

**DISEÑO DE SOFTWARE  
2024-2**

**I. INFORMACIÓN GENERAL**

CURSO	DISEÑO DE SOFTWARE
CLAVE	1INF50
CRÉDITOS	2.5
HORAS DE DICTADO	CLASE: 2 Semanal LABORATORIO: 2 Quincenal EXAMEN:
HORARIO	TODOS
PROFESORES	NATALÍ FLORES LAFOSSE DENNIS STEPHEN COHN MUROY

**II. PLANES CURRICULARES DONDE SE DICTA EL CURSO**

ESPECIALIDAD	ETAPA	NIVEL	CARÁCTER	REQUISITOS
INGENIERÍA INFORMÁTICA	PREGRADO EN FACULTAD	7	OBLIGATORIO	1INF31 ARQUITECTURA DE SOFTWARE [07]

**Tipos de requisito**

- 04 = Haber cursado o cursar simultáneamente
- 05 = Haber aprobado o cursar simultáneamente
- 06 = Promedio de notas no menor de 08
- 07 = Haber aprobado el curso

**III. DESCRIPCIÓN DEL CURSO**

Este curso aporta a las siguientes competencias de la carrera de Ingeniería Informática:

1. Diseño en ingeniería: Analiza, diseña, implementa e implanta soluciones informáticas considerando los componentes de software y hardware, haciendo uso de tecnologías emergentes e integradas a otros dominios, para facilitar el uso de las funcionalidades y contenidos, satisfaciendo con calidad, seguridad y confiabilidad las necesidades y requerimientos de clientes y/o usuarios. Se desempeña eficazmente como parte de un equipo, estableciendo estrategias para un plan de acción que permita alcanzar los objetivos.
2. Comunicación eficaz: Comunica eficazmente ideas con claridad, coherencia y consistencia usando un lenguaje formal, oral o escrito, de acuerdo con diferentes audiencias.
3. Trabajo en equipo: Se desempeña eficazmente como parte de un equipo, estableciendo estrategias para un plan de acción que permita alcanzar los objetivos.

**IV. SUMILLA**

El curso es de naturaleza teórico-práctico cuyo propósito es que el estudiante, trabajando de manera individual y en equipo, diseñe y documente los componentes del software y verifique su implementación. Se desarrolla los temas de conceptos de diseño de software, patrones de diseño, modelos de diseño y casos de prueba

**V. OBJETIVOS**

El curso contribuye al logro de los siguientes Resultados de Aprendizaje

RA 1: Selecciona de manera colaborativa los patrones de diseño adecuados a aplicar según los requerimientos que debe cubrir cada componente específico de la solución respetando la topología de la arquitectura.

RA 2: Selecciona de manera colaborativa los patrones de persistencia de datos a aplicar según la topología de la arquitectura y las necesidades de la solución.

RA 3: Diagrama modelos que representen el diseño detallado de cada componente utilizando el lenguaje de modelado UML.

RA 4: Evalúa de manera colaborativa los tipos de casos de prueba a incluir dentro del proceso de desarrollo de software de acuerdo con el contexto elegido.

RA 5: Diseña casos de prueba que le permitan verificar el funcionamiento de los componentes de acuerdo con los requerimientos funcionales y no funcionales.

## **VI. PROGRAMA ANALÍTICO**

### **UNIDAD 1 DISEÑO DE SOFTWARE (16 horas)**

Capítulo 1 (2 horas):

- Fundamentos del Diseño de Software.

Capítulo 2 (4 horas):

- Notación UML: Diagramas de Clases de Diseño
- Notación UML: Diagramas de Interacción

Capítulo 3 (10 horas):

- Patrones de Diseño
- Patrones de Persistencia
- Patrones de Interacción con Base de Datos
- Antipatrones

### **UNIDAD 2 PRUEBAS DE SOFTWARE (12 horas)**

Capítulo 4 (12 horas):

- Fundamentos de Pruebas de Software
- Tipos de Prueba
- Diseño de Pruebas
- TDD y Desarrollo Ágil
- Pruebas de Componentes

## **VII. METODOLOGÍA**

El curso se organiza en clases teóricas-prácticas y en experiencias prácticas de laboratorio. En las clases teóricas-prácticas, ya sea en aula o sincrónicas (videoconferencia), se tendrá una dinámica basada en aprendizaje activo, significativo y colaborativo. Por ello, se espera que el estudiante responda a las preguntas, ejercicios, casos y problemas que deja el docente para aplicar los contenidos explicados. En los laboratorios, se privilegia el aprendizaje colaborativo, mediante el cual los estudiantes propondrán un problema para analizarlo y dar alternativas de solución empleando los conceptos y herramientas vistas en el curso, de forma activa y participativa con sus compañeros, contando con la orientación del jefe de práctica.

