



**Laboratorio de análisis y diseño de sistemas 2**

<b>Código</b> 785	<b>Créditos</b> 5	
<b>Escuela</b> Ciencias y Sistemas	<b>Área a la que pertenece</b> Área de desarrollo de Software	<b>Vigencia</b> Segundo semestre 2021
<b>Horas por semana</b> 2	<b>Horario laboratorio</b> Sábado 8:50-10:30	
<b>Pre-requisitos:</b> 283-Análisis y Diseño de Sistemas 1		<b>Post-requisitos:</b> 780-Software avanzado
<b>Catedrático:</b> Yessenia Jannira Marroquín Martínez		<b>Tutor:</b> David Omar Enriquez Reyes
<b>Sección:</b> O		
<b>Descripción del laboratorio</b>		
<p>El laboratorio de Análisis y diseño de sistemas 2 es la continuación y el complemento de lo que se aprendió en el curso de Análisis y diseño de sistemas 1, es la aplicación de los conceptos y/o conocimientos adquiridos, pero en la práctica.</p> <p>Muchas empresas a la hora de desarrollar software no tienen implementada ninguna metodología de desarrollo de software, incluso los departamentos se mantienen aislados unos de otros, esto es una desventaja y conlleva a posibles fracasos en los proyectos o incluso al fracaso de la organización.</p> <p>El uso correcto de buenas prácticas y metodologías en conjunto con herramientas adecuadas para el desarrollo de un software nos permite tener mayores ventajas y éxito en la elaboración de proyectos.</p> <p>El curso de análisis y diseño de sistemas 2 permite tener un amplio conocimiento de las buenas prácticas y en el laboratorio se conocerán las herramientas necesarias para que dichas prácticas sean correctamente aplicadas. Una de las prácticas que se manejara en dicho curso es <b>DevOps</b>.</p>		
<b>Objetivos</b>		
<b>General:</b>		
Lograr que el estudiante adquiriera los conocimientos necesarios para poder analizar y diseñar un sistema de acuerdo con las tecnologías y herramientas más recientes, adoptando para ello buenas prácticas y metodologías de desarrollo.		
<b>Específicos:</b>		
<ul style="list-style-type: none"><li>• Que el estudiante comprenda el proceso para efectuar la entrega continua de un software.</li><li>• Que el estudiante comprenda el proceso para efectuar el despliegue continuo de un software.</li><li>• Que el estudiante conozca y ponga en práctica los conceptos de arquitectura de software abarcados durante el curso.</li><li>• Que el estudiante comprenda y practique los distintos patrones de diseño que existen.</li><li>• Familiarizar al estudiante con las herramientas disponibles para aplicar un completo y correcto desarrollo de software.</li></ul>		
<b>Habilidades</b>		
<ul style="list-style-type: none"><li>• Que el estudiante investigue, comprenda y aplique los conocimientos adquiridos durante cursos anteriores, para diseñar y analizar sistemas de forma óptima.</li><li>• Analizar los requerimientos de un software para la realización de un sistema que cumpla con todas las etapas del ciclo de vida del software.</li></ul>		

- Extraer y representar el conocimiento necesario para construir e implementar una solución para un proyecto de software cumpliendo prácticas de administración de la configuración y el diseño de soluciones informáticas a un problema propuesto.

### Competencias

- Que el estudiante pueda aplicar una buena administración de la configuración.
- Capacidad para analizar y determinar requerimientos de software.
- Resolver y reconocer problemas clásicos de la administración de proyectos.
- Capacidad para crear y utilizar herramientas de integración continua, control de versiones y pruebas de software.
- Capacidad de adaptarse a cambios en los requerimientos iniciales de software.

### Metodología

- Se impartirán exposiciones virtuales en salas meet con temas enfocados a la práctica de los conceptos del curso.
- El laboratorio estará dividido en una parte teórica y en una parte práctica, para que la reunión sea un poco más dinámica y los conceptos puedan ser aplicados.
- Se realizarán exámenes cortos como parte de la evaluación de prácticas y contenido impartido en el laboratorio.
- Se realizarán prácticas donde se busca que el estudiante conozca y experimente con las distintas herramientas mencionadas a lo largo del contenido del laboratorio, aplicándolas sobre problemas reales.
- Se realizarán tareas y hojas de trabajo para que el estudiante pueda complementar y comprender de mejor manera los temas tratados.

