Taller Integrador: Patrones de diseño

# Sistema de Requerimientos Académicos.

# Objetivos

* Evaluar que patrones de diseño se deben utilizar en el desarrollo de un sistema.
* Aplicar varios patrones de diseño dentro de un mismo sistema.

# Requerimientos del sistema:

La empresa de facturación electrónica Stupendo lo ha contratado para desarrollar el sistema facturación de uno de sus clientes.

Este sistema debe ser capaz de emitir tres tipos de comprobantes electrónicos:

* Factura
* Guia de remision
* Nota de crédito

Entre los requerimientos solicitados por cliente se encuentran:

* Todos estos comprobantes electrónicos además del membrete obligatorio con los detalles del emisor deben poder personalizarse con el logo de su negocio y un lema al final.
* Debido a que actualmente el SRI maneja dos esquemas de facturación electrónica (offline y online) el cliente solicitó que al momento de pedir la autorización del comprobante al SRI, pueda escoger cualquiera de los dos esquemas para autorizar.

En el esquema Onine, el número de autorización es un nuevo código asignado por el SRI, mientras que en el esquema Offline el número de autorización es igual a la clave de acceso.

Las facturas tienen la siguiente estructura:

* Detalles del emisor
* Nombre del cliente
* Número de autorización
* Clave de acceso
* Código
* Fecha
* Lista de productos
* Total

Las guías de remisión tienen:

* Detalles del emisor
* Número de autorización
* Clave de acceso
* Fecha
* Destino
* Placa del vehículo de transporte
* Lista de productos

Las notas de crédito tienen :

* Detalles del emisor
* Número de autorización
* Clave de acceso
* Nombre del cliente
* Código de comprobante a modificar
* Fecha
* Detalle de modificación
* Valor a pagar

# Desarrollar

1. Indique que patrones podrían servir dentro del desarrollo de este sistema. (explique)
   1. Creacionales.

Se escogió el patrón creacional Abstract Factory ya que tenemos que las clases facturas, guías de remisión y las notas de crédito comparten métodos y para ello es conveniente tener una interfaz que contenga la declaración de dichos métodos los cuales deberán ser implementados de acuerdo con las especificaciones de cada documento.

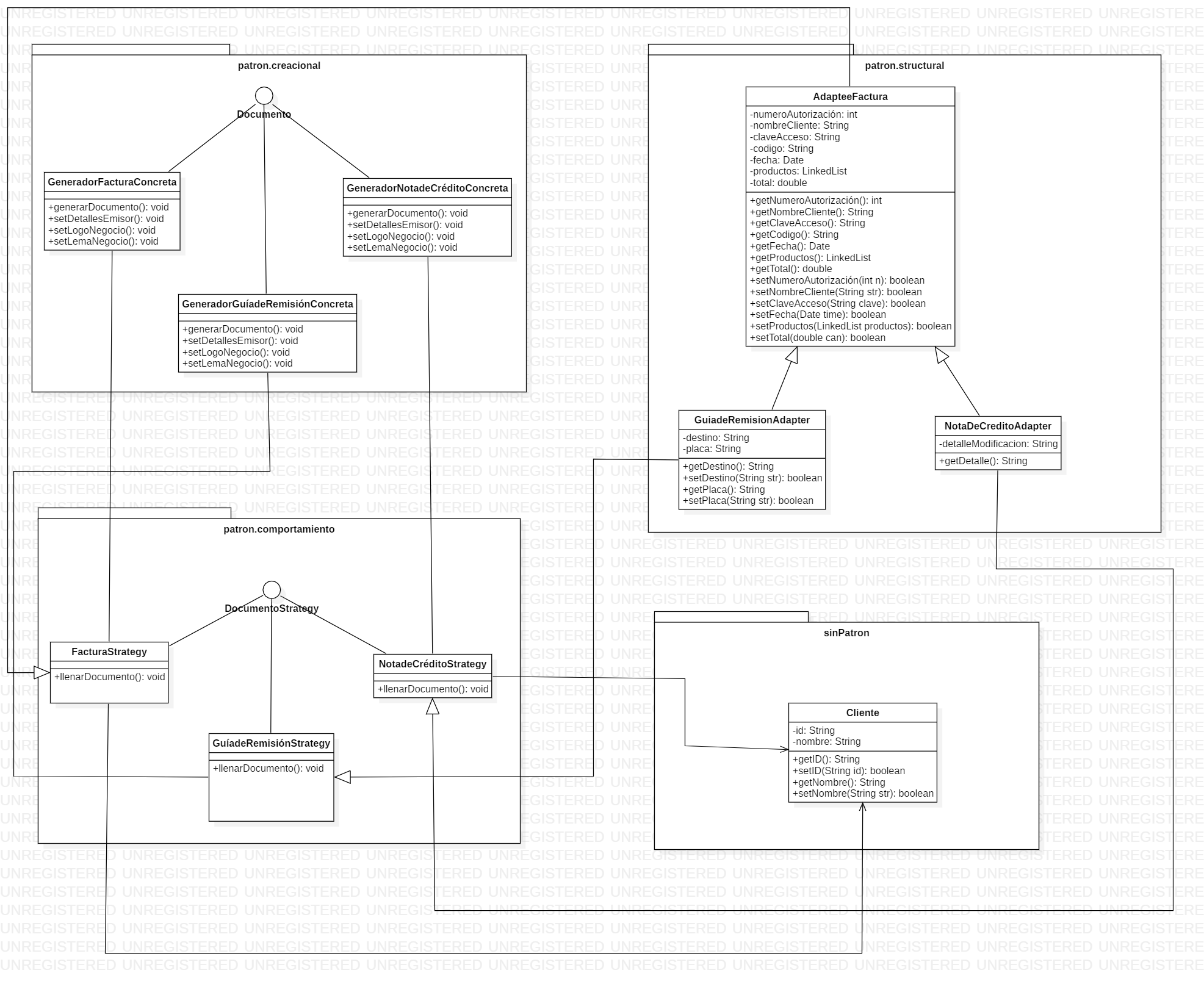
* 1. Estructurales.

El patrón escogido fue adapter para poder reutilizar la clase de AdapteeFactura con las clases GuíadeRemisiónAdapter y NotadeCréditoAdapter y así usar estas clases existentes para las que no sería conveniente heredar de cada una para adoptar su interfaz, así se puede aprovechar que la factura es el comprobante que comparte las características de los demás comprobantes y que mediante un adaptador se puede acceder a las diferentes operaciones que tendrán los otros comprobantes, así como sus distintos atributos, la ventaja es que si el cliente requiere agregar métodos adicionales para cada documento, éstos se pueden crear sin necesidad de modificar la clase AdapterFactura, solamente se agregan a cada una de las clases hijas, e incluso se pueden modificar los métodos en las clases hijas sin afectar al “Adaptee”.

* 1. De Comportamiento.

Se escogió el patrón de comportamiento Strategy para que una interfaz, denominada DocumentoStrategy, sea implementada por los diferentes tipos de documentos establecidos, y así dicho documentos puedan ser llenados con la información respectiva y requerida.

1. Diseñe un diagrama de clases del sistema



1. Implemente el sistema
   1. Cree un paquete por cada tipo de patrón
      * patron.creacional
      * patron.structural
      * patron.comportamiento
      * sinPatron

## En el programa principal muestre las funcionalidades del sistema.