicada para el programa.

Requiere 162 KB de espacio en disco duro.

No requiere conexión a Internet.

Debian-Linux x64 bits.

Requiere un procesador de 64 bits.

Al menos 50 MB de memoria dedicada para el programa.

Requiere 51 KB de espacio en disco duro.

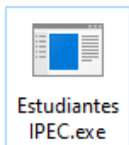
No requiere conexión a Internet.

## Instalación

---

Windows x64 bits.

1. Ejecutar sin permisos de administrador, el archivo: “EstudiantesIPEC.exe”



*Imagen 1. Ejecutable principal.*

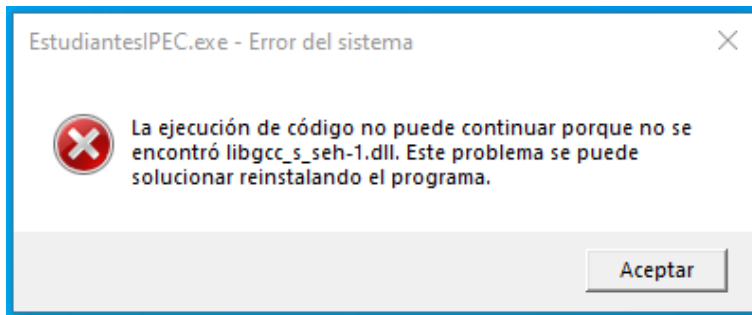
2. El sistema operativo debe tener instalado 2 librerías necesarias, para ejecutar de forma correcta el programa.



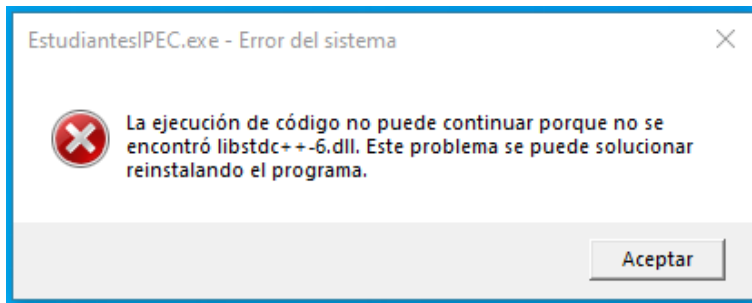
Librería. libgcc\_s\_seh-1.dll

Librería. libstdc++-6.dll

En caso de no contar con esos archivos, se deben descargar y mover a la carpeta “C:\Windows\System32”.



*Imagen 2. Mensaje de Error. Librería libgcc\_s\_seh-1.dll.*



*Imagen 3. Mensaje de Error. Librería libstdc++-6.dll.*

## Descripción general de herramientas

---

El programa es desarrollado por consola, al ser ejecutado por el usuario. Se muestra un banner y con opciones generales del software.

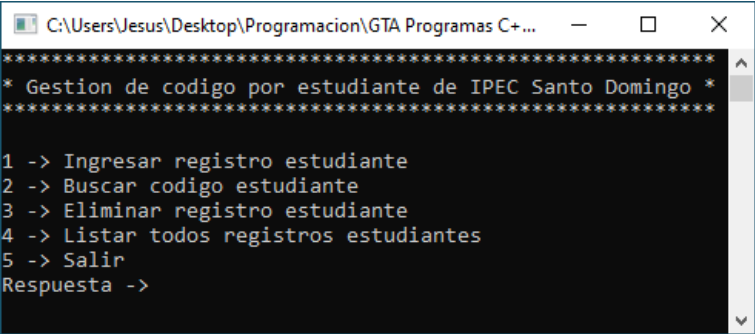


Imagen 4. Pantalla de bienvenida.

El usuario puede:

1. Ingresar registro de nuevo estudiante.
2. Buscar por código de estudiante.
3. Eliminar registro por código de estudiante.
4. Listar todos los registros de estudiantes.
5. Salir del programa.

## Librerías del software

---

### Librerías externas.

El software utiliza las siguientes librerías:

1. Librería “iostream”. Es un componente de la biblioteca estándar (STL) del lenguaje de programación C++ que es utilizado para operaciones de entrada/salida.

Funciones de la librería a utilizar.	
cin	Flujo de entrada. Pedir caracteres por teclado.
cout	Flujo de salida. Imprimir caracteres por pantalla.

2. Librería “string.h”. Es un archivo de la biblioteca estándar (STL) del lenguaje de programación C que contiene la definición de macros, constantes, funciones y algunas operaciones de manipulación de memoria.

