Unidad 2

Herramientas de Modelado del Análisis y Diseño OO





Diagrama de Clases

Unidad 2.2

OBJETIVOS

- Reconocer diferentes componentes de un diagrama de clases y poder utilizarlos correctamente según el contexto.
- Conocer la metodología para transformar un diagrama de clases en código utilizando C#.



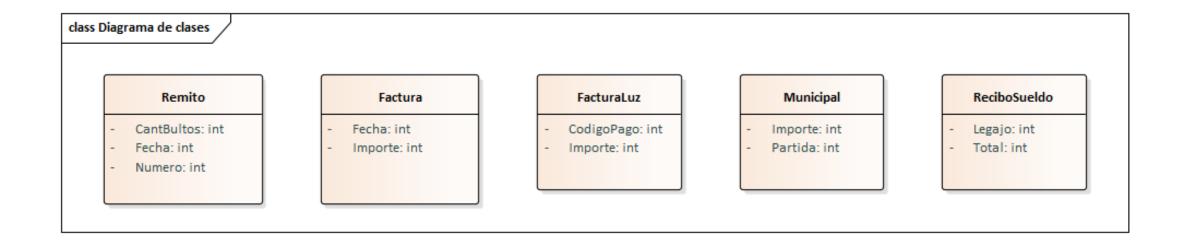
Ejemplo: Uso de Interfaces y Clases Abstractas



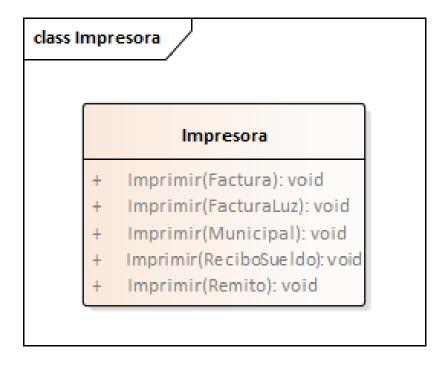


- Una empresa en crecimiento posee cada vez más cantidad de documentos diferentes en sus archivos: remitos, facturas, recibos de sueldo, impuestos (facturas de luz, gas, etc).
- Deciden hacer un repositorio para poder guardar todos esos documentos digitalmente y poder imprimirlos cuando se necesiten.

PRIMERA SOLUCIÓN

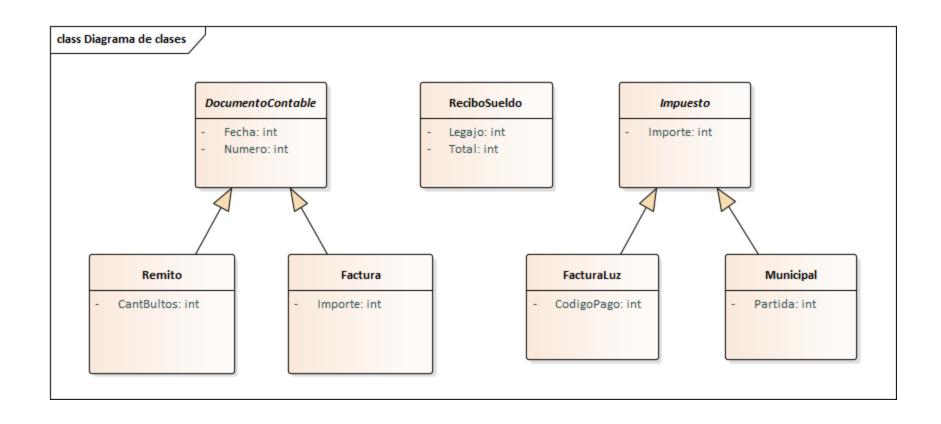


Luego de la primera solución, se dieron cuenta que necesitaban poder imprimir esos documentos para utilizarlos ante cualquier necesidad.

