

PROGRAMA DE LABORATORIO

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS



Análisis y Diseño de Sistemas 2

CÓDIGO:	785	PONDERACIÓN:	5 CRÉDITOS
ESCUELA DE INGENIERÍA EN:	CIENCIAS Y SISTEMAS	ÁREA A LA QUE PERTENECE:	DESARROLLO DE SOFTWARE
PRE-REQUISITO:	283 - Análisis y Diseño de Sistemas 1	POST REQUISITO:	0780 – Software Avanzado
CATEGORÍA:	OBLIGATORIO	VIGENCIA:	SEGUNDO SEMESTRE 2025
HORAS POR SEMANA DEL CURSO:	4	HORAS POR SEMANA DEL LABORATORIO:	2
HORAS DE AUTOAPRENDIZAJE:	285	TOTAL DE HORAS DE APRENDIZAJE:	22
CATEDRÁTICO (A):	MIRNA IVONNE ALDANA LARRAZABAL	AUXILIAR:	MELYZA ALEJANDRA RODRIGUEZ CONTRERAS
EDIFICIO:	MEET	SECCIÓN:	B
SALÓN DEL CURSO:	MEET	SALON DEL LABORATORIO:	MEET
DÍAS QUE SE IMPARTE EL CURSO:	MARTES Y JUEVES	DÍAS QUE SE IMPARTE EL LABORATORIO:	JUEVES
HORARIO DEL CURSO:	07:10 AM – 08:50 AM	HORARIO DEL LABORATORIO:	17:20 – 19:00

Breve descripción del Laboratorio

El curso Análisis y Diseño de Sistemas 2, representa una parte muy importante de la carrera de Ingeniería en Ciencias y Sistemas que aborda las herramientas necesarias para llevar a cabo el desarrollo de proyectos de software alineado a las buenas prácticas aceptadas para este propósito.

Este curso brinda conocimientos acerca de las diferentes metodologías de desarrollo, así como de arquitecturas y patrones de diseño, herramientas de ingeniería de software y los conceptos de Despliegue e integración continua. Se implementan técnicas de análisis y diseño que buscan un correcto desempeño para un correcto planteamiento, configuración, elaboración de pruebas y gestión de proyectos.

Índice

Contenido

Competencias Vinculadas al Perfil del Egresado.....	5
Competencias Específicas	5
Competencias Generales	5
Competencias del Laboratorio.....	6
Competencia(s) Específica(s)	6
Competencia(s) General(es).....	6
Diseño Didáctico por Competencias	7
Sesión de Diagnóstico.....	7
Evaluación de conocimientos previos	7
Presentación del tutor	7
Presentación de los estudiantes	7
Presentación del programa del curso.....	8
Evaluación de conocimientos del laboratorio actual.....	8
Sesión No. 1, Unidad No. 1 - Frameworks	9
Valor de la semana (Saber ser)	9
Conocimiento (Saber)	9
Habilidades (Saber Hacer)	9
Sesión No. 2, Unidad No. 2 – Patrones de diseño.....	10
Valor de la semana (Saber ser)	10
Conocimiento (Saber)	10
Habilidades (Saber Hacer)	10
Sesión No. 3, Unidad No. 2 – Patrones de diseño	11
Valor de la semana (Saber ser)	11
Conocimiento (Saber)	11
Habilidades (Saber Hacer)	11
Sesión No. 4, Unidad No. 2 – Patrones de diseño	12
Valor de la semana (Saber ser)	12
Conocimiento (Saber)	12
Habilidades (Saber Hacer)	12
Sesión No. 5, Unidad No. 2 – Patrones de diseño	13
Valor de la semana (Saber ser)	13