Macros de la librería a utilizar.	
NULL	Representa la constante puntero nulo; un valor de puntero que no apunta a ninguna dirección válida de objeto alguno en memoria.

Funciones de la librería a utilizar.	
strcmp	Comparar dos cadenas alfabéticamente ('a' != 'A').
strcpy	Copiar una cadena en otra.
strlen	Devolver la longitud de una cadena.

Librerías creadas.

El software utiliza las siguientes librerías:

1. Librería “ListaDoble.hpp”. Gestionar una lista doblemente enlazada, para almacenar una variable estándar o un objeto creado por medio de la plantilla template <class T>.

Clases de la librería a utilizar.	
Clase “Nodo”	Guardar el objeto T mediante la memoria dinámica y dos punteros del mismo tipo. Puntero siguiente. Puntero anterior.
Clase “ListaDoble”	Ingresar, buscar, imprimir y eliminar de la lista, objetos T por medio de tres punteros de la clase Nodo. Puntero primerNodo. Puntero ultimoNodo. Puntero actual.

EstudiantesIPEC.cpp

---

Librerías.

El archivo importa las siguientes librerías:

Librerías a utilizar.
-----------------------



Librería “iostream”	Es un componente de la biblioteca estándar (STL) del lenguaje de programación C++ que es utilizado para operaciones de entrada/salida.
Librería “ListaDoble.hpp”	Gestionar una lista doblemente enlazada, para almacenar una variable estándar o un objeto creado por medio de la plantilla template <class T>.

### Variables.

El archivo utiliza las siguientes variables globales:

Tipo de variable	Identificador	Descripción
char	selected_option	Almacenar la opción seleccionada del menú opciones.
bool	loop	Bandera para el control del loop del programa principal.
ListaDoble<CuentaEstudiante>	listaRegEstudiantes	Almacenar la lista de registros de estudiantes.

### Métodos.

El archivo utiliza los siguientes métodos:

Identificador de Método	Retorno	Parámetro	Descripción
Menu()	void	void	Imprimir el menú de opciones principales y retornar opción que se selecciona.
EsValidoCodigoEstudiante()	bool	char[]	Validar si es correcto el código de registro del estudiante.
Loop()	void	void	Loop del programa principal.
main()	int	int, char const *[]	Función del control principal del software.

### Clases.

El archivo utiliza las siguientes clases:

Clases a utilizar.	
Clase “CuentaEstudiante”	Almacenar el código y nombre del estudiante.

## Método “Menu()”

---

### Descripción del método.

Imprimir el menú de opciones principales y retornar opción que se selecciona.

### Funcionamiento.

1. Imprimir las 5 opciones principales del software.
2. Obtener el carácter por teclado.
3. Almacenar el carácter en la variable global “selected\_option”.

Opciones del menú.

- 1 -> Ingresar registro estudiante.
- 2 -> Buscar codigo estudiante.
- 3 -> Eliminar registro estudiante.
- 4 -> Listar todos registros estudiantes.
- 5 -> Salir.

## Método “EsValidoCodigoEstudiante()”

---

### Descripción del método.

Validar si es correcto el código de registro del estudiante.

### Funcionamiento.

1. Retornar True. No se implementa código.

## Método “Loop()”

---

### Descripción del método.

Loop del programa principal.

### Funcionamiento.

1. Llamar al método “Menu()” para imprimir las opciones del programa.
2. Comparar mediante if, la variable global “selected\_option” con el carácter respectivo de la funcionalidad.
3. Ejecutar los métodos necesarios para esa opción que se seleccionó del menú.

Ingresa estudiante

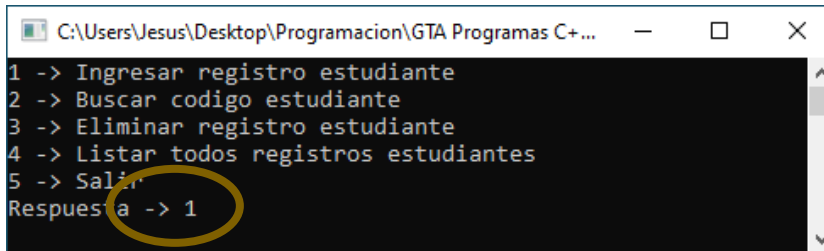
---

### Funcionamiento.

Se obtiene mediante variables locales el nombre y código del estudiante. Luego se crea un objeto “CuentaEstudiante” con las variables locales y se procede a ingresar al final de la lista, el registro del estudiante; mediante la llamada al método “ingresarFinal” del objeto “listaRegEstudiantes”.