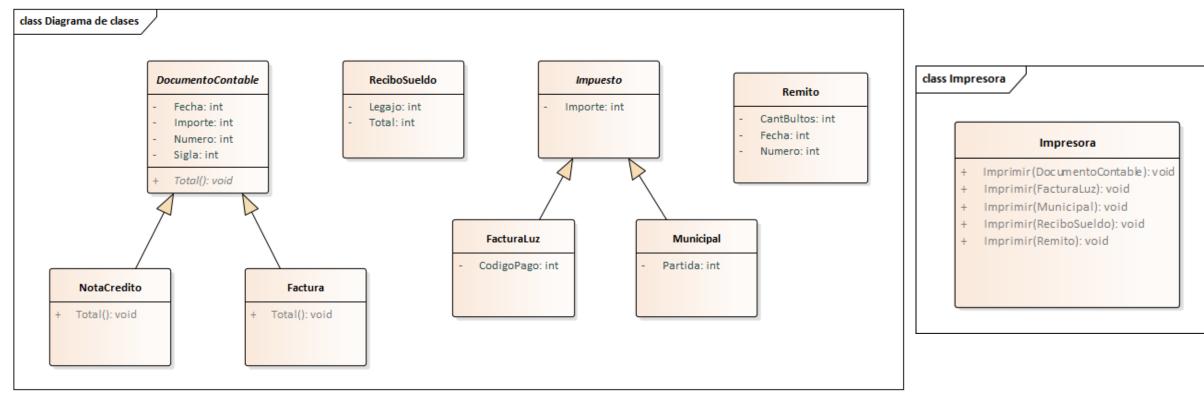


Posteriormente, se dieron cuenta que el código se podría mejorar un poco más, por lo que decidieron implementar algunas abstracciones para tal fin, logrando un poco más de flexibilidad al momento de poder incluir nuevos documentos para el repositorio

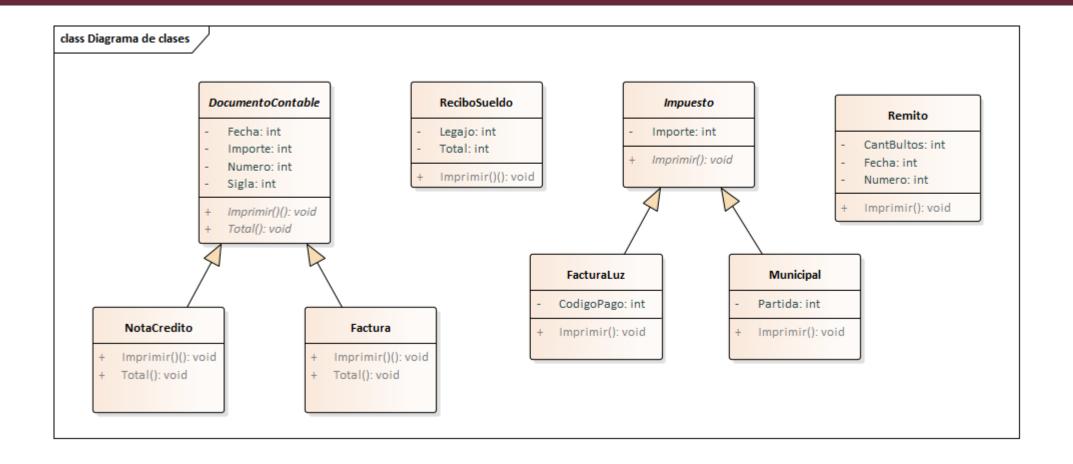
MEJORA DEL CÓDIGO



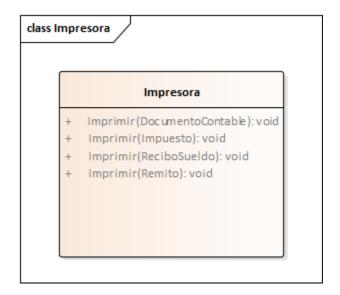
Vemos a continuación como queda el diagrama de clases luego de la mejora



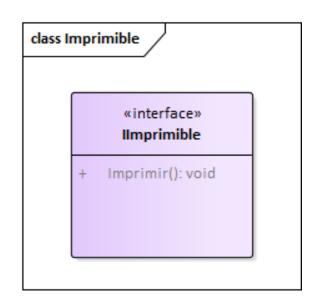
 Vemos que sigue siendo difícil poder agregar nuevos productos ya que la impresora también debe ajustarse. Vamos a intentar bajar ese acoplamiento, en principio, cambiando la responsabilidad de la operación Imprimir()

