



Pontificia Universidad  
**JAVERIANA**  
Bogotá

ESPECIALIZACIÓN EN  
ARQUITECTURA EMPRESARIAL DE SOFTWARE

# Patrones de Software

Juan Carlos Cerón Barreto

[ceron.juan@javeriana.edu.co](mailto:ceron.juan@javeriana.edu.co)

Especialista en Arquitectura Empresarial de Software

MCTS – MCPD - SOA Profesional - Scrum Master



Pontificia Universidad  
**JAVERIANA**  
Bogotá

ESPECIALIZACIÓN EN  
ARQUITECTURA EMPRESARIAL DE SOFTWARE

# INFORMACIÓN GENERAL



# Presentación

- Nombre
- Empresa
- Cargo / Funciones
- Responsabilidades
- Experiencia
- Expectativas de la especialización
- Expectativas de la materia





# Objetivos Generales

- Estructurar soluciones a partir de arquitecturas bien definidas.
- Capacitar en técnicas y estándares para la creación de arquitecturas de software.
- Desarrollar habilidades de abstracción y descomposición para estructurar problemas complejos.



# Estructura de Trabajo

- Presentación de patrones
- Lecturas preseleccionadas
- Clases
- Desarrollo de talleres y trabajos prácticos
- Socializaciones - casos de estudio
- Quices
- 2 Parciales
- Proyecto final.



# Evaluación del Curso

- Socializaciones 45%
  - 3 Socializaciones 15% c/u
- Nota Individual 15%
  - Quices 10%
  - Taller Individual 5%
- Proyecto y Parciales 40%
  - Parciales 10% c/u
  - Proyecto Final 20%





# Reglas de Juego

- Inicio de clase 6:00 pm
- Uso celular fuera del salón
- Lecturas antes de clase
- Break entre 15 – 20 min
- Comunicación asertiva
- Criticas constructivas
- Participación activa