### Usuario.

1. Para ingresar un nuevo estudiante en la institución, se debe escribir como respuesta la letra “1”, sin guiones y sin espacios. Para ejecutar el comando, presionar la tecla “entrar” de su teclado.



```
C:\Users\Jesus\Desktop\Programacion\GTA Programas C+...
1 -> Ingresar registro estudiante
2 -> Buscar codigo estudiante
3 -> Eliminar registro estudiante
4 -> Listar todos registros estudiantes
5 -> Salir
Respuesta -> 1
```

Imagen 5. Opción 1. Opciones del menú.

2. Escribir el nombre completo del estudiante y presionar la tecla “entrar” de su teclado.

Ingresa nombre del estudiante a crear -> Jesus Alberto Moscoso Agüero

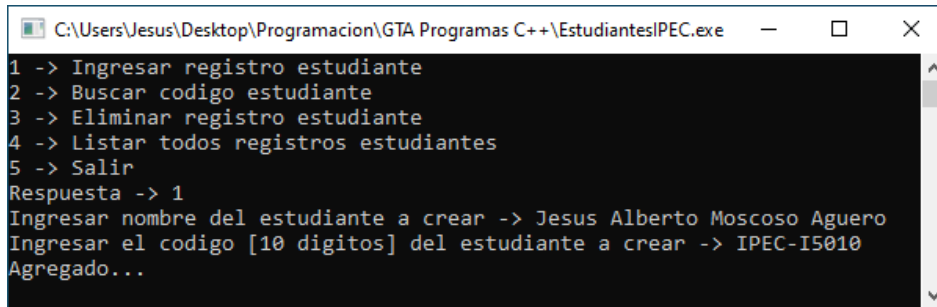
3. Escribir el código del estudiante y presionar la tecla “entrar” de su teclado. El código debe ser una cifra de 10 dígitos de tipo alfanumérico.

Ingresar el codigo [10 digitos] del estudiante a crear -> IPEC-I5010

4. Si los datos son correctos y almacenados en el registro de la institución, se muestra el mensaje de confirmación “Agregado...”. En caso contrario debe volver a introducir el código del estudiante.

Agregado...

Ejemplo de ejecución del programa.



```
C:\Users\Jesus\Desktop\Programacion\GTA Programas C++\EstudiantesIPEC.exe
1 -> Ingresar registro estudiante
2 -> Buscar codigo estudiante
3 -> Eliminar registro estudiante
4 -> Listar todos registros estudiantes
5 -> Salir
Respuesta -> 1
Ingresar nombre del estudiante a crear -> Jesus Alberto Moscoso Aguero
Ingresar el codigo [10 digitos] del estudiante a crear -> IPEC-I5010
Agregado...
```

Imagen 6. Opción 1. Ingresar estudiante.

## Buscar estudiante

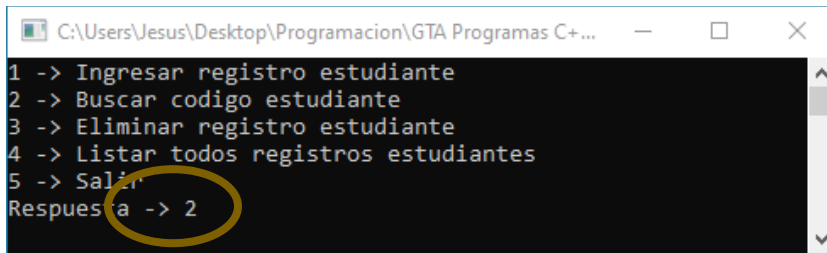
---

### Funcionamiento.

Se obtiene mediante variable local el código del estudiante. Luego se crea un objeto “CuentaEstudiante” con la variable local y se procede a buscar en la lista, el registro del estudiante; mediante la llamada al método “buscar” del objeto “listaRegEstudiantes”.

### Usuario.

1. Para buscar un estudiante en la institución, se debe escribir como respuesta la letra “2”, sin guiones y sin espacios. Para ejecutar el comando, presionar la tecla “entrar” de su teclado.



```
C:\Users\Jesus\Desktop\Programacion\GTA Programas C+...
1 -> Ingresar registro estudiante
2 -> Buscar codigo estudiante
3 -> Eliminar registro estudiante
4 -> Listar todos registros estudiantes
5 -> Salir
Respuesta -> 2
```

Imagen 7. Opción 2. Opciones del menú.

