**SOBRE O CURSO**

This comprehensive course on software design patterns and architectures is designed for software engineers and developers seeking to enhance their understanding of advanced design concepts. Participants will explore various design patterns and architectural principles to create robust, maintainable, and scalable software systems. Through a combination of lectures, hands-on exercises, and real-world examples, students will gain the skills needed to make informed architectural decisions and solve complex design challenges.

**PRÉ REQUISITOS**

* Proficiency in at least one programming language (e.g., Java, C++, Python, etc.)
* Basic knowledge of object-oriented programming
* Familiarity with software development concepts

**CONTEÚDOS PROGRAMÁTICOS**

**Introduction to Software Architecture**

* Understanding software architecture
* The role of architecture in software development
* Architectural styles (e.g., monolithic, microservices, serverless)
* Architectural patterns (e.g., MVC, MVP, MVVM)
* Architectural decision-making

**Design Principles and SOLID**

* Single Responsibility Principle (SRP)
* Open/Closed Principle (OCP)
* Liskov Substitution Principle (LSP)
* Interface Segregation Principle (ISP)
* Dependency Inversion Principle (DIP)
* Applying SOLID principles to real-world examples

**Creational Design Patterns**

* Singleton Pattern
* Factory Method Pattern
* Abstract Factory Pattern
* Builder Pattern
* Prototype Pattern
* Practical applications and trade-offs

**Structural Design Patterns**

* Adapter Pattern
* Bridge Pattern
* Composite Pattern
* Decorator Pattern
* Facade Pattern
* Flyweight Pattern
* Proxy Pattern
* Real-world scenarios for structural patterns

**Behavioral Design Patterns**

* Observer Pattern
* Strategy Pattern
* Command Pattern
* Chain of Responsibility Pattern
* State Pattern
* Visitor Pattern
* Interpreter Pattern
* Use cases and examples of behavioral patterns

**Architectural Patterns and Best Practices**

* Model-View-Controller (MVC) and its variants
* Layered Architecture
* Domain-Driven Design (DDD)
* Microservices Architecture
* Event-Driven Architecture
* Design patterns in distributed systems
* Scalability, performance, and reliability considerations

**Final Project:**

* Students will work on a real-world software design and architecture project, applying the principles and patterns covered in the course. They will present their project at the end of the course.

**FORMADOR:**

**DURAÇÃO**: 30h

**DATA**

Dias

Das 19h às 21h00

**LOCAL**

Formação Remota – Plataforma Microsoft Teams