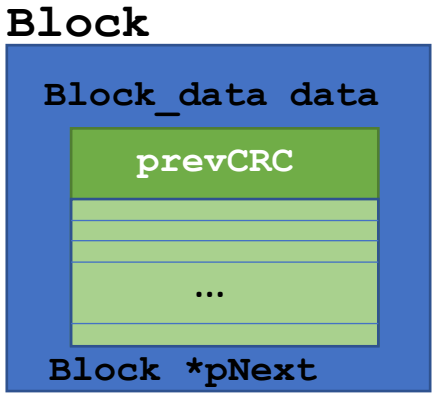
BlockChain

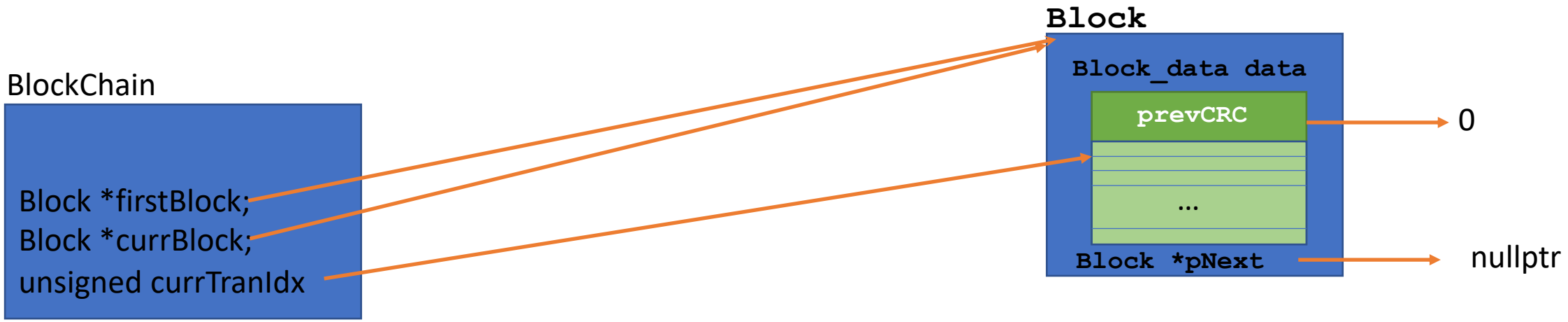
Block *firstBlock;
Block *currBlock;
unsigned currTranIdx

```cpp
class BlockChain
{
public:
    // Crea una nueva cadena de bloques y la inicializa como vacía
    BlockChain(); // TODO

    // destruye una cadena de bloques liberando todos
    // los recursos asociados
    ~BlockChain(); // TODO

    // Agrega una nueva transacción a la cadena
    void addTransaction(Transaction t);  // TODO

    // función que devuelve verdadero o falso indicando la validez de la cadena
    bool chainValid();  // TODO

private:
    Block* firstBlock;          // primer bloque de la cadena
    Block* currBlock;           // bloque en generación
    unsigned currTransacIdx;    // índice de transacción dentro de currBlock
};
```

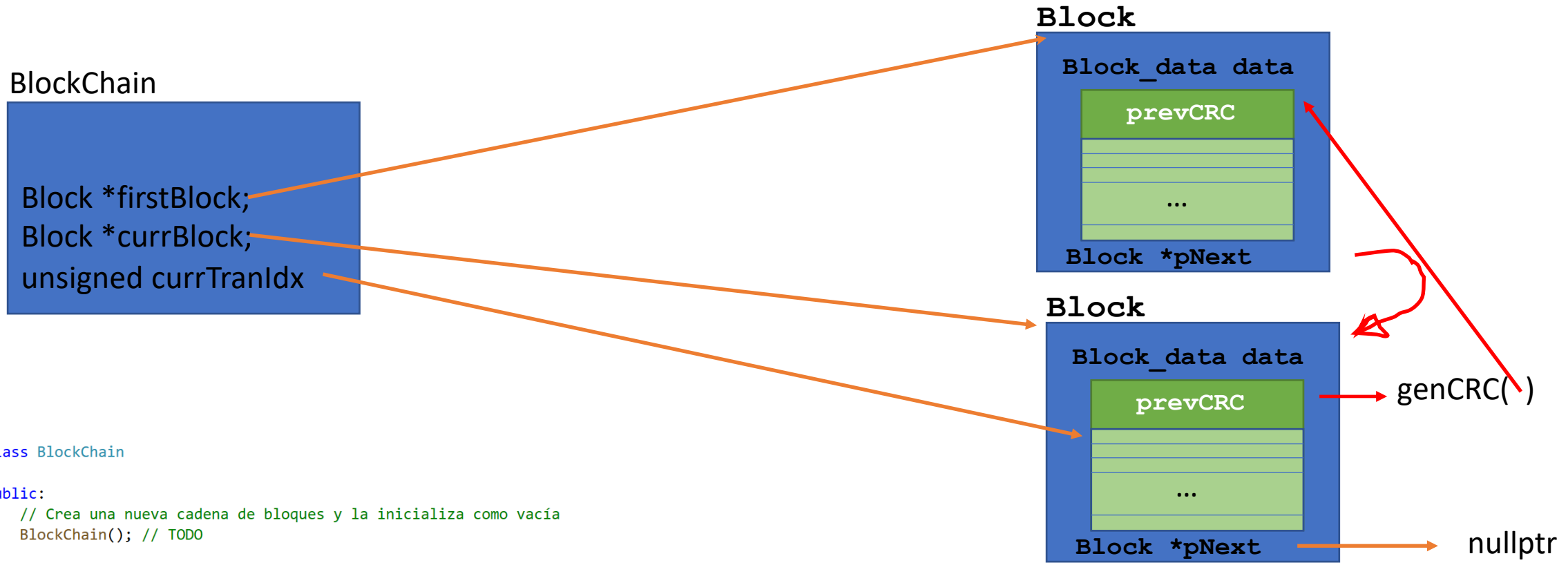
```cpp
struct Block_data
{
    static const unsigned N = 32;
    Block_data(unsigned pcrc) : prevCRC(pcrc) {}
    unsigned prevCRC;       // checksum del bloque previo, 0 si es el primer bloque
    Transaction transactions[N] = { 0 }; // transacciones del bloque
