



Anexo 1

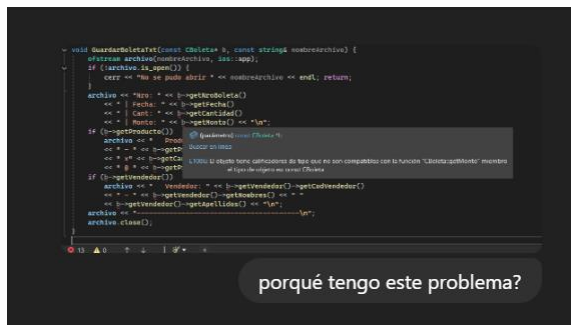
Bitácora del uso de IA



Curso: Algoritmos y Estructura de Datos
Ciclo 03
Evaluación TB1
NRC 1999

#	Fecha	Herramienta de IA	Qué consultamos	Resultado IA	Qué se cambió	Cómo se validó
		Chat GPT, Copilot, Gemini, Claude, etc.	Consulta o prompt	Resumen del contenido generado por la IA	Qué parte se aceptó, qué se descartó y por qué	Formas en que se comprobó el resultado (compilación, ejecución en Visual Studio, pruebas unitarias, explicación en clase).
1	2/10/25	Chat GPT				

Problema en función void GuardarBoletaTxt. Tengo problemas con el parámetro b:



RESULTADO:

Ese error (E1086) sale porque estás pasando un **puntero const** a CBoleta:

```
void GuardarBoletaTxt(const CBoleta* b, const string& nombreArchivo)
```

...y luego llamas a métodos como `b->getMonto()`.

Si en CBoleta esos getters **no están marcados como const**, el compilador no te deja invocarlos desde un objeto `const`.

Arreglo recomendado (const-correctness)

En **CBoleta.h** cambia las firmas de los getters (y, si tienes .cpp, también sus definiciones) para que sean `const`:

```
class CBoleta {  
    // ...  
public:  
    const string& getNroBoleta() const;  
    const string& getFecha() const;  
    int getCantidad() const;  
    float getMonto() const;
```

```

        // si expones relaciones:
        CProducto* getProducto() const;    // o mejor: const CProducto*
getProducto() const;
        CVendedor* getVendedor() const;  // o mejor: const CVendedor*
getVendedor() const;

        string toString() const;          // si lo usas en objetos const
};

```

Ejemplo de definiciones (si están en .cpp):

```

const string& CBoleta::getNroBoleta() const { return aNroBoleta; }
const string& CBoleta::getFecha() const    { return aFecha; }
int    CBoleta::getCantidad() const        { return aCantidad; }
float  CBoleta::getMonto() const           { return aMonto; }

CProducto* CBoleta::getProducto() const    { return aProducto; }
// o: const CProducto* CBoleta::getProducto() const { return aProducto; }
CVendedor* CBoleta::getVendedor() const    { return aVendedor; }
// o: const CVendedor* CBoleta::getVendedor() const { return aVendedor; }

string CBoleta::toString() const { /* ... */ }

```

Con eso, tu `GuardarBoletaTxt(const CBoleta* b, ...)` compilará sin quejarse al llamar `b->getMonto()`, `b->getFecha()`, etc.

Alternativa rápida (menos limpia)

Quita el `const` del parámetro:

```
void GuardarBoletaTxt(CBoleta* b, const string& nombreArchivo)
```

Compilará, pero **pierdes** la garantía de no modificar `b`. La opción correcta es la primera (marcar getters como `const`).

