

	PROGRAMA ACADÉMICO DE LA ASIGNATURA	CÓDIGO: GFR-FOR-005
		VERSIÓN: 3
		FECHA ELABORACIÓN DEL DOCUMENTO: 15/ENE/2021

1. IDENTIFICACIÓN

Nombre asignatura:	Ingeniería de Software I
Código	103022
Departamento:	Ciencias Computacionales
Programa (s) en los que se ofrece:	Ingeniería de Sistemas
Número de créditos:	3
Horas de trabajo presencial:	3
Horas de trabajo dirigido:	3
Horas de trabajo independiente:	3
Prerrequisitos:	Ingeniería de Software I
Periodo académico:	2022-1
Docente:	Yaneth Mejía Rendón
E-mail docente:	yaneth.mejia@autonoma.edu.co

2. JUSTIFICACIÓN

Los ingenieros de sistemas están constantemente relacionados con actividades que tienen que ver con el desarrollo de software. Estas actividades deben llevarse a cabo según algún proceso establecido, aplicando técnicas y métodos propios de la profesión, y cumpliendo con las expectativas de los usuarios.

Con base en el paradigma del desarrollo de software basado en modelos, este curso presenta técnicas y métodos que permiten afrontar con el profesionalismo requerido las actividades de diseño y pruebas en los proyectos de desarrollo de software, junto con actividades de soporte relacionadas con procesos de calidad y gestión.

3. RESULTADOS DE APRENDIZAJE

- Desarrolla un sistema de información fundamentado en el conocimiento de métodos, técnicas modernas, metodologías, buenas prácticas y estándares de calidad.
- Contribuye al trabajo en equipos multidisciplinarios, comunicándose asertivamente con los demás miembros y cumpliendo con las actividades del rol seleccionado según la metodología propuesta.

4. CONTENIDOS

I CORTE

1. INTRODUCCIÓN AL CURSO
 - a. Historia
 - b. Definición

	PROGRAMA ACADÉMICO DE LA ASIGNATURA	CÓDIGO: GFR-FOR-005
		VERSIÓN: 3
		FECHA ELABORACIÓN DEL DOCUMENTO: 15/ENE/2021

- c. Categorías y clasificación de los patrones
- d. Principios GRASP
- 2. PRINCIPIOS SOLID
 - a. S – Single Responsibility Principle (SRP)
 - b. O – Open/Closed Principle (OCP)
 - c. L – Liskov Substitution Principle (LSP)
 - d. I – Interface Segregation Principle (ISP)
 - e. D – Dependency Inversion Principle (DIP)
- 3. REPASO
 - a. Introducción a *Scrum*
 - i. Roles
 - ii. Ceremonias
 - 1. Scrum Daily
 - 2. Planning Sprint
 - 3. Sprint review
 - 4. Sprint retrospective
 - 5. Grooming
 - iii. Product Backlog - Sprints
 - 1. Herramientas: Jira y GitHub
 - 2. Planning poker
 - b. Calidad de software
 - i. Definiciones
 - ii. Pruebas unitarias en el framework que se implemente el proyecto
 - iii. Pruebas automatizadas en el framework que se implemente el proyecto
 - c. Otras metodologías
 - i. Kanban
 - ii. Lean
 - iii. otras

II CORTE

- 4. PATRONES GOF
 - a. Patrones de creación
 - i. Abstract Factory
 - ii. Builder
 - iii. Prototype
 - iv. Singleton.
 - b. Patrones estructurales
 - i. Adapter
 - ii. Decorator
 - iii. Fachade
 - c. Patrones de comportamiento
 - i. Memento
 - ii. State
 - iii. Visitor
- 5. PATRONES POSA
 - a. Patrones de diseño
 - i. Master-Slave
 - ii. Proxy

	PROGRAMA ACADÉMICO DE LA ASIGNATURA	CÓDIGO: GFR-FOR-005
		VERSIÓN: 3
		FECHA ELABORACIÓN DEL DOCUMENTO: 15/ENE/2021

iii. Command Processor

III CORTE

- b. Patrones de arquitectura
 - i. Layers
 - ii. Broker
 - iii. Model-View-Controller
 - iv. Data Access Object
 - v. SPA

5. ESTRATEGIAS DE ENSEÑANZA

En cada sesión del curso se discutirán los conceptos teóricos de cada tema, **que los estudiantes han preparado previamente**, y se desarrollarán talleres individuales o en grupos, con ejercicios de aplicación. La preparación de cada tema incluye la lectura del mismo, identificar las ideas más importantes y sobre todo establecer los puntos que necesitan explicación por parte del docente. Algunas lecturas y actividades serán en idioma inglés, considerando que gran parte de la producción en el tema es en dicho idioma.

Se establecerán espacios para asesorías, donde se resolverán dudas relacionadas con los temas de la materia y se realizará seguimiento al desarrollo de los talleres y del proyecto. Se contará con un aula virtual –en la plataforma institucional UAMVirtual– donde se publicará material del curso (estándares, talleres, lecturas) y las fechas para las diferentes actividades. En el aula se contará con un foro donde se pueden plantear discusiones relacionadas con temas de la materia.