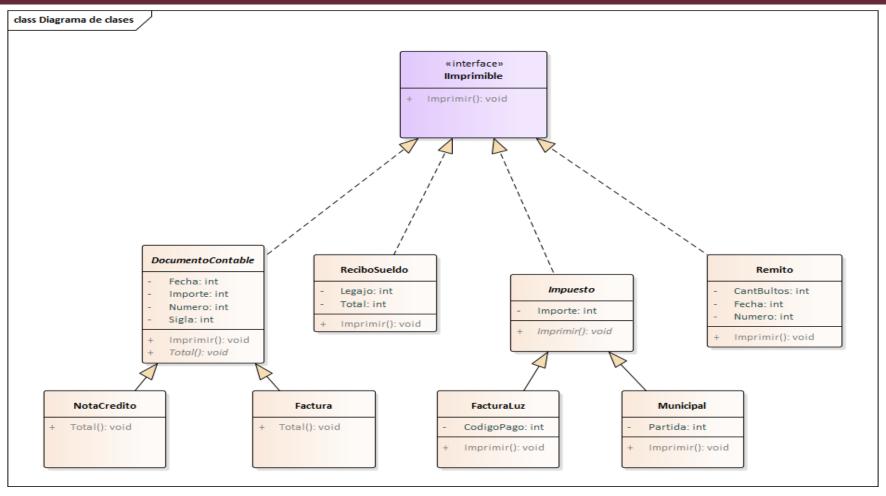


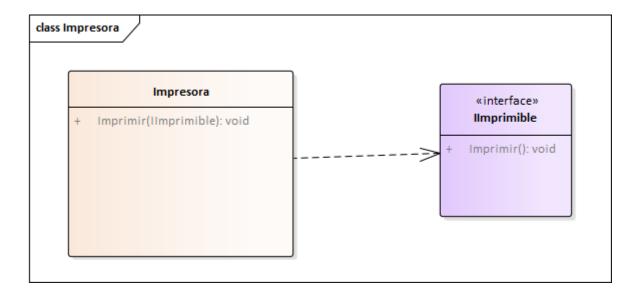
- Vemos como queda la impresora luego del refactor.
- Aún así, vemos que sigue siendo difícil continuar agregando nuevos documentos ya que todos tienen interfaces diferentes para imprimir (aunque tengan el mismo nombre de operación, no son interfaces compatibles).



Para ello creamos una nueva interface "Ilmprimible" la cuál será implementada en todos los documentos.





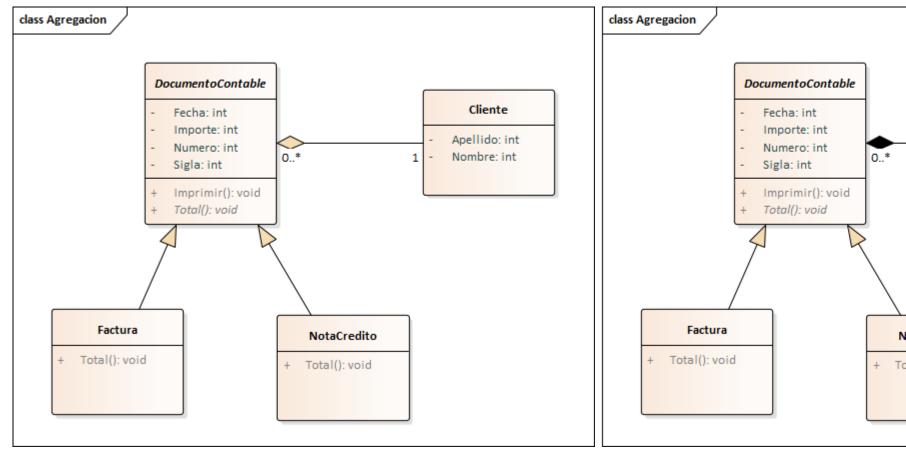


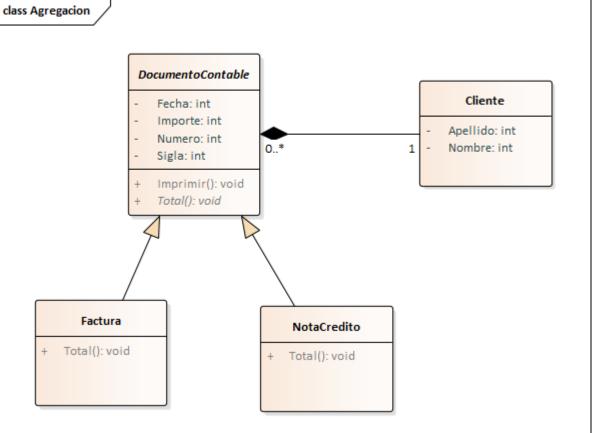
Ejemplo: Agregación y Composición













Fin de la clase