Pontificia Universidad  
**JAVERIANA**  
Bogotá

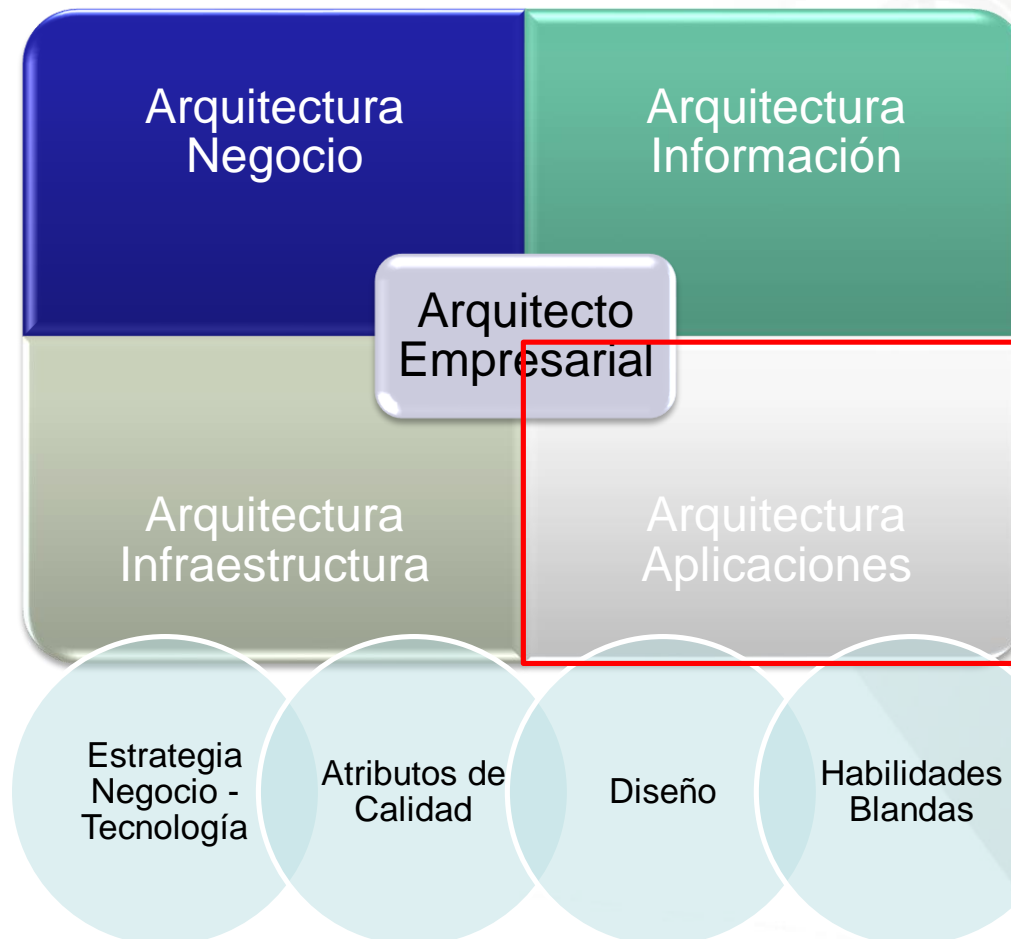
ESPECIALIZACIÓN EN  
ARQUITECTURA EMPRESARIAL DE SOFTWARE

# INTRODUCCIÓN





# Habilidades Arquitecto Empresarial





# Que es el Diseño?

- El diseño son las justificaciones, consideraciones, restricciones y trade offs que respaldan las decisiones
- No son solo diagramas de UML
- Apoya necesidades del negocio
  - Requerimientos funcionales
  - Requerimientos no funcionales



# Arquitectura de software

- Definición del SEI:
  - “*La Arquitectura De Software es la estructura o estructuras del sistema que comprende **elementos de software**, las propiedades visibles de forma externa de estos elementos y las relaciones entre los elementos*”



# Patrones

1. Historia
2. Definiciones
3. Patrones
4. Elementos, Descripción
5. Catálogos de patrones



# Historia

- El concepto de patrones nace desde la arquitectura civil con el objetivo de reutilizar los diseños que se habían aplicado en construcciones y se calificaron como completos



# Historia

- **Christopher Alexander** estudió arquitectura en la Universidad de Harvard.
- Considera que los **usuarios** saben más sobre los edificios que necesitan que lo que podría llegar a saber un arquitecto
- En **1979** produjo y validó, junto con otros, un **lenguaje de patrones** diseñado para empoderar a cualquiera en el diseño y construcción a cualquier escala



# Historia

- En **1987**, **Ward Cunningham y Kent Beck** adoptan las ideas de Alexander para el campo de la **ingeniería de software**
- Proponen cinco patrones relacionados con el **diseño de interfaces** de usuario
  - Window Per Task
  - Few Panes Per Window
  - Standard Panes
  - Short Menus
  - Nouns and Verbs



# Historia

- En **1992** se publica el libro **Advanced C++ Programming Styles and Idioms** de **Jim Coplien** donde se recopilan patrones específicos para el lenguaje C++ y se denominan idiomas





# Historia

- Desde **1990 a 1994**, Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides (**GoF**) realizaron un primer **catálogo de patrones de diseño**.
- Fue calificado como el mejor libro de orientación al objeto de 1995 y de todos los tiempos por la revista JOOP (Journal of Object Oriented Programming)



# Historia

- En **1996**, se publica el libro **Pattern Oriented Software Architecture**: Frank Buschmann et al.
- En **1997** Brad Appleton publica "Patterns and Software: Essential Concepts and Terminology".



# Definiciones

- *Christopher Alexander*
  - Cada patrón describe un **problema** que ocurre una y otra vez en nuestro entorno y luego describe el núcleo de la solución a ese problema, de tal forma que pueda usarse la **solución muchas veces** más, sin tener que repetir el proceso más de una vez”.





# Definiciones

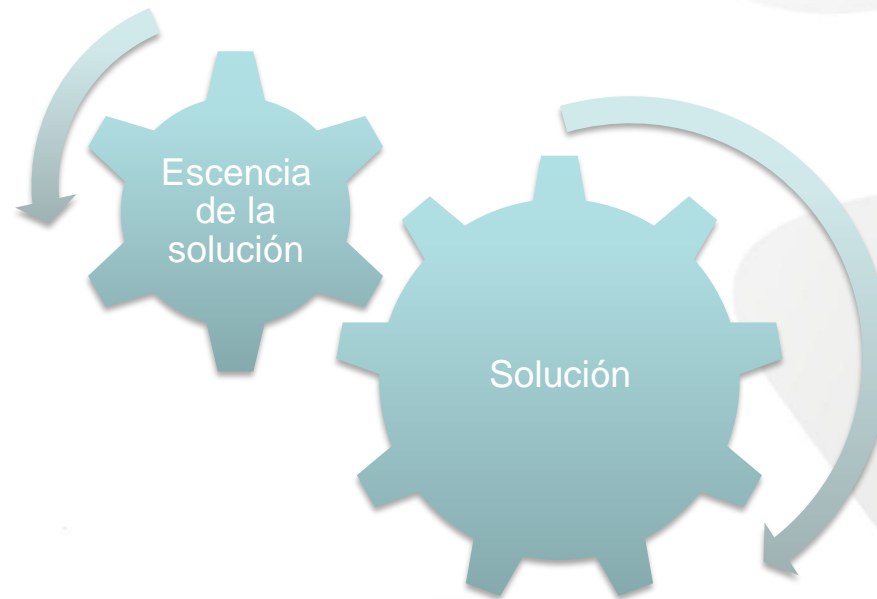
- *POSA*
  - “Un **patrón de arquitectura de software** describe un **problema particular recurrente de diseño** que surge en contextos específicos de diseño, y presenta un esquema genérico de solución probado. El **esquema de solución** se especifica describiendo sus componentes constituyentes, sus responsabilidades y relaciones, y las formas en que colaboran.”