Conocimiento (Saber)	13
Habilidades (Saber Hacer)	13
Sesión No. 6, Unidad No. 3 – Elementos de diseño	14
Valor de la semana (Saber ser)	14
Conocimiento (Saber)	14
Habilidades (Saber Hacer)	14
Sesión No. 7, Unidad No. 3 – Elementos de diseño	15
Valor de la semana (Saber ser)	15
Conocimiento (Saber)	15
Habilidades (Saber Hacer)	15
Sesión No. 8, Unidad No. 4 – Pruebas de software	16
Valor de la semana (Saber ser)	16
Conocimiento (Saber)	16
Habilidades (Saber Hacer)	16
Sesión No. 9, Unidad No. 4 – Pruebas de software	17
Valor de la semana (Saber ser)	17
Conocimiento (Saber)	17
Habilidades (Saber Hacer)	17
Sesión No. 10, Unidad No. 5 - Seguridad	18
Valor de la semana (Saber ser)	18
Conocimiento (Saber)	18
Habilidades (Saber Hacer)	18
Sesión No. 11, Unidad No. 6 – Despliegue en la nube (Cloud)	19
Valor de la semana (Saber ser)	19
Conocimiento (Saber)	19
Habilidades (Saber Hacer)	19
Sesión No. 12, Unidad No. 6 - Despliegue en la nube (Cloud)	20
Valor de la semana (Saber ser)	20
Conocimiento (Saber)	20
Habilidades (Saber Hacer)	20
Sesión No. 13, Unidad No. 6 - Despliegue en la nube (Cloud)	21
Valor de la semana (Saber ser)	21
Conocimiento (Saber)	21
Habilidades (Saber Hacer)	21
Sesión No. 14, Unidad No. 6 - Despliegue en la nube (Cloud)	22

Valor de la semana (Saber ser)	22
Conocimiento (Saber)	22
Habilidades (Saber Hacer)	22
Tiempo de Auto-aprendizaje.....	23
Rúbrica de Evaluación.....	23
Resumen de Ponderaciones.....	23
Normativa Académica y Ética del Curso	23
Equipo Académico.....	25
Coordinador del Área.....	25
Sección A.....	25
Sección B.....	26
Bibliografía	28
E-Grafía	28

Competencias Vinculadas al Perfil del Egresado

Competencias Específicas

No.	Competencia
1	Aplica los conocimientos de su disciplina en la elaboración, fundamentación y defensa de argumentos para prevenir y resolver problemas complejos en su campo profesional, identificando y aplicando innovaciones.
2	Toma decisiones profesionales con base en fundamentos teóricos, datos e información pertinente, válida y confiable.
3	Identifica oportunidades y riesgos para la innovación y adaptación de conocimientos y tecnologías para resolver problemas.
4	Demuestra destreza y habilidad en la selección, uso y adaptación de herramientas metodológicas, tecnológicas, equipos especializados y en la lectura e interpretación de datos, pertinentes al contexto de su ejercicio profesional.

Competencias Generales

No.	Competencia
1	Aplica principios básicos de ingeniería, ciencias de la computación y sistemas de información y comunicación en la formulación y resolución adecuada de problemas complejos.
2	Aplica conocimientos tecnológicos con ética profesional, respetando y cuidando los recursos naturales, humanos y financieros.
3	Aplica estándares de calidad, eficiencia y seguridad en la implementación adecuada de soluciones de software, hardware y tecnologías de la información y comunicación (TIC) en general.
4	Maneja e interpreta adecuadamente datos masivos, sean estos estructurados o no estructurados, facilitando su visualización e interpretación de forma eficaz en apoyo a la toma de decisiones.

Competencias del Laboratorio

Competencia(s) Específica(s)

No.	Competencia	Nivel de Aprendizaje
1	Identifica y describe conceptos de frameworks aplicando casos de uso y características comunes para seleccionar adecuadamente un framework de acuerdo al proyecto.	Comprender
2	Diseña y aplica patrones de diseño de software utilizando principios de diseño y catálogos de estilos, garantizando la calidad estructural y de comportamiento del sistema.	Aplicar y crear
3	Implementa y evalúa pruebas funcionales y no funcionales empleando técnicas de pruebas unitarias, de carga y de seguridad, asegurando el correcto funcionamiento y desempeño del software.	Evaluar y aplicar
4	Aplica y adapta modelos de despliegue en la nube mediante arquitecturas por capas, microservicios y plataformas como Docker y Terraform, optimizando la disponibilidad y escalabilidad de los sistemas.	Aplicar
5	Integra y automatiza procesos DevOps utilizando herramientas de integración y entrega continua, con el propósito de mejorar el ciclo de vida de las aplicaciones.	Aplicar y analizar

