Patrones de Diseño

Percy Taquila Carazas, Katerin Merino Quispe, Abraham Lipa Calabilla, Edwart Balcon Coahila, Lisbeth Espinoza Caso

December 3, 2020

Abstract

Resumen

Los patrones de diseño dan un mecanismo codificado para describir problemas y su solución en forma tal que permiten que la comunidad de ingeniería de software diseñe el conocimiento para que sea reutilizado. Un patrón describe un problema, indica el contexto y permite que el usuario entienda el ambiente en el que sucede el problema, y enlista un sistema de fuerzas que indican cómo puede interpretarse el problema en su contexto, y el modo en el que se aplica la solución. El patron Abstract Factory suele implementarse con metodos de fabricacion que tambien generalmente son llamados desde el interior de Template Method.

I. Introduccion

Los

II. Desarrollo

```
package template;

public abstract class Comida {

    // template method
        public final void imprimir() {
            ingredientes();
            cocinar();
            comer();
        }

        public abstract void ingredientes();
        public abstract void cocinar();
        public abstract void comer();

}
```

La clase Hamburguesa extiende de Comida e implementa los tres métodos abstractos de Comida.

III. Conclusiones

La conclusión

IV. Recomendaciones

- Cuando se conoce el efecto colateral que conlleva el patrón de diseño y es viable la aparición de este efecto.
- Suministrar alternativas de diseño para poder tener un software flexible y reutilizable.

Referencias

[1] Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John(1995).Design Patterns: Elements of Reusable Object- Oriented Software. Reading,Massachusetts: Addison Wesley Longman, Inc.

[2] Nesteruk, D. (2019). Design Patterns in .NET: Reusable Approaches C# in and F# for Object-Oriented Software Design (1st ed.). Apress.

[3] Patrones de Diseño. Elementos de software orientado a objetos reutilizable. ERICH GAMMA. RICHARD HELM. RALPH JOHNSON. JOHN VLISSIDES.

[4] Patrones de diseño en Java: Los 23 modelos de diseño: descripción y solución ilustradas en UML 2 y Java Autor: Laurent Debrauwer