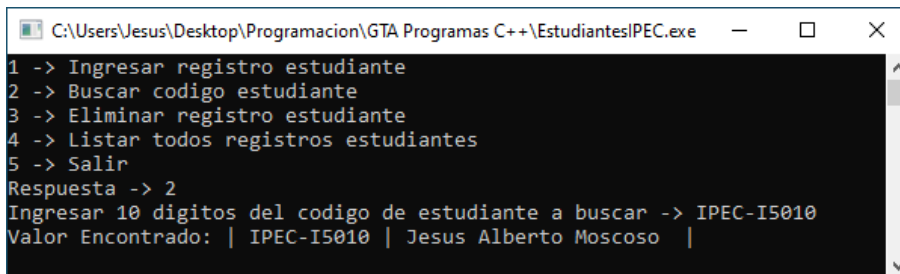
2. Escribir el código de 10 dígitos del estudiante a buscar y presionar la tecla “entrar” de su teclado.

```
Ingresar 10 digitos del codigo de estudiante a buscar -> IPEC-I5010
```

3. Si el código del estudiante se encuentra con éxito, se muestra el mensaje de confirmación “Valor Encontrado”, seguido del respectivo código y nombre del estudiante.

```
Valor Encontrado: | IPEC-I5010 | Jesus Alberto Moscoso |
```

Ejemplo de ejecución del programa.



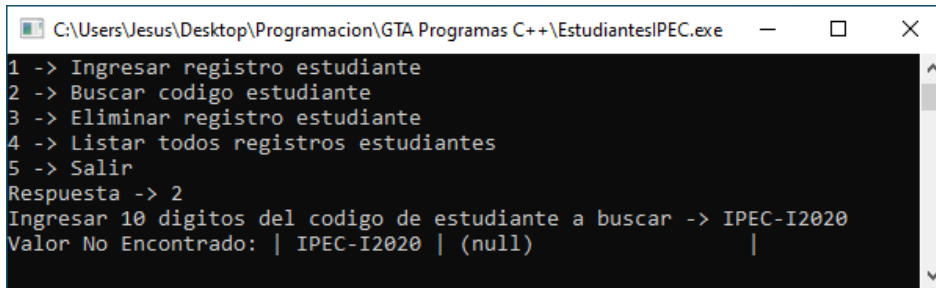
```
C:\Users\Jesus\Desktop\Programacion\GTA Programas C++\EstudiantesIPEC.exe
1 -> Ingresar registro estudiante
2 -> Buscar codigo estudiante
3 -> Eliminar registro estudiante
4 -> Listar todos registros estudiantes
5 -> Salir
Respuesta -> 2
Ingresar 10 digitos del codigo de estudiante a buscar -> IPEC-I5010
Valor Encontrado: | IPEC-I5010 | Jesus Alberto Moscoso |
```

Imagen 8. Opción 2. Buscar estudiante. Valor encontrado.

4. En caso contrario, si el código del estudiante no se encuentra; se muestra el mensaje de confirmación “Valor No Encontrado” con el respectivo código que se busca.

```
Valor No Encontrado: | IPEC-I2020 | (null)
```

Ejemplo de ejecución del programa.



```
C:\Users\Jesus\Desktop\Programacion\GTA Programas C++\EstudiantesIPEC.exe
1 -> Ingresar registro estudiante
2 -> Buscar codigo estudiante
3 -> Eliminar registro estudiante
4 -> Listar todos registros estudiantes
5 -> Salir
Respuesta -> 2
Ingresar 10 digitos del codigo de estudiante a buscar -> IPEC-I2020
Valor No Encontrado: | IPEC-I2020 | (null)
```

Imagen 9. Opción 2. Buscar estudiante. Valor no encontrado.

