

## Curso Patrones de Diseño

**Descripción:** El curso cubre los patrones en tres de las áreas base: Creación, Estructural, y Comportamiento. Es un curso práctico conformado con proyectos de diseño y laboratorios de programación.

**Audiencia:** Desarrolladores de aplicación, programadores, diseñadores de sistemas y administradores de proyectos que necesitan mejorar el desarrollo de los sistemas usando patrones de diseño.

**Prerrequisitos:** Conocimiento profesional en Programación Orientada a Objetos, tecnologías orientada a objetos y diagramas UML. Conocimiento básico de Java.

# Contenido

## INTRODUCTION

- What's our World?
- OK – So Just What is a Design Pattern?
- Design Patterns are not Esoteric
- Why Use Patterns?
- The Adapter Pattern
- Reviewing Interfaces & Abstract Classes
- Interface Types
- Interface Definitions
- Abstract Methods
- Abstract Classes
- Using Abstract Classes
- Important Principal of OO Design

## THE ITERATOR PATTERN

- Patterns: Traversing a Collection
- A Simple ArrayList
- Using Our ArrayList
- Using Our Simple Collection
- Another Design for Collection Traversal
- Using Our New Collection
- Differences in Traversing Our Collection
- Why is This Important?
- Why is This a Design Pattern?
- We Will Expand on Our Design

## DESIGN PATTERNS – BACKGROUND

- Design Patterns Arise From Architecture
- Christopher Alexander
- The TimelessWay
- A Core Principle of His Books
- Patterns in A Pattern Language
- Sitting Circle (185)
- Different Chairs (251)
- Patterns Evolution in Software
- OOPSLA 88
- Patterns Evolution in Software
- Patterns Today

## DECORATOR PATTERN

- Motivation – Forces and Solution
- Structure
- Participants an Collaborations
- Structure
- Writer and FilterWriter Classes
- UpperCaseFilterWriter Class
- Consequences
- Implementation
- Known Uses and Related patterns

## COMPOSITE PATTERN

- Motivation – Forces
- Motivation – Solution
- Structure
- Participants
- Collaborations
- Consequences
- Implementation
- Known Uses and Related Patterns

## TEMPLATE METHOD PATTERN

- Motivation – Forces and Solution
- Structure
- Participants and Collaborations
- Consequences
- Implementation
- Known Uses and Related Patterns

## COMMAND PATTERN

- Motivation – Forces and Solution
- Structure
- Participants and Collaborations
- Consequences
- Implementation
- Undo and Redo
- Known Uses

## UML OVERVIEW

- Unified Modeling Language (UML)
- Using UML
- UML Diagrams
- Class Diagram
- Class Diagram Notation
- Association Relationships in Detail
- Class Diagram Notation
- Abstract Class Notation
- Interface Notation
- Another Class Diagram

## GANG OF FOUR DESIGN PATTERNS DESCRIPTION

- What Do We Know Now About Patterns
- GOF Pattern Description
- Iterator: Overview
- Iterator: Motivation
- Iterator: Applicability
- Iterator: Structure – Java
- Iterator: Structure – General
- Iterator: Participants
- Iterator: Collaborations and Consequences
- Iterator: Implementation
- Implementation: Who Controls the Iteration
- Implementation: Who Defines the Traversal
- Implementation: Robustness
- Iterator: Known Uses and Related Patterns
- So – What is a Design Pattern?

## THE GOF PATTERNS CATALOG

- Organizing the Catalog
- Creational, Structural, and Behavioral Purpose
- Class and Object Scope
- Design Pattern Space
- The GOF Catalog of Design Patterns

## FACTORY METHOD PATTERN

- Motivation – Forces and Solution
- Motivation
- Factory Method: Iterator Usage
- Factory Method: General Structure
- Participants
- Collaborations and Applicability
- Applicability
- Consequences
- Implementation
- Known Uses and Related Patterns

## CHAIN OF RESPONSIBILITY PATTERN

- Motivation – Forces
- Motivation – Solution
- Structure
- Participants and Collaborations
- Consequences/Applicability
- Implementation
- Known Uses and Related Patterns

## FAÇADE PATTERN

- Motivation – Forces and Solution
- Structure
- Participants and Collaborations
- Consequences/Applicability
- Implementation
- Known Uses

## PATTERNS FOR ENTERPRISE SYSTEMS

- Meeting the Challenge – Technologies
- Meeting the Challenge – Best Practices
- Some Patterns for Enterprise Systems
- Business Delegate
- Business Delegate: Solution
- Business Delegate: Structure
- Business Delegate: Consequences
- Value Object
- Value Object: Solution
- Value Object: Structure
- Value Object: Consequences
- Data Access Object (DAO)
- DAO: Solution
- DAO: Structure
- DAO: Consequences
- Lazy Load
- Lazy Load: Solution
- Lazy Load: Consequences

## WRAP-UP

- What Have We Done?
- So – What Do You Think About Patterns?
- Where Do We Go From Here?
- Do We Fit Into Alexander's Vision?
- Design Patterns Isn't All You Need
- Have Fun

## STRATEGY PATTERN

- Motivation – Forces and Solution
- Structure
- Alternative to Strategy
- How Do We Choose Among Alternative?
- Participants
- Collaborations and Applicability
- Consequences
- Implementation
- Known Uses and Related Patterns
- Difference From Factory Method

### Incluye:

- ❖ Un manual del curso por participante.
- ❖ Estacionamiento si el curso es en nuestras instalaciones.
- ❖ Descanso con servicio de café, agua, té, refrescos y galletas.

### Formas y condiciones de pago:

El pago debe hacerse de alguna de las siguientes maneras, pagando el total antes de la fecha de inicio del curso:

- ❖ Depósito Banamex cuenta 4923239 Suc. 575 a nombre de Desarrollo y Capacitación en Internet, S. A. de C. V. o transferencia bancaria CLABE 002180057549232394
- ❖ Cheque a nombre de Desarrollo y Capacitación en Internet, S. A. de C. V.