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.

Abstract Factory: porque mejora la consistencia entre productos; el uso de la abstract factory permite forzar a utilizar un conjunto de objetos de una misma familia en este caso los comprobantes electrónicos.

* 1. Estructurales.

Decorator: porque permite añadir responsabilidades a objetos concretos de forma dinámica, en este caso nos ayuda con la personalización del comprobante electrónico añadiendo el logo de el negocio y un lema al final.

* 1. De Comportamiento.

Strategy: porque encapsula algoritmos en clases, permitiendo que éstos sean re-utilizados e intercambiables; es decir que le permite a un programa proporcionar múltiples variantes de un algoritmo o comportamiento, en este caso para implementar el comportamiento de pedir autorización cuando es offline y cuando es online.

1. Diseñe un diagrama de clases del sistema
2. Implemente el sistema
   1. Cree un paquete por cada tipo de patrón

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