# Definiciones

- Los expertos, de muchas disciplinas, suelen reutilizar la esencia de una solución en lugar de inventar una nueva..





# Definiciones

- **Lugar con ventanas**







# Definiciones

- Cada patrón es una reglas de tres partes relacionadas entre si



- Como un elemento de lenguaje, un patrón es una instrucción, que muestra como una configuración especial **puede ser usada, una y otra vez**, para resolver el sistema de fuerzas dadas, cada vez que el **contexto** las haga relevantes.





# Patrones

- Los patrones pueden ser usados como elementos definidores de un vocabulario común para todos los agentes del diseño, como ahorradores de tiempo y para el registro de mantenimiento de soluciones buenas y probadas para problemas recurrentes







# Uso de Patrones

Se pueden emplear patrones para describir la arquitectura de sistemas de software a diferentes niveles

## Patrones de Arquitectura

Afectan a la estructura global del sistema

## Patrones de Diseño

Definen micro-arquitecturas de subsistemas de componentes

## Idiomas

Detalles de la estructura y comportamiento de un componente



# Patrones de Arquitectura

- La arquitectura de software es un plano técnico que explica como el sistema será estructurado
- La arquitectura del sistema describe
  - Como el sistema será descompuesto en subsistemas o módulos
  - Las responsabilidades de cada modulo
  - La interacción entre los módulos
  - Plataformas y tecnologías
  - Atributos de Calidad
- Cada módulo a su vez puede implementar determinados patrones o modelos

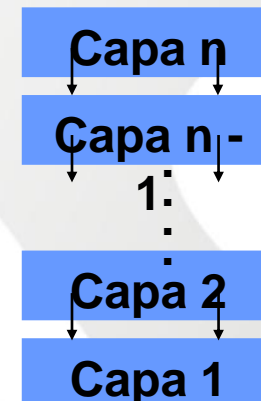
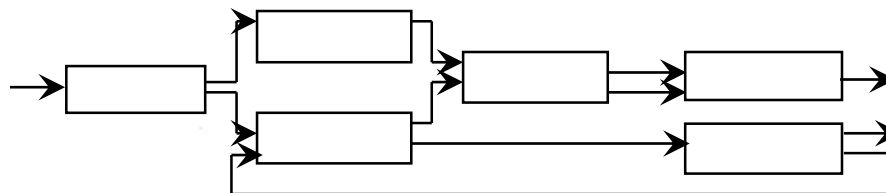


# Patrones de Arquitectura

Patrones de Arquitectura  
Afectan a la estructura global del sistema

Un **patrón de arquitectura o estilo arquitectónico**

- expresa un esquema de organización estructural para sistemas de software.
- Provee un conjunto de tipos de elementos predefinidos,
- Especifica sus responsabilidades
- Incluye reglas y guías para organizar las relaciones entre ellos



# Patrones de Diseño

Patrones de Arquitectura  
Afectan a la estructura global del sistema

Patrones de Diseño  
Definen micro-arquitecturas de subsistemas de componentes

- Refinan un subsistema o componente de un sistema de software
- Son Patrones de escala media
- No dependen del lenguaje de implementación
- Permiten resolver problemas complejos y direccionan la cooperación efectiva entre componentes





# Idiomas

Patrones de Arquitectura

Afectan a la estructura global del sistema

Patrones de Diseño

Definen micro-arquitecturas de subsistemas de componentes

Idiomas

Detalles de la estructura y comportamiento de un componente

- Es un patrón de bajo nivel, específico para un lenguaje de programación.
- Describe como implementar aspectos particulares de elementos o de las relaciones entre ellos usando las características de un lenguaje particular.



