



Ingeniería en computación

Curso

Diseño de software Gr2.

Profesora

Ericka Solano

Alumno

Oscar Chavarría Campos
2014088971

Semestre I

Patrones Creacionales

Cartago

2017

Caso 1: Juego de Tetis

Patrón a implementar: Factory

Justificación: Decidí implementar el Factory debido a que los productos o piezas del juego están definidas, cada pieza al crearla no cambia y por lo tanto se ve el juego como la fábrica y las piezas como sus productos definidos.

Responsabilidades:

AbstractCreator: Es el que va a tener el papel del juego en sí, ósea fabrica los productos o en este caso las piezas.

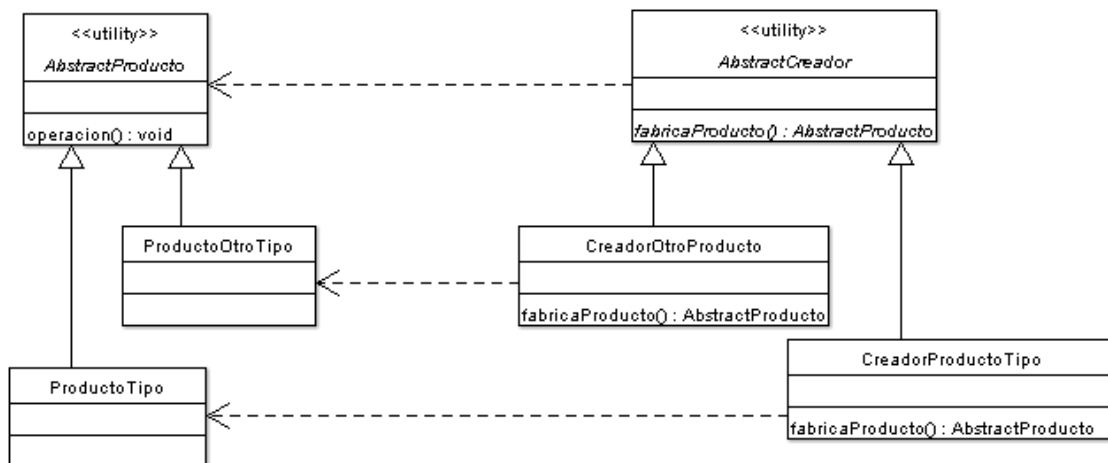
CreadorProducto: Es un clase hija que va tener la posibilidad de crear el producto, ya que cada producto tiene diferentes atributos, como por ejemplo hay diferentes piezas estas piezas ya están definidas por su cantidad de cuadros pero pueden tener diferentes posiciones y color.

Producto: Tendrá la definición del producto y sus atributos, donde cada producto debe de tener o puede tener diferentes atributos, como por ejemplo la figura 1 tiene 4 cuadritos fijos.

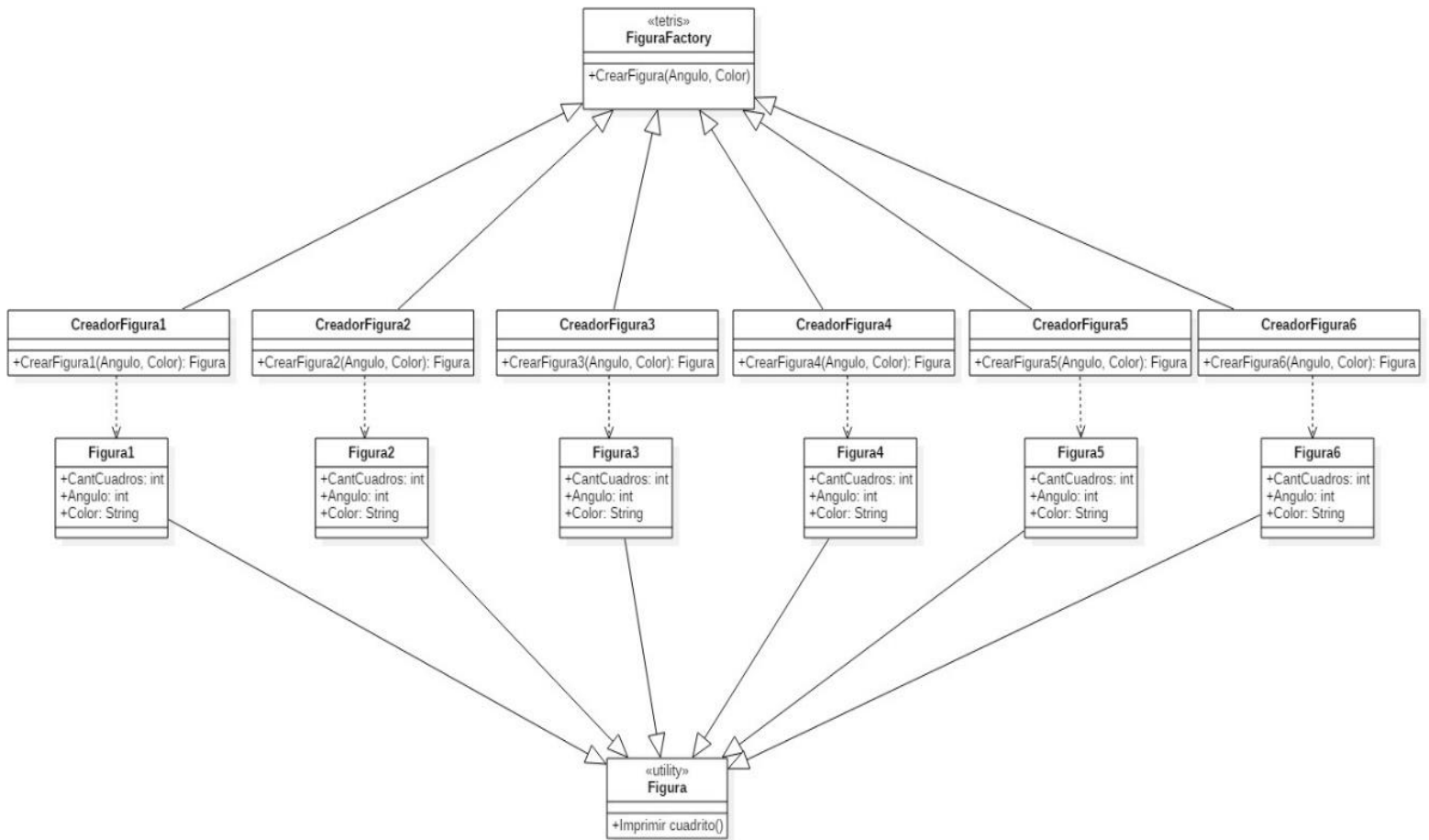
Interfaz: La interfaz contara de 1 método, "imprimirFigura()".

Modelos:

Patron base:



Modelo Propuesto:



Caso 3: Ayuda Sensitiva

Patrón a implementar: Singleton

Justificación: Solo hay una ayuda, ósea solo hay una posible instancia hacia la ayuda, la cual se hace al presionar “F1”.

Responsabilidades:

Singleton: Va a ser llamado a la hora de presionar “F1” dependiendo de la pantalla que lo llame mostrar diferentes tipos de ayuda, en una ventana.

Modelo:

Base:

Singleton
- <u>singleton : Singleton</u>
- Singleton()
+ <u>getInstance() : Singleton</u>

Propuesto:

Ayuda
+Intanciado: Ayuda
+CrearAyuda() +getInstaciado(): Ayuda +MostrarAyuda(int pantalla)