## Eliminar estudiante

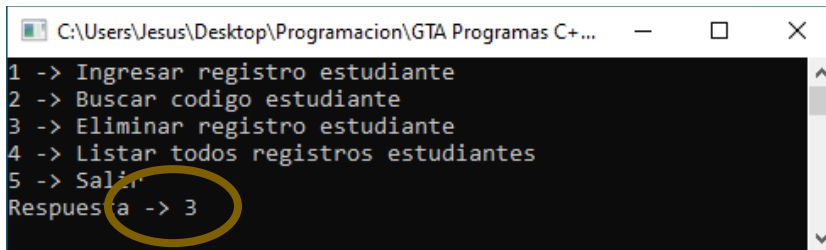
---

### Funcionamiento.

Se obtiene mediante variable local el código del estudiante. Luego se crea un objeto “CuentaEstudiante” con la variable local y se procede a buscar de la lista, el registro del estudiante; mediante la llamada al método “buscar” del objeto “listaRegEstudiantes”. Se guarda en una bandera booleana si se encontró o no el registro. Si se encontró se procede a eliminar el registro; mediante la llamada al método “eliminarActual” del objeto “listaRegEstudiantes”. y sino imprime en pantalla el mensaje “Valor No Eliminado”.

### Usuario.

1. Para eliminar un estudiante en la institución, se debe escribir como respuesta la letra “3”, sin guiones y sin espacios. Para ejecutar el comando, presionar la tecla “entrar” de su teclado.



```
C:\Users\Jesus\Desktop\Programacion\GTA Programas C+...
1 -> Ingresar registro estudiante
2 -> Buscar codigo estudiante
3 -> Eliminar registro estudiante
4 -> Listar todos registros estudiantes
5 -> Salir
Respuesta -> 3
```

Imagen 10. Opción 3. Opciones del menú.

2. Escribir el código de 10 dígitos del estudiante a eliminar y presionar la tecla “entrar” de su teclado.

```
Ingresa 10 digitos del codigo de estudiante a eliminar -> IPEC-I5010
```

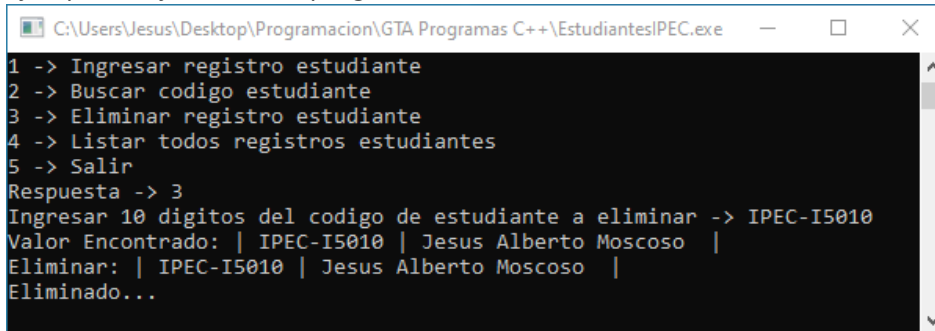
3. Si el código del estudiante se encuentra con éxito, se muestra el mensaje de confirmación “Valor Encontrado”, seguido del respectivo código y nombre del estudiante.

```
Valor Encontrado: | IPEC-I5010 | Jesus Alberto Moscoso |
```

4. Si el código del estudiante se eliminó con éxito, se muestra el mensaje de confirmación “Eliminado...”.

```
Eliminar: | IPEC-I5010 | Jesus Alberto Moscoso |  
Eliminado...
```

Ejemplo de ejecución del programa.



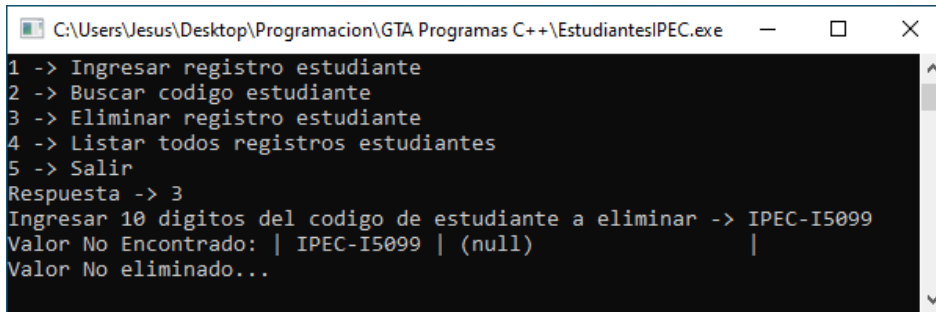
```
C:\Users\Jesus\Desktop\Programacion\GTA Programas C++\Estudiantes\IPEC.exe  
1 -> Ingresar registro estudiante  
2 -> Buscar codigo estudiante  
3 -> Eliminar registro estudiante  
4 -> Listar todos registros estudiantes  
5 -> Salir  
Respuesta -> 3  
Ingresa 10 digitos del codigo de estudiante a eliminar -> IPEC-I5010  
Valor Encontrado: | IPEC-I5010 | Jesus Alberto Moscoso |  
Eliminar: | IPEC-I5010 | Jesus Alberto Moscoso |  
Eliminado...
```

Imagen 11. Opción 3. Eliminar estudiante. Valor encontrado.

