

### LABORATORIO SOFTWARE AVANZADO

CÓDIGO	<b>0780</b>	CRÉDITOS	<b>6</b>
ESCUELA	<b>Ciencias y Sistemas</b>	ÁREA	<b>Desarrollo de Software</b>
PRE REQUISITO	<b>0785</b>	POST REQUISITOS	<b>Ninguno</b>
CATEGORÍA	<b>Obligatorio</b>	SEMESTRE	<b>Primer Semestre 2025</b>
CATEDRÁTICO(A):	<b>Ing. Everest Medinilla</b>	AUXILIARES	<b>Julio Roberto Vasquez Santiago</b>
HORAS POR SEMANA DEL CURSO	<b>4</b>	MINUTOS QUE SE IMPARTE EL LABORATORIO	<b>100</b>
DÍAS QUE SE IMPARTE EL CURSO	<b>Martes y Jueves</b>	DÍAS QUE SE IMPARTE EL LABORATORIO	<b>Sabado</b>
HORARIO DEL CURSO	<b>7:10 - 8:50</b>	HORARIO DEL LABORATORIO	<b>09:40 - 11:20</b>

#### **DESCRIPCIÓN DEL CURSO:**

Software Avanzado es un curso profesional que pertenece al área de software de la carrera de Ingeniería en Ciencias y Sistemas, el cual trata sobre conceptos fundamentales de ingeniería de software, se tiene especial énfasis en tecnologías modernas en la nube, devops y metodologías ágiles.

#### **OBJETIVOS LABORATORIO:**

Reconocer la importancia de aplicar ingeniería de software durante el desarrollo de un proyecto y un producto.

- Poner en práctica técnicas de orquestación de procesos en arquitectura orientada a servicios
- Visualizar el avance de ejecución el desarrollo de un producto a través de métricas estándar de código y equipos de trabajo.

### **METODOLOGÍA:**

- Clases teóricas de conceptos generales.
- Clase práctica de cómo realizar implementación de tecnologías específicas.
- Desarrollo de Tareas prácticas y autoaprendizaje.
- Desarrollo de un proyecto final.

### **EVALUACIÓN DEL RENDIMIENTO ACADÉMICO:**

La nota mínima para aprobar el laboratorio es de 61 puntos de un total de

#### **100. Tareas Prácticas 30 puntos**

Práctica 1	05%
Práctica 2	05%
Práctica 3	05%
Práctica 4	10%
Práctica 5	10%
Práctica 6	10%
Práctica 7	12.5%
Práctica 8	12.5%
Práctica 9	15%
Práctica 10	15%

#### **2 exámenes cortos 5 puntos (2.5 pts. c/u)**

#### **Proyecto 60 puntos**

Fase 1	25%
Fase 2	35%
Fase 3	40%

#### **Examen final 5 puntos**

#### **Total 100 pts.**

### **CONTENIDO DEL LABORATORIO:**

#### 1. Fundamentos de Desarrollo

- Metodología Ágil
  - Principios de la metodología ágil
  - Scrum vs. Kanban
- Herramientas de frontend y backend
  - Frameworks de frontend
  - Backend
  - Bases de datos

- Data en caché
  - APIs REST
  - Autenticación y autorización (OAuth, JWT)
  - Modelo de computación en la nube
    - Computación en la nube: Modelos de servicio (IaaS, PaaS, SaaS)
    - Introducción a AWS, GCP y Azure
    - Costos y escalabilidad en la nube
    - Despliegue de aplicaciones en la nube
  - Docker
    - Imágenes y contenedores
    - Dockerfile y construcción de imágenes personalizadas
    - Volúmenes y redes en Docker
    - Docker Compose para gestión de múltiples contenedores
    - Registro de imágenes (DockerHub, ECR, GCR)
2. Arquitectura de Software, Orquestación
- Kubernetes
    - Conceptos básicos de Kubernetes
    - Pods, Servicios y Deployments
    - ConfigMaps y Secrets
    - Gestión de almacenamiento en Kubernetes (Persistent Volumes)
3. DevOps
- Introducción a DevOps
  - Plan & Code
  - Build & Test
  - Deploy & Release
    - CI/CD
  - Testing
  - Logging & Monitoring
4. DevSecOps: Seguridad en el ciclo de vida del software

**PUNTOS IMPORTANTES A CONSIDERAR:**

- Para tener derecho a nota de laboratorio se debe cumplir con el 80% de asistencia a clase de laboratorio al menos que se presente una constancia médica de una institución pública o privada donde indique la suspensión por enfermedad.
- La tarea que no se entregue a tiempo no tendrá recepción posterior

**BIBLIOGRAFIA:**

[1] Roger S. Presman. Ingeniería de Software. Un enfoque práctico. McGraw Hill, Quinta Edición. E.U.A., 2007.

[2] Documentos Elaborados por Catedrático del Curso. Ian Sommerville. Ingeniería de Software. Prentice Hall. 7ma edición.

[3] Adair, J., Decision Making and Problem Solving Strategies, 2nd Ed., Kogan Page, E.U.A. 2007.

[4] Gerald Kontoya and Ian Sommerville, Requirements Engineering - Process and Techniques

[5] Erl, Thomas. SOA Principles of Service Design. Prentice Hall. 2008.

[6] HashiCorp. Terraform.

<https://www.terraform.io> Docker docs.

<https://docs.docker.com>