Competencia(s) General(es)

No.	Competencia	Nivel de Aprendizaje
1	Identifica y describe conceptos de frameworks aplicando casos de uso y características comunes, con el fin de seleccionar adecuadamente un framework de acuerdo con las necesidades del proyecto.	Crear
2	Diseña, implementa y evalúa arquitecturas de software aplicando frameworks, patrones de diseño, metodologías de pruebas y despliegue en la nube utilizando herramientas colaborativas y de automatización bajo estándares de calidad, seguridad y buenas prácticas de desarrollo de software	Aplicar y analizar

Diseño Didáctico por Competencias

Esta sección organiza las sesiones del laboratorio en función de las competencias que el estudiante debe desarrollar. Cada clase incluye valores (saber ser), contenidos teóricos (saber) y habilidades prácticas (saber hacer), permitiendo un aprendizaje integral y aplicado. Las actividades están alineadas con los objetivos del curso y el perfil del egresado.

Sesión de Diagnóstico

Evaluación de conocimientos previos

Se aplicará una actividad diagnóstica con el objetivo de identificar el nivel de conocimientos y habilidades que los estudiantes poseen al inicio del curso. No influye en la nota final, pero es obligatoria para todos los estudiantes.

Tipo de Actividad	Descripción
Evaluación de conocimiento previos a través de la plataforma de kahoot	Se aplicará un cuestionario interactivo mediante la plataforma Kahoot para diagnosticar los conocimientos previos del estudiante sobre los temas vistos en Análisis y diseño de sistemas 1. Esta actividad busca identificar conceptos clave ya adquiridos y detectar posibles vacíos de aprendizaje, fomentando al mismo tiempo la participación y lúdica desde el inicio del curso.

Presentación del tutor

El tutor se presenta formalmente al grupo, compartiendo su formación académica, experiencia profesional y educativa, así como sus expectativas sobre el curso. También se abordan aspectos como normas de convivencia, canales de comunicación, disponibilidad para consultas y métodos de acompañamiento.

Presentación de los estudiantes

Se escogen un grupo de estudiantes al azar. En su presentación, se les pedirá que compartan información básica como su nombre, intereses personales o profesionales, experiencias previas relacionadas con el curso y sus expectativas. Esta actividad busca promover la interacción, el reconocimiento entre pares y la construcción de un entorno participativo y respetuoso.

Presentación del programa del curso

Se presenta el contenido del programa del curso, se aclaran dudas y se fomenta el compromiso del estudiante con su aprendizaje.

Evaluación de conocimientos del laboratorio actual

Se realiza una evaluación o práctica que permite conocer el grado de familiaridad de los estudiantes con las herramientas, entornos o competencias técnicas necesarias para el laboratorio actual.

<i>Tipo de Actividad</i>	<i>Descripción</i>
Evaluación de conocimiento actuales del curso a través de la plataforma de kahoot	Se realizará una evaluación interactiva a través de la plataforma Kahoot con el objetivo de medir el nivel de comprensión y dominio de los contenidos abordados durante el curso. Esta actividad permitirá identificar fortalezas y áreas de mejora en el aprendizaje, promoviendo la participación dinámica y el refuerzo de los temas clave.

Sesión No. 1, Unidad No. 1 - Frameworks

Valor de la semana (Saber ser)

Nombre: Responsabilidad
La responsabilidad es fundamental al seleccionar y aplicar un framework, ya que implica tomar decisiones técnicas conscientes, evaluando sus características, beneficios y compatibilidad con los requerimientos del proyecto. Un uso adecuado de frameworks demuestra compromiso con la calidad del software, la eficiencia del desarrollo y la sostenibilidad del código a largo plazo.

