

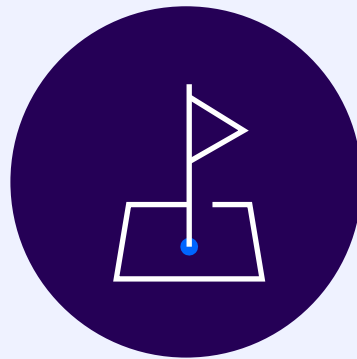
# JAVA DESIGN: PRINCIPLES & PATTERNS

Java Core, Principles, Patterns, Design,  
Modeling, OOP y Advanced.

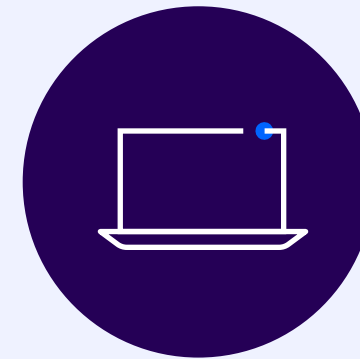




Inicio:  
**06 DE MAYO**



Finalización:  
**03 DE JUNIO**



**26 HORAS**  
académicas



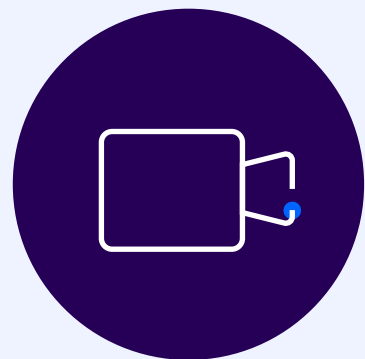
Sábado  
**DE 09:30H - 14:30H.**



Soporte  
**TÉCNICO**



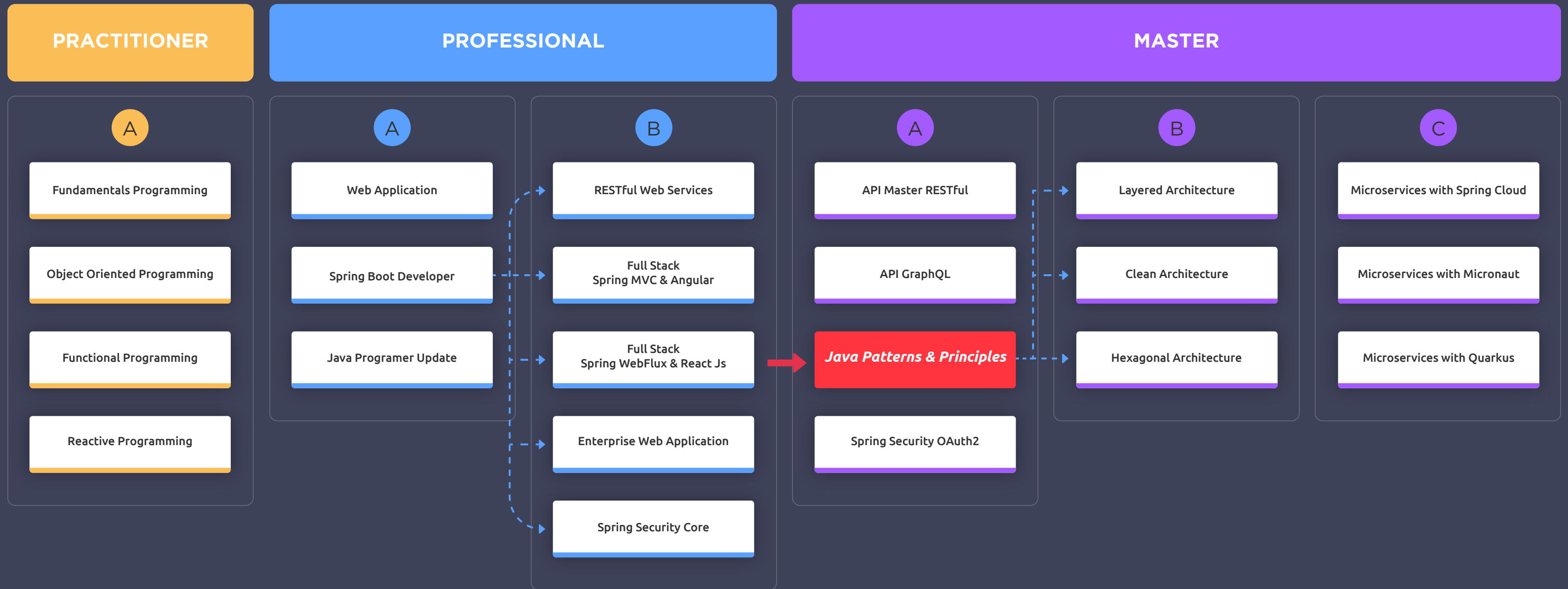
Plataforma  
**DIGITAL**



Aplicativo  
**ZOOM**



**CERTIFICACIÓN  
DIGITAL**  
Previa aprobación  
de examen



Este curso es pre-requisito para complementar el nivel de especialización **Master B.**

## Instructor

# ARISTEDES NOVOA

Ingeniero de Sistemas Colegiado de la Universidad Nacional del Callao (UNAC), con maestría en Computación y Sistemas de la Universidad de San Martín de Porres (USMP) y Doctorado en Ingeniería Industrial de la Universidad Nacional Mayor de San Marcos (UNMSM). Especialista en optimización, sistematización y automatización de procesos e implementación de soluciones empresariales utilizando Java; con más de 15 años de experiencia.

## Área de Especialización: JAVA

### Principales Tecnologías

- Java Core
- Java SE
- Java EE
- Microservicios
- Spring Framework
- Docker y Kubernetes



### Certificaciones Internacionales

- Java SE (Oracle)
- Agile (SDC, SMC, SPOC, SSMC, SSPOC, STC)
- DevOps y CI/DI (DEPC, Jenkins)
- Big Data y Data Science (Hadoop, Spark, IA, ML, DL)



### Líneas de Investigación


- IA, ML, DL
