



Laboratorio de análisis y diseño de sistemas 2

Código 785	Créditos 5	
Escuela Ciencias y Sistemas	Área a la que pertenece Área de desarrollo de Software	Vigencia Segundo semestre 2021
Horas por semana 2	Horario laboratorio Sábado 8:50-10:30	
Pre-requisitos: 283-Análisis y Diseño de Sistemas 1		Post-requisitos: 780-Software avanzado
Catedrático: Claudia Liceth Rojas Morales		Tutor: Mónica Raquel Calderón Muñoz
Sección: O		
Descripción del laboratorio		
<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 DevOps.</p>		
Objetivos		
General:		
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.		
Específicos:		
<ul style="list-style-type: none">Que el estudiante comprenda el proceso para efectuar la entrega continua de un software.Que el estudiante comprenda el proceso para efectuar el despliegue continuo de un software.Que el estudiante conozca y ponga en práctica los conceptos de arquitectura de software abarcados durante el curso.Que el estudiante comprenda y practique los distintos patrones de diseño que existen.Familiarizar al estudiante con las herramientas disponibles para aplicar un completo y correcto desarrollo de software.		
Habilidades		
<ul style="list-style-type: none">Que el estudiante investigue, comprenda y aplique los conocimientos adquiridos durante cursos anteriores, para diseñar y analizar sistemas de forma óptima.Analizar los requerimientos de un software para la realización de un sistema que cumpla con todas las etapas del ciclo de vida del software.		

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

- Pruebas de portabilidad

Arquitectura de software	<ul style="list-style-type: none">• Arquitectura por capas• Arquitectura 4 + 1 vistas• MVC
Arquitecturas orientadas al servicio	<ul style="list-style-type: none">• SOA• Web services y SOA• Cloud Computing <ul style="list-style-type: none">▪ Google cloud platform▪ Kubernetes
Patrones de diseño	<ul style="list-style-type: none">• Patrones de creación• Patrones de estructura• Patrones de comportamiento

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"> • 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. 	
Bibliografía	<ul style="list-style-type: none"> ▪ “Version Control with Subversion”, Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michal Pilato. ▪ “Essential Software Architecture”, Ian Gorton ▪ “Headfirst Design Patterns”, Elisabeth Freeman, Eric Freeman, Bert Bates, Kathy Sierra ▪ “Continuous Delivery”, Jez Humble, David Farley ▪ “Service Oriented Architecture, Concepts, Technology and Design”, Thomas Er
No. De Secciones	1
Director de Escuela	Ing. Carlos Alonzo