UNIVERSIDAD NACIONAL DE INGENIERIA

Facultad de Ingeniería Industrial y de Sistemas

N° de Ensayo

- Ensayo N° 06

Título del ensayo realizado

- "Patrón de diseno Abstract Factory"

Nombre del profesor:

- Rony Jordan Hancco Carpio

Nombre del alumno:

- Flores Tambo, Diego Alonso.....20132716H

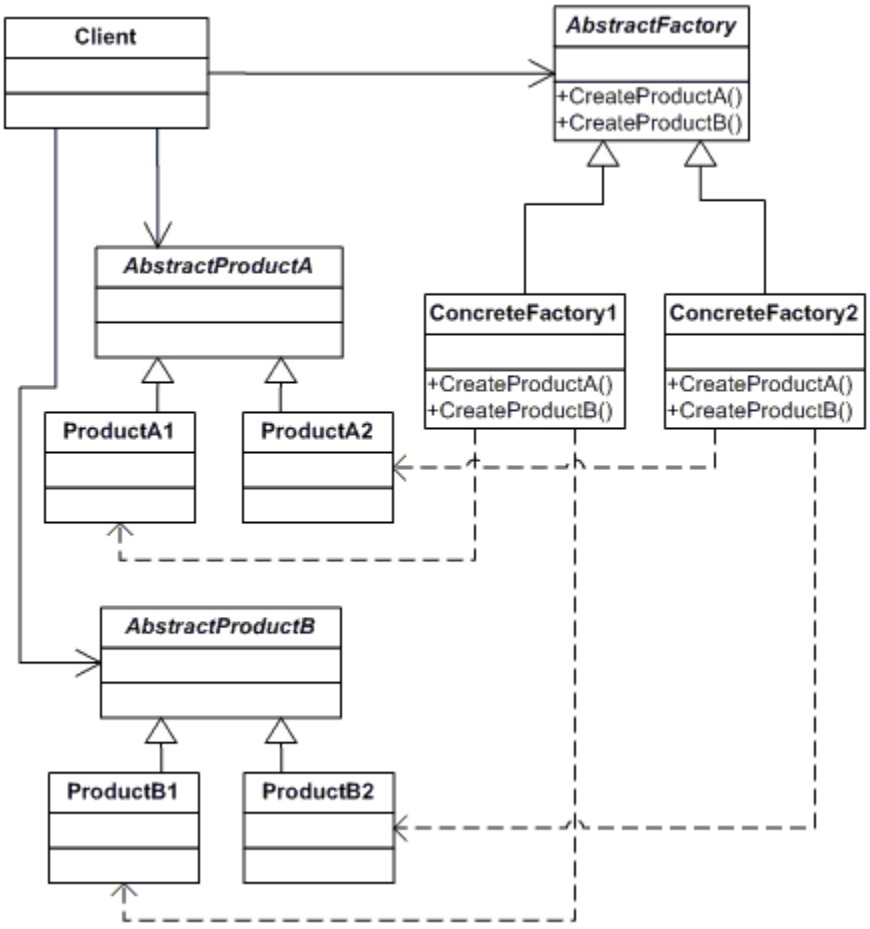
PATRON DE DISENO ABSTRACT FACTORY

La factoría o fábrica vendría a ser una clase que implementan métodos de creación, los cuales se encargan de crear instancias de objetos, ya sea de la misma o de otras clases.

Las factorías abstractas proporcionan una interfaz para crear familias de objetos que estén relacionadas o que dependan entre sí, sin necesidad de especificar sus clases.

Nos permite la creación, la instanciación de objetos de distintas clases, pero que comparten métodos en común.

Lo que busca el patrón es ofrecer la posibilidad de trabajar con varios objetos diferentes pero relacionados y facilitar la nueva adhesión de nuevos objetos relacionados en un futuro para cubrir nuevas necesidades.



Los objetos y clases participantes en el patrón son:

- Factoría abstracta: Interface que provee método para la creación de productos abstractos.
- Factoría concreta: Implementa las operaciones para crear un objeto producto concreto.
- Producto abstracto: Define la interfaz de los objetos que crea el método de fabricación.
- Producto concreto: Implementa la interfaz producto.

Instrucciones: Para el caso de la implementación de un banco y los servicios que ofrece: depósitos, transacciones y préstamos.

1. Implementamos nuestra interfaz que luego implementaran los servicios del banco.

$$public\ interface\ Servicio\ \{$$
$$public\ double\ montoServicio;$$
$$\}$$

2. Creamos la interfaz encargada de la creación de Servicio.

$$public\ interface\ ServicioBanco\ \{$$
$$public\ Servicio\ crearServicio;$$
$$\}$$

3. Ahora creamos los 3 servicios que ofrece el banco.

$$public\ class\ ServicioDeposito\ implements\ Servicio\ \{$$
$$public\ double\ montoServicio(\quad)\{return\ DEPOSITO;\};$$
$$\}$$
$$public\ class\ ServicioTransaccion\ implements\ Servicio\ \{$$
$$public\ double\ montoServicio(\quad)\{return\ TRANSACCION;\};$$
$$\}$$
$$public\ class\ ServicioPrestamo\ implements\ Servicio\ \{$$
$$public\ double\ montoServicio(\quad)\{return\ PRESTAMO;\};$$
$$\}$$

4. Creamos los Factory para cada servicio.

$$public\ class\ depositoFactory\ implements\ ServicioBanco\ \{$$

$$public\ void\ Servicio\ creaServicio(\quad))\{return\ ServicioDeposito;\};$$

$$\}$$

$$public\ class\ transaccionFactory\ implements\ ServicioBanco\ \{$$

$$public\ void\ Servicio\ creaServicio(\quad))\{return\ ServicioTransaccion;\};$$

$$\}$$

$$public\ class\ prestamoFactory\ implements\ ServicioBanco\ \{$$

$$public\ void\ Servicio\ creaServicio(\quad))\{return\ P\ ServicioPrestamo;\};$$

$$\}$$