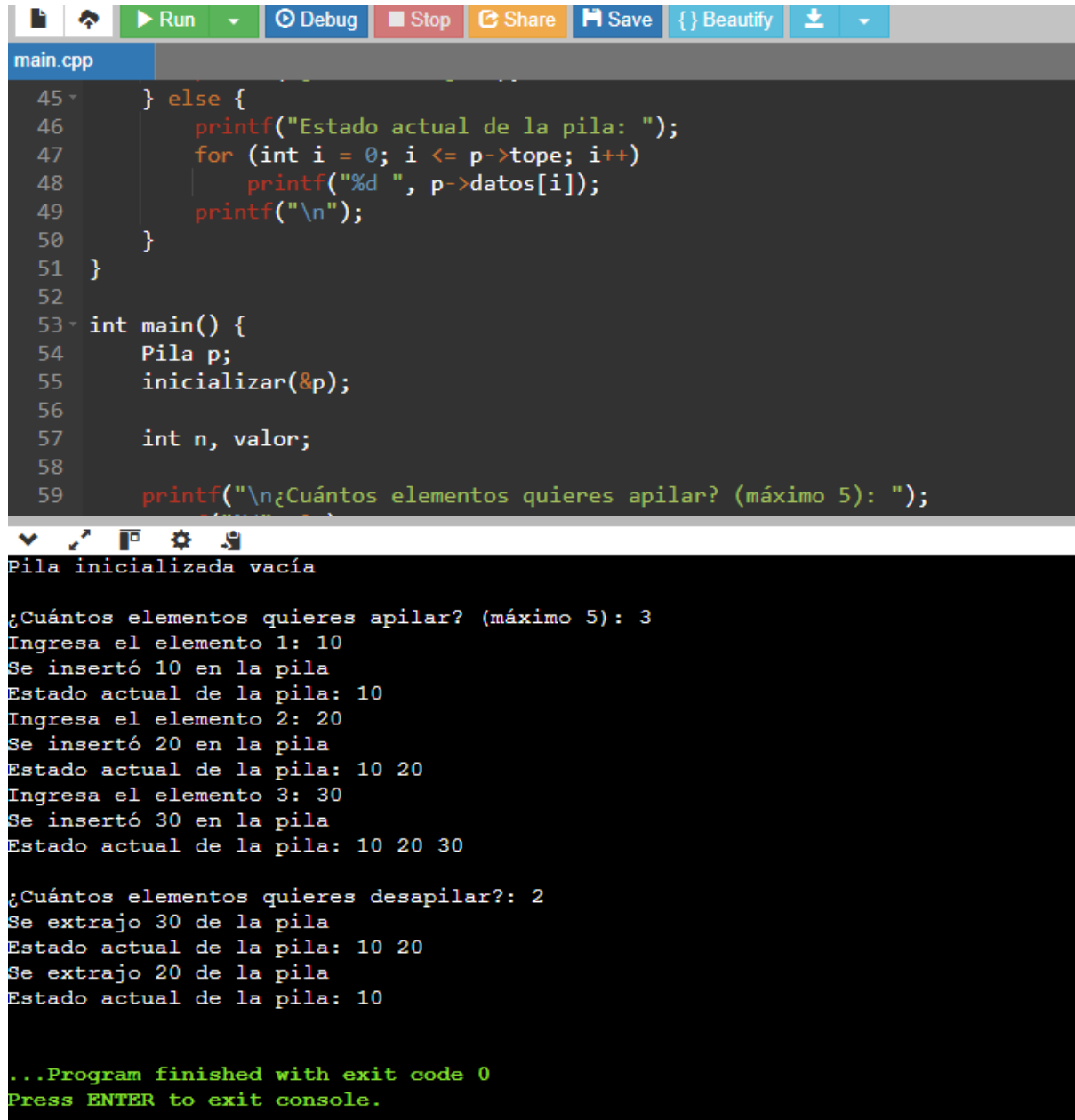


Ramirez Manriquez Luis Fernando

PUSH y POP



The image shows a C++ IDE with a file named `main.cpp`. The code implements a stack using an array. It includes functions for initialization, pushing elements, and popping elements. The `main` function demonstrates these operations by pushing three elements (10, 20, 30) and then popping two elements (30, 20).

```
45 } else {
46     printf("Estado actual de la pila: ");
47     for (int i = 0; i <= p->tope; i++)
48         printf("%d ", p->datos[i]);
49     printf("\n");
50 }
51 }
52
53 int main() {
54     Pila p;
55     inicializar(&p);
56
57     int n, valor;
58