Conocimiento (Saber)

Competencia(s)	
Identifica y describe conceptos de frameworks aplicando casos de uso y características comunes para seleccionar adecuadamente un framework de acuerdo al proyecto.	
Tema	Subtema
Frameworks	Qué es un framework
Frameworks	Tipos de framework
Frameworks	Beneficios del uso de frameworks
Frameworks	Características comunes de los frameworks
Frameworks	Casos de uso específicos
Frameworks	Diferencias entre framework y herramientas

Habilidades (Saber Hacer)

Competencia	Tipo de Actividad	Ponderación
Identifica y describe conceptos de frameworks aplicando casos de uso y características comunes para seleccionar adecuadamente un framework de acuerdo con el proyecto.	Cuestionario corto	0

Sesión No. 2, Unidad No. 2 – Patrones de diseño

Valor de la semana (Saber ser)

Nombre: Compromiso
El compromiso permite al estudiante asumir con seriedad el diseño de soluciones eficientes y sostenibles, entendiendo que la correcta aplicación de patrones y principios de diseño impacta directamente en la calidad del software. Diseñar bien no solo es una habilidad técnica, sino una muestra de responsabilidad profesional con los usuarios, el equipo y el producto final.

Conocimiento (Saber)

Competencia(s)	
Diseña y aplica patrones de diseño de software utilizando principios de diseño y catálogos de estilos garantizando la calidad estructural y de comportamiento del sistema.	
Tema	Subtema
Elementos de diseño	Patrones de diseño
Elementos de diseño	Definición de patrones de diseño
Elementos de diseño	Importancia y Aplicabilidad de Patrones de diseño
Elementos de diseño	Notaciones de patrones de diseño
Elementos de diseño	Contexto de diseño
Elementos de diseño	Principios de diseño de software

Habilidades (Saber Hacer)

Competencia	Tipo de Actividad	Ponderación
Diseña y aplica patrones de diseño de software utilizando principios de diseño y catálogos de estilos garantizando la calidad estructural y de comportamiento del sistema.	Cuestionario corto	0

Sesión No. 3, Unidad No. 2 – Patrones de diseño

Valor de la semana (Saber ser)

Nombre: Disciplina
La disciplina es clave para comprender y aplicar correctamente los patrones creacionales, ya que requiere atención al detalle, seguimiento de estructuras definidas y constancia para entender su funcionamiento y aplicabilidad en distintos contextos de desarrollo.

Conocimiento (Saber)

Competencia(s)	
Diseña y aplica patrones de diseño de software utilizando principios de diseño y catálogos de estilos garantizando la calidad estructural y de comportamiento del sistema.	
Tema	Subtema
Elementos de diseño	Creacional
Elementos de diseño	Factory Method
Elementos de diseño	Abstract Factory
Elementos de diseño	Singleton
Elementos de diseño	Creacional
Elementos de diseño	Builder
Elementos de diseño	Prototype

Habilidades (Saber Hacer)

Competencia	Tipo de Actividad	Ponderación
Diseña y aplica patrones de diseño de software utilizando principios de diseño y catálogos de estilos garantizando la calidad estructural y de comportamiento del sistema.	Cuestionario corto	0

Sesión No. 4, Unidad No. 2 – Patrones de diseño

Valor de la semana (Saber ser)

Nombre: Colaboración
La colaboración es fundamental al aplicar patrones estructurales, ya que su esencia es mejorar la interacción segura y eficiente entre objetos o clases. Fomenta el trabajo en equipo, el modularidad y el respeto mutuo entre componentes del sistema, valores esenciales para un diseño sólido y mantenible.

Conocimiento (Saber)

Competencia(s)	
Diseña y aplica patrones de diseño de software utilizando principios de diseño y catálogos de estilos garantizando la calidad estructural y de comportamiento del sistema.	
Tema	Subtema
Elementos de diseño	Estructurales
Elementos de diseño	Adapter
Elementos de diseño	Decorator
Elementos de diseño	Facade
Elementos de diseño	Proxy

Habilidades (Saber Hacer)

Competencia	Tipo de Actividad	Ponderación
Diseña y aplica patrones de diseño de software utilizando principios de diseño y catálogos de estilos garantizando la calidad estructural y de comportamiento del sistema.	Cuestionario corto	0

Sesión No. 5, Unidad No. 2 – Patrones de diseño

Valor de la semana (Saber ser)

Nombre: Adaptabilidad
La adaptabilidad es esencial al aplicar patrones de comportamiento, ya que estos se centran en la forma en que los objetos se comunican y reaccionan ante los cambios. Desarrollar esta cualidad permite a los estudiantes diseñar soluciones más flexibles y preparadas para entornos dinámicos y cambiantes.

