**Patrones Factory**

García Solís Jose Elvis

Junio 2017.

Instituto de Emprendedores USIL

Lima

Herramientas De Desarrollo De Software

Copyright © 2017 por Jose Gacrcia & “IEUSIL” Miguel Gutiérrez. Todos los derechos reservados.

**Dedicatoria**

A mis amigos que siempre están en las buenas y malas**Agradecimientos**

Gracias por su futura atención ante mi trabajo**Abstract**

Este es un documento de Word de ejemplo que puede ser usado como plantilla para dar formato a su tesis o disertación. El abstract o resumen debe contar con **250 palabras o menos**.

**Prefacio**

Esta página es opcional.**Tabla de Contenidos**

[Capítulo 1 Pratones de diseño Factory 1](#_Toc410628920)

[Que es un Patron de Diseño? 1](#_Toc410628921)

[Capítulo 2 Method Factory 2](#_Toc410628925)

[Definicion 2](#_Toc410628926)

[Ejemplos. 2](#_Toc410628927)

[Capítulo 3 Abtract Factory](#_Toc410628929)

[Ejemplos 5](#_Toc410628929)

[List of References 6](#_Toc410628930)

[Apéndice 7](#_Toc410628931)

[Vita 8](#_Toc410628932)

**Patrón de diseño Factory**

Cuando programamos en un lenguaje orientado a objetos siempre tenemos que crear objetos de clases que creamos e incluso de clases ya existentes.

El problema viene cuando tenemos objetos que tienen información similar, pero son divididos en clases distintas, es decir, que cada objeto se identifique por sus diferencias. Muchas confusiones se logran cuando se quiere instanciar de una cierta clase, pero eso se puede evitar utilizando un patrón de diseño.¿Qué es un patrón de diseño?

Básicamente los patrones de diseño son soluciones a problemas sencillos del desarrollo de software. Hay patrones de diseño que se enfocan en distintas tareas, pero el que veremos aquí es un patrón creacional.

Los patrones creacionales ayudan a generar una forma flexible de crear objetos sin tener que recurrir a usar el operador new y el uso del constructor original de la clase.

# Método Factory (Factory Method)

# El método Factory es bastante sencillo de explicar e implementar. Se trata de un simple método agregado dentro de una clase que, por medio de un parámetro o varios parámetros, se decide qué tipo de objeto crear para entonces retornarlo y poder utilizarlo. Esto hace que no tengamos que importar numerosas clases y mucho menos instanciar muchos objetos de esas mismas clases.

# Aquí se muestra un ejemplo elaborado en Javascript de cómo se emplea el método Factory. Hay que tomar en cuenta que en Javascript no existen las clases ni la herencia, por lo que todo se hace en base a funciones.

# 

# Ejemplos



# Abstract Factory

# Al contrario del método Factory, El Abstract Factory se enfoca a tener una fábrica única para esos objetos. Se crean fábricas en clases separadas que se enfocan en crear un cierto tipo de objeto y entonces evitar llamar los constructores originales.

# En el siguiente ejemplo se crean fábricas de objetos de tipo libro y revista. Ambos contienen el mismo tipo de campo, pero se tratan como objetos distintos.

# Ejemplos

# 