En relación con recursos y herramientas: se señala el uso de pizarra acrílica, equipo multimedia, internet y una bibliografía complementaria como materiales de la clase teórica; así como el uso de Zoom, pizarras digitales y software de diagramación o de tipo CASE para las clases virtuales. Los laboratorios emplearán software de ofimática y software de diagramación o de tipo CASE. En general, para colocar los materiales, guías de laboratorio, bibliografía complementaria del curso, realizar entregas y otras actividades como evaluaciones se empleará la plataforma virtual Paideia.

## **VIII. EVALUACIÓN**

### **Sistema de evaluación**

Nº	Codigo	Tipo de Evaluación	Cant. Eval.	Forma de aplicar los pesos	Pesos	Cant. Eval. Eliminables	Consideraciones adicionales	Observaciones
----	--------	--------------------	-------------	----------------------------	-------	-------------------------	-----------------------------	---------------

**FACULTAD DE CIENCIAS E INGENIERÍA**  
**1INF50 - DISEÑO DE SOFTWARE**

1	Pb	Práctica tipo B	5	Por Evaluación	Pb1=150 Pb2=150 Pb3=150 Pb4=150 Pb5=150			
2	Ex	Examen	2	Por Evaluación	Ex1=125 Ex2=125			

**Modalidad de evaluación: 2**

**Fórmula para el cálculo de la nota final**

$$(150Pb1 + 150Pb2 + 150Pb3 + 150Pb4 + 150Pb5 + 125Ex1 + 125Ex2) / 1000$$

Aproximación de la nota final No definido

**Consideraciones adicionales**

- En caso el estudiante no pueda rendir una Evaluación Continua, debe justificar por correo al docente como máximo al día siguiente. El docente evaluará la reprogramación.
- Se recomienda que los estudiantes asistan a por lo menos el 75% de las clases. Algunas sesiones, debidamente informadas, contarán con actividades calificadas.
- Para la calificación en los laboratorios se tomarán en cuenta los siguientes criterios:
  - \* Presentación del avance correspondiente a la sesión del laboratorio.
  - \* Levantar las observaciones del laboratorio previo.
  - \* Autoevaluación de cada estudiante del grupo.
  - \* Individualmente se evaluará el aporte de los compañeros del grupo.

## IX. BIBLIOGRAFÍA

**Referencia obligatoria**

- Libro  
Fowler, M.  
2012  
Patterns of Enterprise Application Architecture  
*Addison-Wesley.*
- Libro  
Gamma, E., Helm, R., Johnson, R., & Vlissides, J.  
1995  
Design patterns: elements of reusable object-oriented software.  
*Pearson Deutschland GmbH.*
- Libro  
Sommerville, I.  
2011  
Software engineering (ed.). America  
*Pearson Education Inc.*
- Libro  
Spillner, A., & Linz, T.  
2021  
Software Testing Foundations: A Study Guide for the Certified Tester Exam-Foundation Level-ISTQB¿ Compliant  
*dpunkt. verlag*

**Referencia complementaria**

- Libro  
Freeman, E., & Robson, E.  
2020  
Head First Design Patterns  
*O'Reilly Media*.
- Libro  
Gough, J., Bryant, D., & Auburn, M.  
2021  
Mastering API Architecture.
- Libro  
Medjaoui, M., Wilde, E., Mitra, R., & Amundsen, M.  
2021  
Continuous API management.
- Página web  
Design patterns and refactoring. (n.d.-b).  
<https://sourcemaking.com/antipatterns>

## **X. POLÍTICA CONTRA EL PLAGIO**

Para la corrección y evaluación de todos los trabajos del curso se va a tomar en cuenta el debido respeto a los derechos de autor, castigando severamente cualquier indicio de plagio con la nota CERO (00). Estas medidas serán independientes del proceso administrativo de sanción que la facultad estime conveniente de acuerdo a cada caso en particular. Para obtener más información, referirse a los siguientes sitios en internet

[www.pucp.edu.pe/documento/pucp/plagio.pdf](http://www.pucp.edu.pe/documento/pucp/plagio.pdf)