

# Bienvenida

Apreciado estudiante, sea nuevamente bienvenido a un curso más de Ingeniería de Software, esta vez, Metodologías de Diseño!

En este curso abordaremos aspectos “menos blandos” de la ingeniería de software. A partir de los requerimientos que aprendimos a obtener en Metodologías de Análisis, ahora debemos aprender a elaborar soluciones técnicas, tomando decisiones de alto nivel y también de ingeniería de detalle. En este curso meteremos las manos al código, tanto propio como ajeno, para aprender a “ver el diseño” dentro del código, y también a documentarlo.

Uno de los principales objetivos es reconocer qué es diseño y cómo se hace, y para ello trabajaremos con ejemplos de código en distintos lenguajes, ya que el diseño es transversal a los lenguajes de programación. El segundo es aprender a **diseñar algo antes de hacerlo**, y a **diseñar mejoras para lo que ya está hecho**.

Nuevamente trabajaremos en equipo, por lo que te sugiero elegir bien con quién trabajarás el resto del semestre. No es obligación mantener ni el equipo ni el proyecto del curso anterior.

Cuenta con el profesor y la ayudante Rosa para cualquier consulta a través de medios formales (correo institucional), y con menor posibilidad de respuesta (pero no nula) por medios informales como Facebook.

Espero que disfrutes este curso, tal como los anteriores. Saludos!

Atte.  
El profesor.

## Programa de la asignatura

Sigla	Asignatura	Tema
INC404	Metodología de Diseño	NA

Prerrequisitos	Semestre	Grupo
INC313	Séptimo	Ciencias de la Computación

Semanas	Cátedras x Semana	Horas x cátedra	Cátedras x semestre
18	2	3	$18 * 2 = 36$

Horas x semestre	Horas ayudantía x semana	Horas laboratorio x semana
$18 * 2 * 3 = 108$	1,5	0

Descripción
<p><b>Objetivo General</b></p> <p>Comprender el proceso de diseño dentro del contexto del ciclo de vida del software; conocer y aplicar los principios y fundamentos del proceso de diseño; y conocer y aplicar metodologías de apoyo al proceso de diseño.</p> <p><b>Objetivos Específicos</b></p> <ul style="list-style-type: none"> <li>• Comprender qué es un problema de diseño, independiente de la tecnología de desarrollo (lenguaje de programación) y enfoque de modelado.</li> <li>• Entender, especificar y evaluar decisiones de diseño.</li> <li>• Conocer, seleccionar y evaluar estrategias de diseño dentro del contexto de un proceso de software.</li> </ul>

Metodología
<ul style="list-style-type: none"> <li>• Aprendizaje basado en problemas para actividades de aula.</li> <li>• Aprendizaje basado en proyectos como trabajo fuera del aula.</li> <li>• Controles de lectura para asentar conocimientos teóricos necesarios para actividades.</li> </ul>

Evaluación
------------

**NFProblemas:** Corresponde al promedio de las 5 actividades de aprendizaje realizadas y evaluadas en el aula. Cada actividad tiene la misma ponderación. **Equivalen a la nota de certámenes.**  
Por cada actividad: 30% nota personal evaluada por profesor, 50% nota grupal evaluada por profesor y pares (otros equipos), 20% nota personal de participación en equipo evaluada por compañeros de equipo.

**NFProyecto:** corresponde al promedio de las notas de cada hito del proyecto. Cada hito tiene la misma ponderación. Proyecto grupal, 4 integrantes. Todos con la misma evaluación.

Por cada hito: Informe: 50%, Presentación: 40%, Participación en presentaciones: 10%, ponderado por un factor de evaluación individual de parte de los compañeros de equipo.

**NFControles:** promedio de todos los controles de lectura realizado, menos el peor evaluado.

Toda actividad de trabajo en equipo será ponderada por una nota de evaluación de pares (secreta).

- NiPre: Nota individual de la presentación a preguntas del profesor (individual)
- NiPar: Nota de evaluación de pares dentro del grupo (individual)
- NgPre: Nota de la presentación evaluada por el profesor (grupal)
- NgPar: Nota de evaluación de pares de otros grupos (grupal)
- Nota Problema  $i = \sum_{I=1}^4 NP_i = (NiControl + NiPre)*30\% + NigPre*40\% + NigPar*10\% + NiPar*20\%$
- $NFProblemas = \sum_{I=1}^4 NP_i / 4$
- $NF: Nota Final = 0,4 * NFProyecto + 0,4 * NFProblemas + 0,2 * NFControles$

### Unidades

1. Fundamentos y principios de diseño.
2. Diseño arquitectónico.
3. Diseño orientado a objetos.
4. Diseño con reutilización.
5. Enfoques de Diseño de Software
6. Conceptos generales de Ingeniería de Software

Nota: los conceptos de las unidades serán abordados en forma transversal en cada problema abordado

### Bibliografía y Recursos

1. 1. Ingeniería de Software, Sommerville, I. Editorial Pearson Educación, 2002.