59     printf("\n¿Cuántos elementos quieres apilar? (máximo 5): ");
```

The output of the program is as follows:

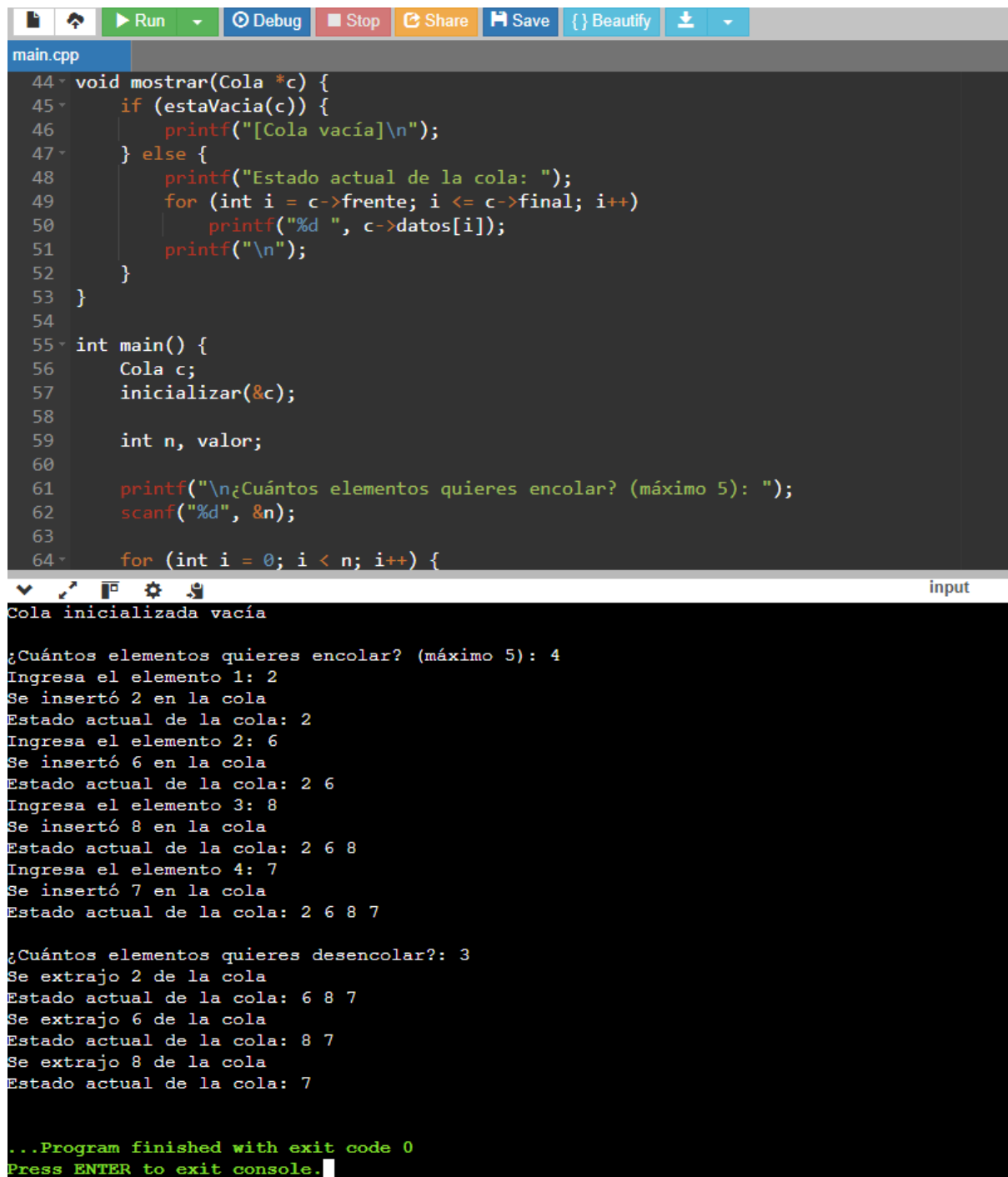
```
Pila inicializada vacía

¿Cuántos elementos quieres apilar? (máximo 5): 3
Ingresa el elemento 1: 10
Se insertó 10 en la pila
Estado actual de la pila: 10
Ingresa el elemento 2: 20
Se insertó 20 en la pila
Estado actual de la pila: 10 20
Ingresa el elemento 3: 30
Se insertó 30 en la pila
Estado actual de la pila: 10 20 30

¿Cuántos elementos quieres desapilar?: 2
Se extrajo 30 de la pila
Estado actual de la pila: 10 20
Se extrajo 20 de la pila
Estado actual de la pila: 10

...Program finished with exit code 0
Press ENTER to exit console.
```

