a

WENDY JULIETH GUIO QUINTERO

uninpahu  institucion educativo

abstract factory

**1. Introducción**

Este documento describe el proyecto **AbstractFactory**, proporcionando detalles sobre su propósito, características y estructura. Se incluyen aspectos técnicos sobre su implementación y configuración.

**2. Descripción del Proyecto**

El proyecto **AbstractFactory** implementa el patrón de diseño **Abstract Factory** en Java. Su objetivo es demostrar la utilidad de este patrón para la creación de objetos sin especificar sus clases concretas, facilitando la escalabilidad y modularidad del código.

**3. Características**

* Implementación del patrón **Abstract Factory** en Java.
* Modularidad y separación de responsabilidades mediante interfaces y clases abstractas.
* Uso de **Java** como lenguaje de programación principal.
* Posible integración con **JUnit** para pruebas unitarias.

**4. Requisitos Técnicos**

**4.1. Software Requerido**

* **JDK 8 o superior**
* **IDE compatible con Java (NetBeans, Eclipse, IntelliJ, etc.)**
* **Apache Ant o Maven** (si el proyecto utiliza scripts de construcción)

**4.2. Configuración y Ejecución**

Para ejecutar el proyecto, sigue estos pasos:

1. Abrir el proyecto en un IDE compatible con Java.
2. Asegurarse de que **JDK 8 o superior** está instalado.
3. Si se usa Ant, ejecutar:
4. ant clean build
5. Para ejecutar la aplicación:
6. java -jar AbstractFactory.jar

**5. Patrones de Diseño Utilizados**

El proyecto implementa el patrón **Abstract Factory**, que se emplea para crear familias de objetos relacionados sin especificar sus clases concretas. Se utiliza en situaciones donde se requiere flexibilidad en la creación de instancias.

**6. Conclusiones**

Este proyecto facilita el aprendizaje del patrón **Abstract Factory**, proporcionando una estructura organizada y escalable para la creación de objetos en Java. Su implementación modular permite una fácil adaptación a diferentes escenarios de desarrollo.