- Blockchain
- Sistemas Reactivos



anovoa@galaxy.edu.pe



Linkedin: <https://bit.ly/3cMcfv3>



## RESUMEN

Análisis, diseño e implementación de los principales principios y patrones de diseño utilizando Java. Además de conocer las principales recomendaciones, buenas prácticas y escenarios para su implementación en proyectos de desarrollo de software.



## OBJETIVO

Contar con los conocimientos y entrenamiento práctico en el uso de principios y patrones de diseño de software utilizando Java Core y Spring Framework; estar preparado para llevar cursos de nivel avanzado en Arquitectura de Aplicaciones implementando patrones avanzados y tecnologías complementarias.



## METODOLOGÍA

- Exposición de aspectos teóricos de los temas.
- Desarrollo de casos prácticos progresivos ( instructor y alumno).
- Experiencias del instructor y alumnos del curso.
- Desarrollo de proyecto de curso.



## REQUISITOS ACADÉMICOS

- Programación orientada a objetos y funcional de preferencia.
- Modelamiento y diseño de clases utilizando UML.
- Fundamentos de arquitectura de aplicaciones en capas o similares.
- Conocimientos de Java EE/Jakarta EE y Spring Framework.

## PRINCIPLES & PATTERNS

- Design Principles
- Creational Design Patterns
- Behavioral Patterns
- Behavioral Patterns
- Structural Design Patterns

UNIFIED  
MODELING  
LANGUAGE



## PROJECT APPLICATION

- Implementación de Principles & Patterns en un caso empresarial.
- Retroalimentación y mejoras.



