

LABORATORIO SOFTWARE AVANZADO

CÓDIGO	0780	CRÉDITOS	6
ESCUELA	Ciencias y Sistemas	ÁREA	Desarrollo de Software
PRE REQUISITO	0785	POST REQUISITOS	Ninguno
CATEGORÍA	Obligatorio	SEMESTRE	Segundo Semestre 2026
CATEDRÁTICO (A)	Marco Tulio Aldana Prillwitz	AUXILIAR	Kevin Pozuelos Estrada
EDIFICIO		SECCIÓN	P
SALÓN DEL CURSO	Curso a distancia	SALÓN DEL LABORATORIO	Curso a distancia
HORAS POR SEMANA DEL CURSO	4	HORAS POR SEMANA DEL LABORATORIO	2
DÍAS QUE SE IMPARTE EL CURSO	Miércoles y Viernes	DÍAS QUE SE IMPARTE EL LABORATORIO	Jueves
HORARIO DEL CURSO	17:20 - 19:00	HORARIO DEL LABORATORIO	17:20 - 19:00

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 del 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.
- Entrega de tareas y tareas prácticas.
- 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.		
DESCRIPCIÓN	PUNTOS	%
Tareas Prácticas (total)	30	
Práctica 1	-	05%
Práctica 2	-	05%
Práctica 3	-	05%
Práctica 4	-	05%
Práctica 5	-	12.5%
Práctica 6	-	12.5%
Práctica 7	-	12.5%
Práctica 8	-	12.5%
Práctica 9		15%
Práctica 10	-	15%
2 Exámenes cortos (2.5 pts. c/u)	5	
Proyecto (total)	60	
Fase 1	-	25%
Fase 2	-	35%
Fase 3	-	40%
Examen final	5	
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 • loop engineering • harness engineering1 • 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, a 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.

BIBLIOGRAFÍA:

- Roger S. Presman. Ingeniería de Software. Un enfoque práctico. McGraw Hill, Quinta Edición. E.U.A., 2007.
- Documentos Elaborados por Catedrático del Curso. Ian Sommerville. Ingeniería de Software. Prentice Hall. 7ma edición.
- Adair, J., Decision Making and Problem Solving Strategies, 2nd Ed., Kogan Page, E.U.A. 2007.
- Gerald Kontoya and Ian Sommerville, Requirements Engineering - Process and Techniques.
- Erl, Thomas. SOA Principles of Service Design. Prentice Hall. 2008.
- HashiCorp. Terraform. <https://www.terraform.io>
- Docker docs. <https://docs.docker.com>