5. En caso contrario, si el código del estudiante no se encuentra y no se eliminó; se muestra el mensaje de confirmación “Valor No Encontrado” con el respectivo código que se busca y el mensaje de confirmación “Valor No Eliminado...”.

```
Valor No Encontrado: | IPEC-I5099 | (null) |  
Valor No eliminado...
```

Ejemplo de ejecución del programa.



```
C:\Users\Jesus\Desktop\Programacion\GTA Programas C++\EstudiantesIPEC.exe
1 -> Ingresar registro estudiante
2 -> Buscar codigo estudiante
3 -> Eliminar registro estudiante
4 -> Listar todos registros estudiantes
5 -> Salir
Respuesta -> 3
Ingresar 10 digitos del codigo de estudiante a eliminar -> IPEC-IS099
Valor No Encontrado: | IPEC-IS099 | (null)
Valor No eliminado...
```

Imagen 12. Opción 3. Eliminar estudiante. Valor no encontrado.

## Listar todos los estudiantes

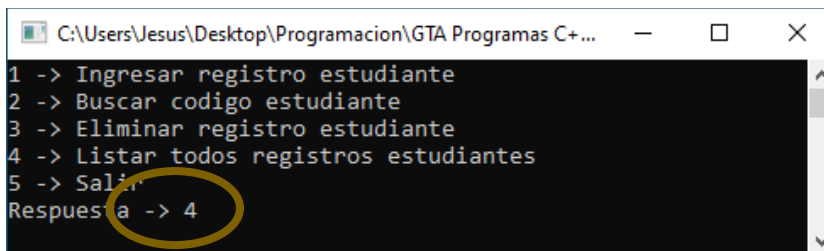
---

### Funcionamiento.

Se procede a llamar el método de imprimir toda la lista, para imprimir cada registro de estudiante; mediante la llamada al método “imprimirTodo” del objeto “listaRegEstudiantes”.

### Usuario.

1. Para eliminar un estudiante en la institución, se debe escribir como respuesta la letra “4”, sin guiones y sin espacios. Para ejecutar el comando, presionar la tecla “entrar” de su teclado.



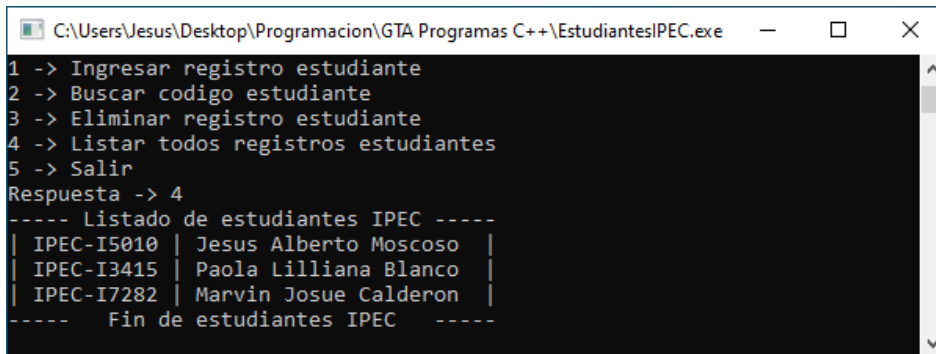
```
C:\Users\Jesus\Desktop\Programacion\GTA Programas C+...
1 -> Ingresar registro estudiante
2 -> Buscar codigo estudiante
3 -> Eliminar registro estudiante
4 -> Listar todos registros estudiantes
5 -> Salir
Respuesta -> 4
```

Imagen 13. Opción 4. Opciones del menú.

2. Se imprime el listado de estudiantes.

Ejemplo de ejecución del programa.





```
C:\Users\Jesus\Desktop\Programacion\GTA Programas C++\EstudiantesIPEC.exe
1 -> Ingresar registro estudiante
2 -> Buscar codigo estudiante
3 -> Eliminar registro estudiante
4 -> Listar todos registros estudiantes
5 -> Salir
Respuesta -> 4
----- Listado de estudiantes IPEC -----
| IPEC-I5010 | Jesus Alberto Moscoso |
| IPEC-I3415 | Paola Lilliana Blanco |
| IPEC-I7282 | Marvin Josue Calderon |
----- Fin de estudiantes IPEC -----
```

Imagen 14. Opción 4. Listar todos los estudiantes.