};
```

```cpp
struct Block
{
    Block(unsigned crc) : data(crc), pNext(nullptr) {}
    Block_data data;        // bloque de datos
    Block* pNext;           // siguiente bloque en la cadena
};
```

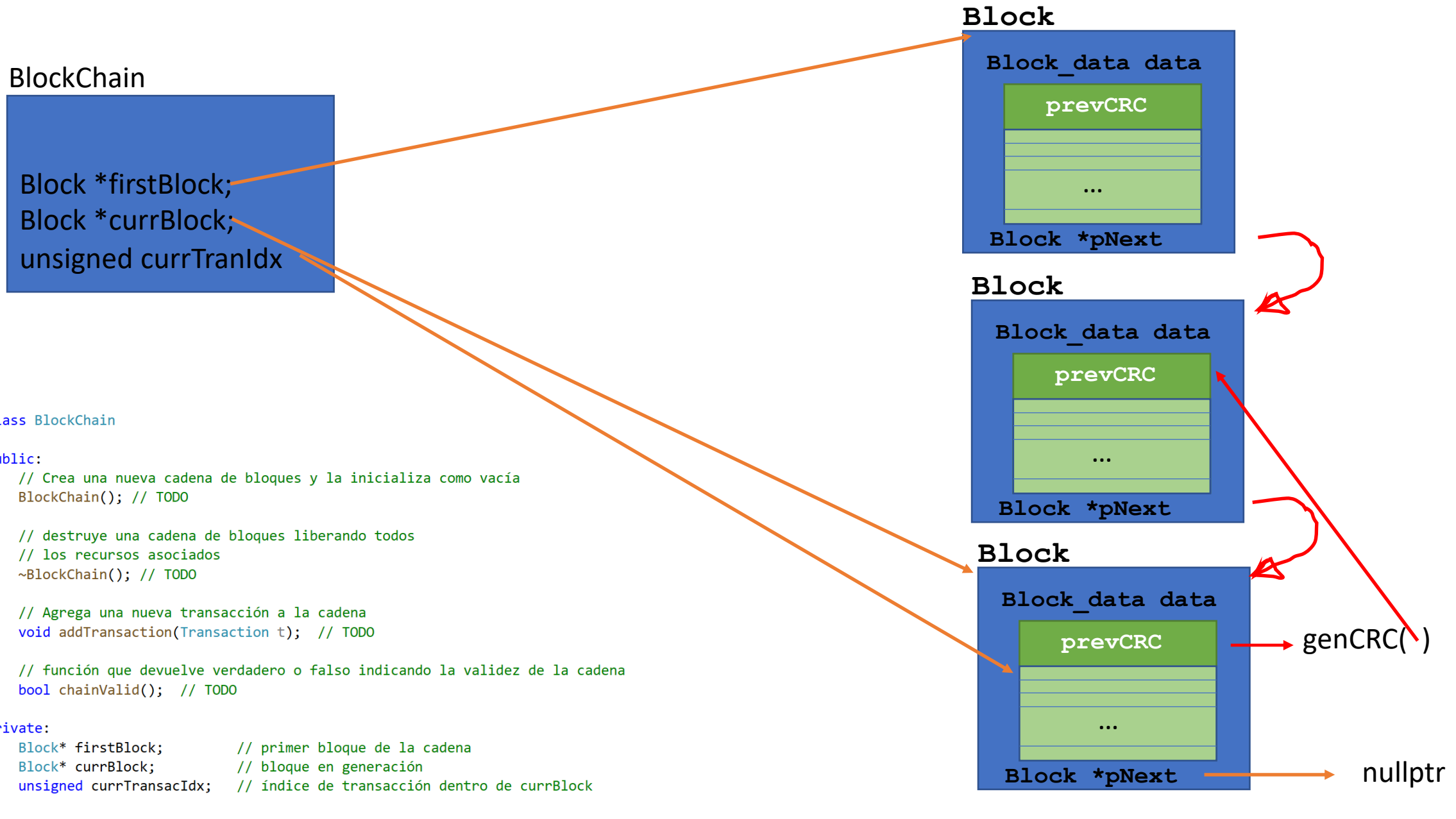





```
class BlockChain
{
public:
    // Crea una nueva cadena de bloques y la inicializa como vacía
    BlockChain(); // TODO

    // destruye una cadena de bloques liberando todos
    // los recursos asociados
    ~BlockChain(); // TODO

    // Agrega una nueva transacción a la cadena
    void addTransaction(Transaction t);  // TODO

    // función que devuelve verdadero o falso indicando la validez de la cadena
    bool chainValid();  // TODO

private:
    Block* firstBlock;          // primer bloque de la cadena
    Block* currBlock;           // bloque en generación
    unsigned currTransacIdx;    // índice de transacción dentro de currBlock
};
```




```
class BlockChain
{
public:
    // Crea una nueva cadena de bloques y la inicializa como vacía
    BlockChain(); // TODO

    // destruye una cadena de bloques liberando todos
    // los recursos asociados
    ~BlockChain(); // TODO

    // Agrega una nueva transacción a la cadena
    void addTransaction(Transaction t);  // TODO

    // función que devuelve verdadero o falso indicando la validez de la cadena
    bool chainValid();  // TODO

private:
    Block* firstBlock;          // primer bloque de la cadena
    Block* currBlock;           // bloque en generación
    unsigned currTransacIdx;    // índice de transacción dentro de currBlock
};
```




```
class BlockChain
{
public:
    // Crea una nueva cadena de bloques y la inicializa como vacía
    BlockChain(); // TODO

    // destruye una cadena de bloques liberando todos
    // los recursos asociados
    ~BlockChain(); // TODO

    // Agrega una nueva transacción a la cadena
    void addTransaction(Transaction t);  // TODO

    // función que devuelve verdadero o falso indicando la validez de la cadena
    bool chainValid();  // TODO

private:
    Block* firstBlock;          // primer bloque de la cadena
    Block* currBlock;           // bloque en generación
    unsigned currTransacIdx;    // índice de transacción dentro de currBlock
};
```