Además del estudio individual de cada tema, los estudiantes deberán desarrollar un proyecto de software en equipo, donde se apliquen los conceptos vistos en el curso, junto con temas de investigación.

Como elementos investigativos adicionales, los estudiantes deberán utilizar herramientas de software apoyen el desarrollo del proyecto, y realizarán aportes sobre temas de actualidad relacionados con la Ingeniería de Software mediante herramientas Web 2.0.

6. EVALUACIÓN

CORTE	TIPO DE EVALUACIÓN	% PARCIAL	% TOTAL
I	Quiz	15%	30%
	Parcial	40%	
	Talleres	15%	
	Reto	30%	
II	Quiz	15%	35%
	Parcial	40%	

	PROGRAMA ACADÉMICO DE LA ASIGNATURA	CÓDIGO: GFR-FOR-005
		VERSIÓN: 3
		FECHA ELABORACIÓN DEL DOCUMENTO: 15/ENE/2021

III	Talleres	15%	
	Reto	30%	
	Quiz	10%	35%
	Parcial	40%	
	Talleres	15%	
	Reto final	35%	

7. BIBLIOGRAFÍA

- ALUR, Deepak; CRUPI, John y MALKS, Dan. *Core J2EE Patterns: Best Practices and Design Strategies*. Pearson Education, 2001.
- BUSCHMANN, Frank; MEUNIER, Regine; ROHNERT, Hans; SOMMERLAND, Peter; and STAL, Michael. *PATTERN-ORIENTED SOFTWARE ARCHITECTURE VOLUME 1: A System of Patterns*, John Wiley & Sons. 1996
- ERIKSSON, Hans – Erik. *UML 2 Toolkit*. Wiley, 2003.
- FOWLER, Martin. *Patterns of Enterprise Application Architecture*. Addison-Wesley, 2002.
- GAMMA, Erich, et. al. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1995
- GRAND, Mark. *Patterns in Java: A Catalog of Reusable Design Patterns Illustrated with UML*, Volume 1, Second Edition. Wiley Publishing, 2002
- YACCOUB, Sherif y AMMAR, Hany. *Pattern-Oriented Analysis and Design: Composing Patterns to Design Software Systems*. Addison-Wesley Professional, 2004.
- BENNET, Simon, McROBB, Steve y FARMER, Ray. *Análisis y diseño orientado a objetos de sistemas*, Tercera Edición. McGraw-Hill, 2006. [005.1 B355]
- BRUEGGE, Bernd y DUTOIT, Allen. *Ingeniería de software orientado a objetos*. Prentice Hall/Pearson, 2002. [005.1 B773]
- Object Oriented Software Engineering: using UML, Patterns and Java, 3 ed. Prentice Hall, 2009. [005.1 B773o]
- GAMMA, Erich et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Publishing Company, 1995. [005.12 G155]
- HAMBLING, Brian (editor). *Software Testing: An ISTQB-BCS Certified Tester Foundation Guide*, 3 ed. BCS, 2015. [005.14 S681]
- International Organization for Standardization ISO. *Software and Systems Engineering – Software Testing ISO-IEC-IEEE-29119*. ISO, 2007. [005.108 18 I573]
- OTERO, Carlos E. *Software Engineering Design: Theory and Practice*. CRC, 2012. [005.1 O737]
- PIATTINI, Mario; CALVO-MANZANO, José; CERVERA, Joaquín y FERNÁNDEZ, Luis. *Análisis y diseño de aplicaciones informáticas de gestión: Una perspectiva de ingeniería del software*. Alfaomega-Rama, 2004. [004.21 P417]
- PRESSMAN, Roger. *Ingeniería del software: Un enfoque práctico*, Séptima Edición. McGraw Hill Interamericana, 2010 [005.1 P737]

Recursos en Internet

- Scrum Reference Card. Enlace: <http://scrumreferencecard.com/>

	PROGRAMA ACADÉMICO DE LA ASIGNATURA	CÓDIGO: GFR-FOR-005
		VERSIÓN: 3
		FECHA ELABORACIÓN DEL DOCUMENTO: 15/ENE/2021

- Guía de Scrum. Enlace: <http://www.scrumguides.org/scrum-guide.html>

Enlaces Internet

- Core J2EE Pattern! <http://java.sun.com/blueprints/corej2eepatterns/>
- Microsoft Patterns & Practices: Patterns, <http://msdn2.microsoft.com/en-us/practices/bb190384.aspx>
- Object-Oriented Analysis and Design, Colorado University, <http://www.cs.colorado.edu/~kena/classes/6448/f08/lectures/>
- Patterns in Enterprise Software: <http://www.martinfowler.com/articles/enterprisePatterns.html>
- Patterns: <http://java.sun.com/blueprints/patterns/>