### Contenido

Tópico	Subtemas	Herramientas
Administración de la configuración	<ul style="list-style-type: none"> <li>• Control de Versiones</li> <li>• Herramientas de control de versiones</li> <li>• Integración Continua</li> <li>• Ciclo general de la integración continua</li> <li>• Herramienta para integración continua</li> </ul>	<ul style="list-style-type: none"> <li>▪ Git</li> <li>▪ GitLab</li> <li>▪ Nodejs</li> <li>▪ Docker</li> </ul>
Entrega continua	<ul style="list-style-type: none"> <li>• Ambientes del software</li> <li>• Flujo de entrega de software</li> <li>• Componentes de la entrega continúa</li> </ul>	<ul style="list-style-type: none"> <li>▪ GitLab-runner</li> <li>▪ Docker-compose</li> </ul>
Pruebas	<ul style="list-style-type: none"> <li>• Pruebas funcionales <ul style="list-style-type: none"> <li>○ Pruebas unitarias</li> <li>○ Pruebas de aceptación</li> <li>○ Pruebas de integración</li> <li>○ Pruebas de regresión</li> </ul> </li> <li>• Pruebas no funcionales: <ul style="list-style-type: none"> <li>○ Pruebas de carga</li> <li>○ Pruebas de estrés</li> <li>○ Pruebas de escalabilidad</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>▪ Selenium</li> <li>▪ K6</li> <li>▪ Mocha</li> </ul>

- Pruebas de portabilidad

---

Arquitectura de software	<ul style="list-style-type: none"><li>• Arquitectura por capas</li><li>• Arquitectura 4 + 1 vistas</li><li>• MVC</li></ul>
--------------------------	--

---

Arquitecturas orientadas al servicio	<ul style="list-style-type: none"><li>• SOA</li><li>• Web services y SOA</li><li>• Cloud Computing</li></ul>	<ul style="list-style-type: none"><li>▪ Google cloud platform</li><li>▪ Kubernetes</li></ul>
--------------------------------------	--	--

---

Patrones de diseño	<ul style="list-style-type: none"><li>• Patrones de creación</li><li>• Patrones de estructura</li><li>• Patrones de comportamiento</li></ul>
--------------------	--

---

### **Evaluación**

El laboratorio se aprueba con una nota mayor o igual a 61 puntos

- Proyecto (3 fases) (100 puntos)
  - Fase 1, 33 puntos
  - Fase 2, 33 puntos
  - Fase 3, 34 puntos

### **VIII. Observaciones:**

- Solo se calificarán las actividades de estudiantes asignados al curso. NO se agregan estudiantes en actas.
- Los equipos para trabajar las practicas son de mínimo 3 personas, esto para fomentar el trabajo colaborativo y serán formados por el tutor académico, si algún integrante se queda sin grupoluego de una entrega puede adherirse a otro grupo luego de discutirlo con el tutor y con el grupo al que se va a adherir. NO SE CALIFICARÁN TRABAJOS INDIVIDUALES EN NINGUNA CIRCUNSTANCIA.
- El laboratorio se calificará sobre 100, y será equivalente a 30 puntos de zona.
- Habrá 1 proyecto dividido en tres fases que se realizará en grupo.
- El catedrático revisará las notas obtenidas en el curso y el laboratorio. Podrá decidir si es necesaria una segunda revisión a cada proyecto y considerar nuevamente la ponderación obtenida en cada proyecto.
- Las notas de cada proyecto serán publicadas por el catedrático del curso en el transcurso del semestre, el estudiante tendrá 8 días como máximo para pedir revisión de proyecto.
- El laboratorio debe aprobarse con 61 puntos sobre 100.
- Es obligatorio ganar el laboratorio para tener derecho a evaluación final del curso.
- No habrá proyecto de retrasada, ni reposición de nota de laboratorio. Al final del semestre, no se asignarán trabajos extra para recuperar puntos de zona.
- El curso se aprueba con 61 puntos.
- Las entregas fuera de fecha no son aceptadas.
- Debe existir respeto por las opiniones de los demás.
- Como estudiantes universitarios, se espera que sepan y entiendan las normas de educación, respeto, ética y plagio relacionadas con trabajos de otros autores y con el desarrollo del curso.

### **IX. Normas para clase virtual**

- Todas las Comunicaciones con el profesor y los auxiliares deben ser por los correos electrónicos que se indiquen en clase.
- En toda comunicación escrita se debe mostrar respeto y no utilizar mensajes en mayúsculas.
- Las comunicaciones enviadas por correo electrónico serán atendidas en un máximo de 3 días hábiles.
- Durante los exámenes los estudiantes deben mantener encendida su cámara y estar conectados a la sesión de Google Meet durante todo el tiempo de evaluación.
- Durante las clases los estudiantes deben encender su cámara siempre que el profesor o el auxiliar les hagan una pregunta directa, o bien, cuando el estudiante realice alguna consulta.

<ul style="list-style-type: none"> <li>• Durante las clases los estudiantes pueden hacer consultas por el chat del curso o por la opción de Questions / Answers, según lo indique el profesor, teniendo el cuidado de ser respetuoso y mantener las reglas de cortesía durante la escritura.</li> </ul>	
Bibliografía	<ul style="list-style-type: none"> <li>▪ "Version Control with Subversion", Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michal Pilato.</li> <li>▪ "Essential Software Architecture", Ian Gorton</li> <li>▪ "Headfirst Design Patterns", Elisabeth Freeman, Eric Freeman, Bert Bates, Kathy Sierra</li> <li>▪ "Continuous Delivery", Jez Humble, David Farley</li> <li>▪ "Service Oriented Architecture, Concepts, Technology and Design", Thomas Er</li> </ul>
No. De Secciones	1
Director de Escuela	Ing. Carlos Alonzo