## Enqueue y Dequeue



The image shows a C++ IDE with a code editor and a console window. The code defines a queue structure and functions to enqueue and dequeue elements. The console output shows the program's execution, including the initial state of the queue, the enqueue of four elements (2, 6, 8, 7), and the dequeue of three elements (2, 6, 8).

```
main.cpp
44 void mostrar(Cola *c) {
45     if (estaVacia(c)) {
46         printf("[Cola vacía]\n");
47     } else {
48         printf("Estado actual de la cola: ");
49         for (int i = c->frente; i <= c->final; i++)
50             printf("%d ", c->datos[i]);
51         printf("\n");
52     }
53 }
54
55 int main() {
56     Cola c;
57     inicializar(&c);
58
59     int n, valor;
60
61     printf("\n¿Cuántos elementos quieres encolar? (máximo 5): ");
62     scanf("%d", &n);
63
64     for (int i = 0; i < n; i++) {
```

Cola inicializada vacía

¿Cuántos elementos quieres encolar? (máximo 5): 4

Ingresa el elemento 1: 2

Se insertó 2 en la cola

Estado actual de la cola: 2

Ingresa el elemento 2: 6

Se insertó 6 en la cola

Estado actual de la cola: 2 6

Ingresa el elemento 3: 8

Se insertó 8 en la cola

Estado actual de la cola: 2 6 8

Ingresa el elemento 4: 7

Se insertó 7 en la cola

Estado actual de la cola: 2 6 8 7

¿Cuántos elementos quieres desencolar?: 3

Se extrajo 2 de la cola

Estado actual de la cola: 6 8 7

Se extrajo 6 de la cola

Estado actual de la cola: 8 7

Se extrajo 8 de la cola

Estado actual de la cola: 7

...Program finished with exit code 0

Press ENTER to exit console.

## Actividad

1. ¿Qué diferencias notaron entre el orden de salida de la pila y de la cola?

La diferencia está en el orden en que salen los datos. En la pila, el último que entra es el primero que sale, como cuando apilas platos y quitas el de arriba. En la cola, el primero que entra es el primero que sale, como cuando haces fila y atienden al que llegó primero. Por eso cada una funciona de manera distinta.

2. ¿Qué sucede si intentan hacer pop en una pila vacía o dequeue en una cola vacía?

Si intentas sacar algo cuando la pila o la cola están vacías, simplemente no se puede porque ya no hay nada. El programa debe avisar que está vacía para evitar errores o acciones que no tienen sentido.

3. ¿Qué sucede si intentan insertar más elementos que la capacidad máxima?

Cuando intentas meter más elementos de los que caben, la estructura ya no los acepta. El programa debe avisar que está llena y no permite agregar nada más, evitando que se “sobrepase” el límite.