2. 2. Ingeniería de software clásica y orientada a objetos, Schach, Stephen R. Editorial McGraw-Hill, 2006.
3. Larman C. *Applying UML and Patterns: and Introduction to OO Analysis and Design*, Prentice Hall, 1998, 2002, 2005.
4. E.Gamma, R.Helm, R. Johnson, J. Vlissides. *Design Patterns, Elements of Reusable OO Software*. Addison-Wesley, 1995

## Planificación 2019

Semana	Clase	Tema	Actividad	Evaluación	Trabajo en la semana
12/03	1	Presentación del curso	Presentación del curso.		
	2		Diagnóstico	Revisión de Certamen 2 de Metodología de Análisis.	
<b>Problema 1: Identificando problemas de diseño</b>			<b>Temas asociados:</b> <ul style="list-style-type: none"> <li>- Paradigmas de programación</li> <li>- Calidad de diseño</li> <li>- Métricas de calidad de software</li> <li>- Métricas de calidad de diseño orientado a objetos</li> <li>- Diagramas UML de clases</li> </ul> <b>Herramientas:</b> <ul style="list-style-type: none"> <li>- Modelamiento UML</li> <li>- IDE - captura automatizada de métricas de software.</li> </ul>		
19/03	3	Parte 1	Presentación de código fuente y descripciones asociadas. Estudiantes deben descubrir potenciales problemas de diseño.	De proceso, observar a estudiantes.	
	4		Presentación de principales hallazgos. Discusión.	De proceso, observar a estudiantes.	Lectura para próxima clase: Principios de diseño. Actividad próxima clase: rediseñar el código para resolver el problema de diseño.
26/03	5	Parte 2	Análisis de problemas de diseño y su relación con principios de diseño	Control de lectura: Principios de diseño	
	6		Elaboración de propuestas de soluciones para problemas de diseño.	De proceso, observar a estudiantes.	Validación de propuestas de diseño con estudiantes de cursos superiores/con experiencia en desarrollo de software.
02/04	7	Parte 3	Presentación de soluciones y validación final.	Evaluación Problema 1	
	8		Presentación de soluciones y validación final.	Evaluación Problema 1	

<b>09/04</b>		<b>Semana Mechona</b>			
		<b>Semana Mechona</b>			
16/04	9	Hito 1	Presentación Hito 1	Presentación Hito 1	
	10	Hito 1	Presentación Hito 1	Presentación Hito 1	
<b>Problema 2: Aplicando Patrones de diseño GOF</b>					
23/04	11	Parte 1	Reconociendo la necesidad de patrones de diseño		
	12		Reconociendo la necesidad de patrones de diseño		
30/04		Parte 2	Patrones de Diseño GOF	Control de lectura 2	
			Patrones de Diseño GOF		
07/05	13	Parte 2	Patrones de Diseño GOF	Ejemplos Patrones de Diseño mas usados. Clase Taller	
	14		Patrones de Diseño GOF		
14/05	15	Parte 3	Presentación de soluciones y validación final.	Evaluación Problema 2	
	16		Presentación de soluciones y validación final.		
28/05	17	Hito 2	Presentación Hito 2		
	18	Hito 2	Presentación Hito 2		
<b>Problema 3: Separando la lógica de la presentación: MVC</b>			<b>Tema asociados:</b> <ul style="list-style-type: none"> <li>- Patrones de diseño OO</li> <li>- Diagramas UML de interacción</li> <li>- Buenas prácticas de codificación (coding standards)</li> </ul> <b>Herramientas:</b> <ul style="list-style-type: none"> <li>- IDE-Debugger</li> <li>- Modelamiento UML</li> <li>- Versionamiento GIT</li> </ul>		
04/06	19	Parte 1	GIT-GITHUB		
	20				
11/06	21	Parte 2	MVC		
	22				
18/06	23	Parte 3	Evaluación Teórica	Control de Lectura 3	

	24		Evaluación Práctica	Evaluación Problema 3	
25/06	25	Hito 3	Presentación Hito 3		
	26	Hito 3	Presentación Hito 3		
<b>Problema 4: Refactoring.</b>			Temas asociados: - Aplicación de patrones de diseño a código entregado con malas prácticas. Herramientas: - IDEs - Test de junit		<b>Temas asociados:</b> - Diseño arquitectónico/ de alto nivel - Diseño distribuido - Diagramas UML de componentes y despliegue. <b>Herramientas:</b> - Modelamiento UML
02/07	27	Parte 1	Clase Componentes, despliegue, frameworks		
	28				
09/07	29	Parte 2		Control de lectura 4	
	30		Presentación Frameworks	<b>Evaluación Problema 4</b>	

# Proyecto

1.- Considerando las particularidades de cada proyecto y el resultado del análisis, desarrollar un plan de Construcción del Software, en un plazo de 4 meses. El plan debe considerar 3 iteraciones, la definición de qué funcionalidades de van a abordar en cada una de ellas, y una primera estimación de esfuerzo en programación y construcción.

2.- En cada entrega, se utilizarán distintas técnicas para realizar tanto el diseño de alto nivel como el diseño detallado:

- Entrega 1: Plan de Desarrollo, a tres iteraciones. Estimación de Esfuerzo y Equipo (sin método formal). Elección de tecnologías (Tiene que ser orientado a objetos) (frameworks incluyendo).
- Entrega 2: "CRUD" recorriendo toda la arquitectura. Definición de Arquitectura, documentación UML (Diagrama de Despliegue y Componentes)
- Entrega 3: Construcción de funcionalidad. Documentación UML de Diseño Detallado (clases, diagramas de secuencia).
- Entrega 4: Construcción de funcionalidad. Documentación UML de Diseño Detallado (clases, diagramas de secuencia, Diagrama de Despliegue y Componentes)

## Planificación

El proyecto cuenta con cuatro hitos de entrega.

## Evaluación de Pares

- Por cada hito: Informe: 50%, Presentación: 40%, Participación en otras presentaciones: 10%.
- En cada hito se solicitará una evaluación de pares dentro del equipo. Esta se considerará como un factor de ajuste para la evaluación grupal. Este factor podrá ser de 50% a 110%, con la siguiente escala:

<b>Nota Evaluación de Pares de Equipo</b>	<b>Porcentaje aplicado a nota grupal</b>
1.0	50%
2.0	60%
3.0	70%
4.0	80%
5.0	90%

6.0	100%
7.0	110%

- La nota final del hito es el promedio de la nota final de cada hito.

**EGL/20191S**