Conocimiento (Saber)

Competencia	
Diseña y aplica patrones de diseño de software utilizando principios de diseño y catálogos de estilos garantizando la calidad estructural y de comportamiento del sistema.	
Tema	Subtema
Elementos de diseño	Patrones de comportamiento
Elementos de diseño	Observer
Elementos de diseño	Strategy
Elementos de diseño	Command
Elementos de diseño	Iterator
Elementos de diseño	State

Habilidades (Saber Hacer)

Competencia	Tipo de Actividad	Ponderación
Diseña y aplica patrones de diseño de software utilizando principios de diseño y catálogos de estilos garantizando la calidad estructural y de comportamiento del sistema.	Cuestionario corto	0

Sesión No. 6, Unidad No. 3 – Elementos de diseño

Valor de la semana (Saber ser)

Nombre: Empatía
La empatía es clave en el diseño centrado en el usuario. Comprender las necesidades, emociones y expectativas del usuario permite diseñar experiencias más accesibles, útiles y satisfactorias. Fomentar este valor impulsa soluciones tecnológicas humanas y responsables.

Conocimiento (Saber)

Competencia	
Diseña y aplica patrones de diseño de software utilizando principios de diseño y catálogos de estilos garantizando la calidad estructural y de comportamiento del sistema.	
Tema	Subtema
Estrategias en Diseño de Interfaces y Arquitectura de la Información	Catálogo de estilos
Estrategias en Diseño de Interfaces y Arquitectura de la Información	Arquitectura de la información
Estrategias en Diseño de Interfaces y Arquitectura de la Información	Experiencia de usuario (UX)
Estrategias en Diseño de Interfaces y Arquitectura de la Información	Diseño de interfaces de usuario (UI)

Habilidades (Saber Hacer)

Competencia	Tipo de Actividad	Ponderación
Diseña y aplica patrones de diseño de software utilizando principios de diseño y catálogos de estilos garantizando la calidad estructural y de comportamiento del sistema.	Cuestionario corto	0

Sesión No. 7, Unidad No. 3 – Elementos de diseño

Valor de la semana (Saber ser)

Nombre: Pensamiento crítico
El pensamiento crítico es esencial al seleccionar y aplicar patrones de arquitectura, ya que permite evaluar de forma objetiva los requerimientos del sistema, las ventajas y desventajas de cada enfoque arquitectónico, y tomar decisiones informadas que impactan en la escalabilidad, flexibilidad y mantenibilidad de las soluciones tecnológicas.

Conocimiento (Saber)

Competencia	
Diseña y aplica patrones de diseño de software utilizando principios de diseño y catálogos de estilos garantizando la calidad estructural y de comportamiento del sistema.	
Tema	Subtema
Patrones de arquitectura	Centrada en Datos
Patrones de arquitectura	Centrada en el flujo de datos
Patrones de arquitectura	Arquitectura orientada a eventos
Patrones de arquitectura	Patrones de arquitectura empresarial
Patrones de arquitectura	SOA (Service-Oriented Architecture)
Patrones de arquitectura	EDA (Event-Driven Architecture)
Patrones de arquitectura	Arquitecturas híbridas

Habilidades (Saber Hacer)

Competencia	Tipo de Actividad	Ponderación
Diseña y aplica patrones de diseño de software utilizando principios de diseño y catálogos de estilos garantizando la calidad estructural y de comportamiento del sistema.	Cuestionario corto	0

Sesión No. 8, Unidad No. 4 – Pruebas de software

Valor de la semana (Saber ser)

Nombre: Responsabilidad
La responsabilidad es clave en la ejecución de pruebas funcionales, ya que estas aseguran que el sistema cumple con los requisitos definidos. Ser responsable implica validar el correcto funcionamiento del software desde la perspectiva del usuario, detectando errores antes de que lleguen al entorno productivo.