# Ventajas de los Patrones

- ✓ Facilitan la comunicación interna.
- ✓ Ahorran tiempo y experimentos inútiles para el manejo de la complejidad.
- ✓ Mejoran la calidad del diseño y la implementación.
- ✓ Ayudan a construir arquitecturas de software con propiedades específicas.
- ✓ Son como “normas de productividad”
- ✓ Permite la realización de principios de diseño
- ✓ Relacionan decisiones de diseño a soluciones







# Reflexión



Son un punto de inicio, NO un fin.

No resuelven su problema, pero  
sí resuelven aspectos de este.

Mal aplicados puede llevarlo por el  
camino equivocado.



# Elementos

## Contexto

- Situación en la que se genera el problema

## Problema

- Inconvenientes de diseño

## Fuerzas

- Requerimientos por cumplir
- Restricciones

## Solución

- Elementos, relaciones, responsabilidades y colaboraciones

## Consecuencias

- Resultados de aplicar el patrón





# Ejemplo

- Lugar con ventanas





# Descripción

1. Nombre del patrón
2. Objetivo
3. Contexto
4. Aplicabilidad
5. Solución
6. Consecuencias
7. Implementación
8. Patrones relacionados





# Catálogos





# Catálogos POSA



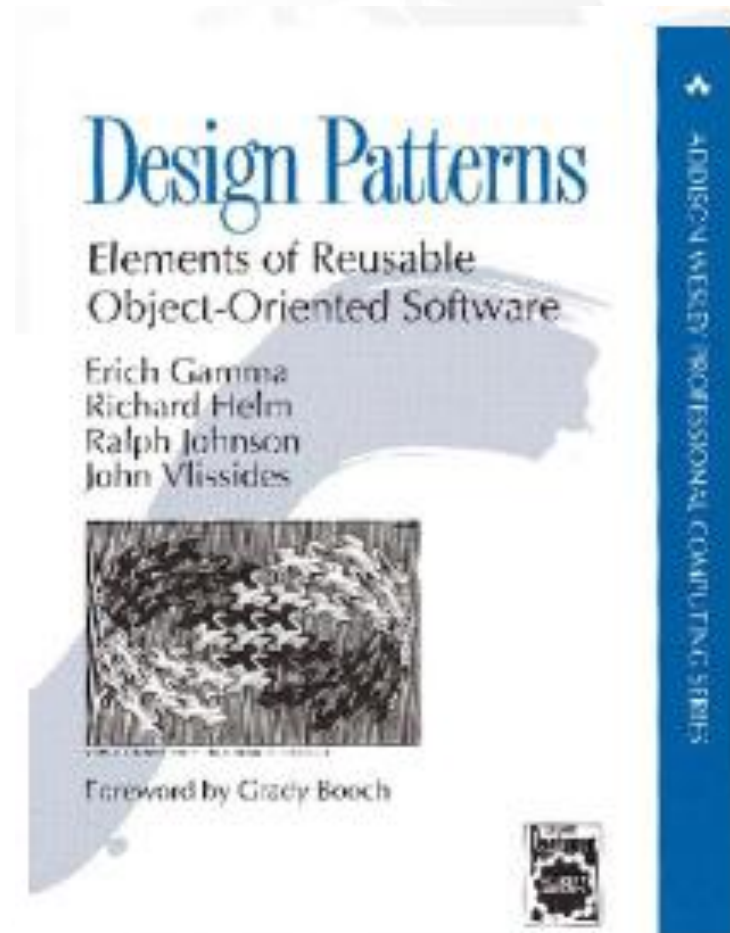
Pattern-Oriented Software  
Architecture: A System of  
Patterns,  
F. Buschmann, R.  
Meunier, H. Rohnert, P.  
Sommerlad, M. Stal.  
John Wiley & Sons, 1996





# Catalogo GoF

- Erich Gamma, Richard Helm, Ralph Johnson, & John Vlissides (1994).  
Design patterns:  
elements of reusable  
object-oriented software.  
Pearson Education.





# Trabajo Individual

- Repaso Lenguaje Patrones
  - POSA Capitulo 1, sección 1.1, 1.2, 1.3



# Bibliografía

- Pattern-Oriented Software Architecture: A System of Patterns, F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal. John Wiley & Sons, 1996
- Enterprise Integration Patterns - Designing, Building And Deploying Messaging Solutions: Gregor Hohpe, Bobby Woolf, The Addison-Wesley Signature Series, 2005
- Patterns of Enterprise Application Architecture: Martin Fowler, Addison-Wesley Professional, 1 edition, November 15, 2002.
- Erl, Thomas, "SOA Design Patterns". Prentice Hall. 2008
- <http://www.sei.cmu.edu/architecture/start/glossary/moderndefs.cfm>



# Preguntas