## Salir del programa

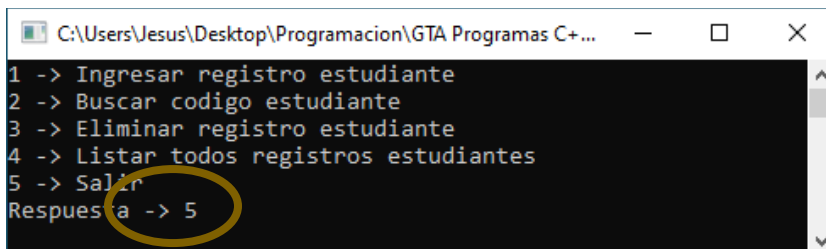
---

### Funcionamiento.

Coloca la variable global “loop” en el valor “false”, para salir del bucle principal del programa y luego salir del software en ejecución.

### Usuario.

1. Para salir del programa, se debe escribir como respuesta la letra “5”, sin guiones y sin espacios. Para ejecutar el comando, presionar la tecla “entrar” de su teclado.



```
C:\Users\Jesus\Desktop\Programacion\GTA Programas C+...
1 -> Ingresar registro estudiante
2 -> Buscar codigo estudiante
3 -> Eliminar registro estudiante
4 -> Listar todos registros estudiantes
5 -> Salir
Respuesta -> 5
```

Imagen 15. Opción 5. Opciones del menú.

## Método “main()”

---

### Descripción del método.

Función del control principal del software.

### Funcionamiento.

Imprimir el banner de la descripción sobre el software y proceder a llamar al método “Loop()” para controlar el loop principal del programa.

## Clase “CuentaEstudiante”

---

### Descripción de la clase.

Almacenar el código del estudiante mediante un puntero a char y nombre del estudiante mediante un puntero a char. La clase implementa constructor, destructor, constructor copia, sobrecarga del operador “=”, sobrecarga del operador “==” y sobrecarga del operador “<<”

### Atributos.

El archivo utiliza las siguientes variables:

Tipo de variable	Ámbito	Identificador	Descripción
char*	privado	codigo	Almacenar el código del estudiante.
char*	privado	nombreEstudiante	Almacenar el nombre del estudiante.

### Métodos.

La clase utiliza los siguientes métodos:

Identificador de Método	Retorno	Parámetro	Descripción
CuentaEstudiante()	void	char *c_codigo,	Constructor de la clase. Copiar el

		char *c_nombreEstudiante	código y nombre del estudiante por referencia de parámetros en las variables privadas de la clase.
~CuentaEstudiante()	void	void	Destructor de la clase. Llamar la función delete del manejo de memoria dinámica para destruir los datos de las variables privadas de la clase.
CuentaEstudiante()	void	const class CuentaEstudiante &clon	Constructor de copia. Copiar la instancia del objeto por referencia a un nuevo objeto. Ejecuta el método de sobrecarga operator “=”.
operator=	CuentaEstudiante&	const class CuentaEstudiante &clon	Sobrecarga del método “=”. Ejecuta un strcpy de la librería string.h para realizar la copia de las variables del objeto por referencia a las variables privadas.
operator==	bool	const class CuentaEstudiante &c const	Sobrecarga del método “==”. Ejecuta un strcmp de la librería string.h para comparar si son iguales la variable privada “codigo” con respecto a la misma variable del objeto por referencia.
getCodigo()	char*	void	Método Get de la variable privada “codigo”.
getNombreEstudiante()	char*	void	Método Get de la variable privada “nombreEstudiante”.
operator<<	ostream&	ostream &os, const class CuentaEstudiante &c	Sobrecarga del método “<<”. Ejecuta la impresión por pantalla de las variables “codigo” y “nombreEstudiante” mediante su metodo Get del objeto por referencia.

### Librerías.

El archivo importa las siguientes librerías:

Librerías a utilizar.	
Librería “iostream”	Es un componente de la biblioteca estándar (STL) del lenguaje de programación C++ que es utilizado para operaciones de entrada/salida.
Librería “string.h”	Es un archivo de la biblioteca estándar (STL) del lenguaje de programación C que contiene la definición de macros, constantes, funciones y algunas operaciones de manipulación de memoria.