Conocimiento (Saber)

Competencia	
Implementa y evalúa pruebas funcionales y no funcionales empleando técnicas de pruebas unitarias, de carga y de seguridad asegurando el correcto funcionamiento y desempeño del software.	
Tema	Subtema
Pruebas funcionales	¿Qué son las pruebas funcionales?
Pruebas funcionales	Conceptos y principios de las pruebas funcionales
Pruebas funcionales	Importancia y aplicabilidad dentro de una arquitectura de software
Pruebas funcionales	¿Qué son las pruebas unitarias?
Pruebas funcionales	¿Qué son pruebas de aceptación?
Pruebas funcionales	¿Qué son las pruebas de integración?
Pruebas funcionales	¿Qué son las pruebas de regresión?

Habilidades (Saber Hacer)

Competencia	Tipo de Actividad	Ponderación
Implementa y evalúa pruebas funcionales y no funcionales empleando técnicas de pruebas unitarias, de carga y de seguridad asegurando el correcto funcionamiento y desempeño del software.	Cuestionario corto	0

Sesión No. 9, Unidad No. 4 – Pruebas de software

Valor de la semana (Saber ser)

Nombre: Rigor técnico
El rigor técnico permite ejecutar pruebas no funcionales con precisión, objetividad y atención al detalle. Este valor es esencial para garantizar la calidad del sistema en aspectos críticos como rendimiento, escalabilidad, portabilidad y estabilidad, más allá de que simplemente "funcione".

Conocimiento (Saber)

Competencia	
Implementa y evalúa pruebas funcionales y no funcionales empleando técnicas de pruebas unitarias, de carga y de seguridad asegurando el correcto funcionamiento y desempeño del software.	
Tema	Subtema
Pruebas no funcionales	¿Qué son las pruebas no funcionales?
Pruebas no funcionales	Conceptos y principios de las pruebas no funcionales
Pruebas no funcionales	Importancia y aplicabilidad dentro de una arquitectura de software
Pruebas no funcionales	¿Qué son las pruebas de carga?
Pruebas no funcionales	¿Qué son las pruebas de estrés?
Pruebas no funcionales	¿Qué son las pruebas de escalabilidad?
Pruebas no funcionales	¿Qué son las pruebas de portabilidad?

Habilidades (Saber Hacer)

Competencia	Tipo de Actividad	Ponderación
Implementa y evalúa pruebas funcionales y no funcionales empleando técnicas de pruebas unitarias, de carga y de seguridad asegurando el correcto funcionamiento y desempeño del software.	Cuestionario corto	0

Sesión No. 10, Unidad No. 5 - Seguridad

Valor de la semana (Saber ser)

Nombre: Ética
La ética es fundamental en el ámbito de la seguridad, ya que el acceso y protección de la información implican una gran responsabilidad. Diseñar sistemas seguros y proteger los datos exige actuar con integridad, respetando la privacidad y cumpliendo con las normas legales y profesionales.

Conocimiento (Saber)

Competencia	
Implementa y evalúa pruebas funcionales y no funcionales empleando técnicas de pruebas unitarias, de carga y de seguridad asegurando el correcto funcionamiento y desempeño del software.	
Tema	Subtema
Introducción a la seguridad de sistemas	Introducción a la seguridad de sistemas
Introducción a la seguridad de sistemas	Modelado de Amenazas
Introducción a la seguridad de sistemas	Diseño seguro de Software
Introducción a la seguridad de sistemas	Criptografía y Protección de Datos
Introducción a la seguridad de sistemas	Monitorización y Respuesta a Incidentes de Seguridad
Introducción a la seguridad de sistemas	Pruebas de seguridad
Introducción a la seguridad de sistemas	Cumplimiento y Normativas de Seguridad

Habilidades (Saber Hacer)

Competencia	Tipo de Actividad	Ponderación
Implementa y evalúa pruebas funcionales y no funcionales empleando técnicas de pruebas unitarias, de carga y de seguridad asegurando el correcto funcionamiento y desempeño del software.	Cuestionario corto	0

Sesión No. 11, Unidad No. 6 – Despliegue en la nube (Cloud)

Valor de la semana (Saber ser)

Nombre: Proactividad
La proactividad es esencial en el despliegue en la nube, ya que anticiparse a fallos, automatizar procesos y optimizar recursos requiere iniciativa, aprendizaje constante y capacidad para tomar decisiones oportunas en entornos cambiantes.