# — • SESIÓN 01

## INTRODUCCIÓN A PRINCIPIOS DE DISEÑO

- Introducción a principios de diseño.
- Lineamientos de principios de diseño:
  - Buenas prácticas: limpio y modular, alta cohesión y bajo acoplamiento
  - Malas prácticas: rígido, fragmentado y no reutilizable
- Beneficios de los principios de diseño orientado a objetos(extensibilidad, testeabilidad, reusabilidad y flexibilidad)
- Principales principios( S.O.L.I.D., DRY, DI, encapsulación, composición vs herencia, delegación)

# — • SESIÓN 02

## APLICACIÓN DE PRINCIPIOS DE DISEÑO S.O.L.I.D. Y RELACIONADOS

- Introducción a principios de diseño S.O.L.I.D.
- ¿Qué es?, ¿Cómo implementarlo? y ¿Cuándo usarlo?
- Single Responsibility Principle (SRP).
- Open Closed Design Principle.
- Liskov Substitution Principle (LSP).
- Interface Segregation Principle.
- Dependency Inversion Principle: Desarrollo de casos prácticos de aplicaciones de cada principio.

## — • SESIÓN 03

### INTRODUCCIÓN A PATRONES DE DISEÑO CREACIONAL

- Introducción a patrones relacionados con la instanciación de objetos (Factory patterns).
- ¿Qué es?, ¿Cómo implementarlo? y ¿Cuándo usarlo?
- Tres tipos diferentes de patrones de creación:
  - Simple factory
  - Factory method pattern
  - Abstract factory pattern

## — • SESIÓN 04

### APLICACIÓN DE PATRONES DE DISEÑO CREACIONAL

- Singleton pattern
- Builder pattern
- Prototype pattern
- Object pool pattern
- Desarrollo de casos prácticos de aplicaciones de cada principio



## — • SESIÓN 05

### INTRODUCCIÓN A PATRONES DE COMPORTAMIENTO

- Introducción a patrones relacionados con la comunicación entre objetos.
- ¿Qué es?, ¿Cómo implementarlo? y ¿Cuándo usarlo?
- Strategy pattern.
- Template method pattern.
- Visitor pattern.
- Command pattern.
- Chain of responsibility pattern.

## — • SESIÓN 06

### APLICACIÓN DE PATRONES DE COMPORTAMIENTO

- Observer pattern.
- Mediator pattern.
- Null object pattern.
- Iterator pattern.
- Interpreter pattern.
- Memento pattern.
- State pattern.
- Desarrollo de casos prácticos de aplicaciones de cada principio.

## — • SESIÓN 07

### INTRODUCCIÓN A PATRONES DE DISEÑO ESTRUCTURAL

- Patrones relacionados con la composición de clases y objetos.
- ¿Qué es?, ¿Cómo implementarlo? y ¿Cuándo usarlo?
- Adapter pattern.
- Decorator pattern.
- Bridge pattern.

## — • SESIÓN 08

### APLICACIÓN DE PATRONES DE DISEÑO ESTRUCTURAL

- Flyweight pattern.
- Proxy pattern.
- Composite pattern.
- Facade pattern.
- Desarrollo de casos prácticos de aplicaciones de cada principio.

# —• CRONOGRAMA

SESIÓN	01 y 02	03 y 04	05 y 06	07 y 08
ACTIVIDAD	INTRODUCCIÓN A PRINCIPIOS DE DISEÑO	INTRODUCCIÓN A PATRONES DE DISEÑO CREACIONAL	INTRODUCCIÓN A PATRONES DE COMPORTAMIENTO	INTRODUCCIÓN A PATRONES DE DISEÑO ESTRUCTURAL
	APLICACIÓN DE PRINCIPIOS DE DISEÑO S.O.L.I.D. Y RELACIONADOS	APLICACIÓN DE PATRONES DE DISEÑO CREACIONAL	APLICACIÓN DE PATRONES DE COMPORTAMIENTO	APLICACIÓN DE PATRONES DE DISEÑO ESTRUCTURAL
FECHA	06 MAY	20 MAY	27 MAY	03 JUN

Este cronograma puede estar sujeto a cambios por parte de Galaxy Training.