### Variables.

El archivo no utiliza variables globales.

### Métodos.

El archivo no utiliza métodos.

### Clases.

El archivo utiliza las siguientes clases:

Clases a utilizar.	
Clase “Nodo”	Almacenar el objeto T mediante la memoria dinámica y dos punteros del mismo tipo. Puntero siguiente. Puntero anterior.
Clase “ListaDoble”	Ingresar, buscar, imprimir y eliminar de la lista, objetos T por medio de tres punteros de tipo clase Nodo. Puntero primerNodo. Puntero ultimoNodo. Puntero actual. Se realiza la estructura de datos, tipo lista doblemente enlazada.

## Clase “Nodo”

---

### Descripción de la clase.

Almacenar el objeto T mediante la memoria dinámica y dos punteros del mismo tipo. Puntero siguiente.  
Puntero anterior.

### Atributos.

El archivo utiliza las siguientes variables:

Tipo de variable	Ámbito	Identificador	Descripción
template <class T>*	privado	dato	Almacenar el objeto por puntero para memoria dinámica.
class Nodo*	publico	sig	Almacenar un puntero del mismo tipo de clase, para el siguiente valor de la lista.
class Nodo*	publico	ant	Almacenar un puntero del mismo tipo de clase, para el anterior valor de la lista.

### Métodos.

La clase utiliza los siguientes métodos:

Identificador de Método	Retorno	Parámetro	Descripción
Nodo()	void	template <class T> dato, Nodo<T> *siguiente=NULL, Nodo<T> *anterior=NULL	Constructor de la clase. Inicializar una copia el objeto T por memoria dinámica a la variable de tipo T dato. Asignar el puntero sig y ant por referencia.
~Nodo()	void	void	Destructor de la clase. Llamar la función delete del manejo de memoria dinámica para destruir los datos del puntero dato. Asignar el puntero sig y ant a NULL.
friend class ListaDoble<T>	void	void	Asignar como clase amiga a la clase ListaDoble para usar las variables privadas de esta clase.

## Clase “ListaDoble”

---

### Descripción de la clase.

Ingresar, buscar, imprimir y eliminar de la lista, objetos T por medio de tres punteros de tipo clase Nodo. Puntero primerNodo. Puntero ultimoNodo. Puntero actual. Se realiza la estructura de datos, tipo lista doblemente enlazada.

### Atributos.

El archivo utiliza las siguientes variables:

Tipo de variable	Ámbito	Identificador	Descripción
class Nodo<T>*	privado	primerNodo	Almacenar el puntero al primer nodo de la estructura lista.
class Nodo<T>*	privado	ultimoNodo	Almacenar el puntero al último nodo de la estructura lista.
class Nodo<T>*	privado	actual	Almacenar el puntero actual como auxiliar para recorrer la estructura lista.

### Métodos.

La clase utiliza los siguientes métodos:

Identificador de Método	Retorno	Parámetro	Descripción
ListaDoble()	void	void	Constructor de la clase. Asignar a NULL todas las variables privadas tipo Nodo<T> Variable primerNodo, ultimoNodo, actual.
~ListaDoble()	void	void	Destructor de la clase. Llamar al miembro de la clase “eliminarActual” para eliminar los objetos restantes de la estructura lista.
esVacia()	bool	void	Retornar si la estructura lista tiene elementos.
moverActualPrimero()	void	void	Apuntar el puntero actual al primer nodo de la estructura lista.
ingresarFinal()	void	template <class T>& dato	Crear un nodo auxiliar con el dato por parámetro por referencia y agregarlo al final de

			la estructura lista. Al finalizar mueve el puntero actual al último nodo de la estructura lista.
eliminarActual()	void	void	Eliminar el nodo puntero actual de la estructura lista y actualizar los nodos anterior y siguiente. Al finalizar mueve el puntero actual al primer nodo de la estructura lista.
buscar()	bool	template <class T>& dato	Utilizar el puntero actual para recorrer desde el primer nodo hasta el último nodo toda la lista, en búsqueda de dato por parámetro por referencia. Imprimir en pantalla si se encuentra o no se encuentra el dato.
imprimirTodo()	void	void	Utilizar el puntero actual para recorrer del primer nodo hasta el último nodo toda la lista, e imprimir por pantalla el valor de cada nodo.