
 <b>UNIVERSIDAD DISTRITAL</b> <b>FRANCISCO JOSÉ DE CALDAS</b>	<b>FORMATO DE SYLLABUS</b>		Código: AA-FR-003	 <b>SIGUD</b> <small>Sistema Integrado de Gestión</small>
	Macroproceso: Direccionamiento Estratégico		Versión: 01	
	Proceso: Autoevaluación y Acreditación		Fecha de Aprobación: 27/07/2023	

<b>FACULTAD:</b>	<b>Ingeniería</b>		
<b>PROYECTO CURRICULAR:</b>	<b>Ingeniería de Sistemas</b>	<b>CÓDIGO PLAN DE ESTUDIOS:</b>	

### I. IDENTIFICACIÓN DEL ESPACIO ACADÉMICO

**NOMBRE DEL ESPACIO ACADÉMICO: Diseño de software**

Código del espacio académico:		Número de créditos académicos:	3			
Distribución horas de trabajo:	HTD	2	HTC	4	HTA	3
Tipo de espacio académico:	Asignatura	X	Cátedra			

### NATURALEZA DEL ESPACIO ACADÉMICO:

Obligatorio Básico		Obligatorio Complementario	x	Electivo Intrínseco		Electivo Extrínseco	
--------------------	--	----------------------------	---	---------------------	--	---------------------	--

### CARÁCTER DEL ESPACIO ACADÉMICO:

Teórico		Práctico		Teórico-Práctico	X	Otros:		Cuál: _____
---------	--	----------	--	------------------	---	--------	--	-------------

### MODALIDAD DE OFERTA DEL ESPACIO ACADÉMICO:

Presencial	X	Presencial con incorporación de TIC		Virtual		Otros:		Cuál: _____
------------	---	-------------------------------------	--	---------	--	--------	--	-------------

### II. SUGERENCIAS DE SABERES Y CONOCIMIENTOS PREVIOS

Fundamentos de ingeniería de software

### III. JUSTIFICACIÓN DEL ESPACIO ACADÉMICO

El diseño de Software ha logrado un crecimiento y madurez suficiente para establecerlo como espacio académico curricular de interés, para apoyar el rol del Ingeniero de Sistemas de la universidad Distrital FJC. En este espacio se propone la revisión de los fundamentos del diseño de software, el uso de estrategias de modelo basado en patrones y arquitecturas y el modelo desde un enfoque de capas que integran la pila de un proyecto de software iniciando desde la concepción organizacional, pasando por la estrategia, procesos de negocio, aplicación, tecnología hasta finalizar con el modelamiento propio del ciclo de vida arquitectónico. Ello genera un fuerte componente para el desarrollo de un proyecto de software.

### IV. OBJETIVOS DEL ESPACIO ACADÉMICO (GENERAL Y ESPECÍFICOS)

**Objetivo General:**  
Identificar los conceptos más importantes que definen el Diseño de Software basado en principios, patrones, modelos y lenguajes que lo caracterizan dentro del cuerpo de conocimiento de la Ingeniería de Software.

**Objetivos específicos:**

- \* Identificar los principios de software que fundamentan todo diseño.
- \* Modelar sistemas de software utilizando patrones de diseño
- \* Utilizar patrones arquitectónicos en la solución de problemas prácticos planteados como ejercicios de aplicación.
- \* Reconocer las malas prácticas en las que se incurren en un diseño desde el enfoque de antipatrones, planteando alternativas de solución.
- \* Integrar las diferentes perspectivas involucradas en el desarrollo de software mediante modelos propuestos desde las vistas propias de la organización hasta el planteamiento de infraestructura y ciclo de vida de la arquitectura de software.
- \* Evaluar el diseño realizando los ajustes necesarios dinámica y estructuralmente.

### V. PROPÓSITOS DE FORMACIÓN Y DE APRENDIZAJE (PFA) DEL ESPACIO ACADÉMICO

<b>Competencias</b>	<b>Dominio-Nivel</b>	<b>RA</b>	<b>Resultados de Aprendizaje</b>
---------------------	----------------------	-----------	----------------------------------

Genera artefactos basados en patrones, estilos y arquitecturas; cuyos modelos integrarán las diferentes perspectivas de un proyecto de software, hacer el seguimiento de su ciclo de vida y su respectiva verificación.	Cognitivo - conocer	1	Identificar los principios de software que fundamentan todo diseño, así como los antipatrones o malas prácticas de diseño.
	Cognitivo - Aplicar	2	Situar específicamente el rol de diseño en el que se integran la fundamentación propia para el modelado, así como estrategias basadas en patrones y modelos arquitectónicos
		3	Fortalecer la capacidad cognitiva desde una perspectiva de abstracción lograda desde el modelado
	Cognitivo - Evaluar	4	Tomar mejores decisiones por el carácter de anticipación que ofrece el diseño
		5	Construir artefactos propios del diseño e Integrar el Diseño de Software al desarrollo de proyectos
		6	Garantizar el ciclo de vida arquitectónico y la verificación del mismo en sus diferentes etapas
VI. CONTENIDOS TEMÁTICOS			

- Principios de Diseño
  - Principios Rectores
  - Principios Fundamentales
  - Principios de Fiabilidad
  - Principios SOLID & GRASP
  - Otros.
- Diseño Basado en Patrones
  - Patrones de Análisis, Interacción y Programación
  - Patrones de Diseño -GoF-
  - Patrones Concurrentes
  - Patrones de Arquitectura
- Antipatrones
  - Análisis
  - Codificación
  - Diseño
  - Otros
- Arquitectura de Software
  - Fundamentos de Arquitectura
  - Lenguajes y notaciones y estándares
  - Modelado por Puntos de Vista
- Evaluación del Diseño
  - Prueba de concepto
  - Análisis Estático y Dinámico

VII. ESTRATEGIAS DE ENSEÑANZA QUE FAVORECEN EL APRENDIZAJE						
Tradicional	X	Basado en Proyectos	X	Basado en Tecnología	X	
Basado en Problemas	X	Colaborativo	X	Experimental	X	
Aprendizaje Activo		Autodirigido		Centrado en el estudiante		
VIII. EVALUACIÓN						
Resultados de aprendizaje (RA) a ser evaluados:	Resultados de aprendizaje asociados a las evaluaciones					
	Actividades Entregables	Talleres	Parciales	Informes de proyecto final	Proyecto final	Exposiciones
RA01	X	X	X	X	X	X
RA02	X	X	X	X	X	X
RA03	X	X	X	X	X	X
RA04	X	X	X	X	X	X
RA05	X	X	X	X	X	X
RA06	X	X	X	X	X	X
Tipo de evaluación**						
Porcentaje de evaluación (%)	25	20	20	5	25	5
Trabajo Individual (I) o Grupal (G)	G	G	I	G	G	I/G
Tipo de nota	0-5	0-5	0-5	0-5	0-5	0-5
IX. MEDIOS Y RECURSOS EDUCATIVOS						
• Salón normal con pizarrón para sesiones de cátedra y para sesiones de discusión.						

- Acceso a Videobeam.
- Página web para publicar material didáctico, guías de trabajo, talleres, etc.
- Videos didácticos alrededor de los temas de la asignatura.
- Talleres investigativos y prácticos.
- Acceso al material bibliográfico recomendado
- \* Acceso a laboratorios de informática para el desarrollo de prácticas y proyectos

#### X. PRÁCTICAS ACADÉMICAS - SALIDAS DE CAMPO

No aplica

#### XI. BIBLIOGRAFÍA

##### Básicas:

- [1] Jayaratchagan N. Elegant Software Design Principles. Made in Usa, 2019.
- [2] Gamma E et al. Design Patterns. Addison Wesley, 1994.
- [3] Bushmann F. et al. Pattern-Oriented Software Architecture. Wiley. 2002.
- [4] J. Rumbaugh et al. Modelado y diseño orientados a objetos. Prentice Hall, 1996.
- [5] G. Booch. Object-oriented analysis and design with applications. Benjamin/Cummings Pub. Co, 1994.
- [6] I. Jacobson et al. Object-Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley, 1992.
- [7] R.C Lee y W.M Tepfenhart. UML AND C++, A PRACTICAL GUIDE TO OBJECT-ORIENTED DEVELOPMENT. Prentice Hall, 1997.
- [8] R.C. Martin. "Design Principles and Design Patterns". En: objectmentor.com (2000). url: [http://www.objectmentor.com/resources/articles/Principles\\_and\\_Patterns.pdf](http://www.objectmentor.com/resources/articles/Principles_and_Patterns.pdf).
- [9] B. Meyer. Construcción de software Orientado a Objetos. Prentice-Hall, 1998.
- [10] B. Liskov. "Data Abstraction and Hierarchy". En: OOPSLA 87 Addendum to the Proceedings (1987).
- [11] B. Liskov y J.M Wing. "A Behavioral Notion of Subtyping". En: ACM Transactions on Programming Languages and Systems (1994).
- [12] W. P. Stevens, G. J. Myers y L. L. Constantine. "Structured design". En: IBM Systems Journal 13.2 (1974), págs. 115-139. doi: 10.1147/sj.132.0115.
- [13] T.J. McCabe. "A Measure Complexity". En: IEEE Transactions on Software Engineering (1976). url: <https://martinfowler.com/articles/injection.html>.
- [14] Brooks F. The Mythical Man-Month. Addison-Wesley, 1995.
- [15] M. Shaw y D. Garlan. Software Architecture. Prentice Hall, 1996.

##### Complementarias:

- Pressman R. Software Engineering a practitioners approach, seventh edition. Mc Graw Hill, 2009.
- Somerville I. Ingeniería del software. Addison Wesley, 2005.

##### Revistas

- IEEE Transactions on Software Engineering.
- IEEE Software Engineering.
- Publicaciones de Elsevier en "Notes of Computer Science".
- Publicaciones de ACM en Ciencias de la Computación e Ingeniería de Software

##### Páginas web

#### XII. SEGUIMIENTO Y ACTUALIZACIÓN DEL SYLLABUS

Fecha revisión por Consejo Curricular:			
Fecha aprobación por Consejo Curricular:		Número de acta:	