Conocimiento (Saber)

Competencia	
Aplica y adapta modelos de despliegue en la nube mediante arquitecturas por capas, microservicios y plataformas como Docker y Terraform optimizando la disponibilidad y escalabilidad de los sistemas.	
Tema	Subtema
Despliegue en la nube (cloud)	Introducción a la nube
Despliegue en la nube (cloud)	Modelos de servicio: IaaS, PaaS, SaaS
Despliegue en la nube (cloud)	Contenedores y Docker
Despliegue en la nube (cloud)	CDN (Content Delivery Network)
Despliegue en la nube (cloud)	Análisis de Costos en Nube (Cost Management)
Despliegue en la nube (cloud)	Módulos y Reutilización de Código en Terraform
Despliegue en la nube (cloud)	Ansible Playbooks y Roles para Automatización Compleja

Habilidades (Saber Hacer)

Competencia	Tipo de Actividad	Ponderación
Aplica y adapta modelos de despliegue en la nube mediante arquitecturas por capas, microservicios y plataformas como Docker y Terraform optimizando la disponibilidad y escalabilidad de los sistemas.	Cuestionario corto	0

Sesión No. 12, Unidad No. 6 - Despliegue en la nube (Cloud)

Valor de la semana (Saber ser)

Nombre: Responsabilidad
La responsabilidad es clave en el despliegue de soluciones en la nube, ya que implica el compromiso de seguir buenas prácticas, proteger los recursos, garantizar la seguridad de los datos y asumir las consecuencias de las decisiones técnicas que afectan la operación de los sistemas.

Conocimiento (Saber)

Competencia	
Aplica y adapta modelos de despliegue en la nube mediante arquitecturas por capas, microservicios y plataformas como Docker y Terraform optimizando la disponibilidad y escalabilidad de los sistemas.	
Tema	Subtema
Despliegue en la nube (cloud)	Introducción a DevOps y CI/CD
Despliegue en la nube (cloud)	Beneficios DevOps
Despliegue en la nube (cloud)	Principios Devops
Despliegue en la nube (cloud)	Plan/ Code / Build / Test
Despliegue en la nube (cloud)	Release / Deploy / Operate / Monitor

Habilidades (Saber Hacer)

Competencia	Tipo de Actividad	Ponderación
Aplica y adapta modelos de despliegue en la nube mediante arquitecturas por capas, microservicios y plataformas como Docker, Terraform y Ansible optimizando la disponibilidad y escalabilidad de los sistemas.	Cuestionario corto	0

Sesión No. 13, Unidad No. 6 - Despliegue en la nube (Cloud)

Valor de la semana (Saber ser)

Nombre: Rigor
El rigor en el despliegue en la nube implica actuar con precisión, responsabilidad y adherencia a buenas prácticas técnicas. Requiere analizar cada cambio en la infraestructura, validar el impacto de las decisiones y mantener un control estricto sobre la automatización, asegurando calidad, estabilidad y trazabilidad en entornos productivos.

Conocimiento (Saber)

Competencia	
Aplica y adapta modelos de despliegue en la nube mediante arquitecturas por capas, microservicios y plataformas como Docker y Terraform optimizando la disponibilidad y escalabilidad de los sistemas.	
Tema	Subtema
Despliegue en la nube (cloud)	¿Qué es IaC?
Despliegue en la nube (cloud)	Imperativo vs declarativo
Despliegue en la nube (cloud)	Beneficios: repetible, auditable, automatizable
Despliegue en la nube (cloud)	¿Qué es Terraform?
Despliegue en la nube (cloud)	Ventajas de Terraform
Despliegue en la nube (cloud)	Principios de Terraform
Despliegue en la nube (cloud)	Sintaxis y recursos de Terraform
Despliegue en la nube (cloud)	Flujos de trabajo de Terraform

Habilidades (Saber Hacer)

Competencia	Tipo de Actividad	Ponderación
Aplica y adapta modelos de despliegue en la nube mediante arquitecturas por capas, microservicios y plataformas como Docker, Terraform y Ansible optimizando la disponibilidad y escalabilidad de los sistemas.	Cuestionario corto	0

Sesión No. 14, Unidad No. 6 - Despliegue en la nube (Cloud)

Valor de la semana (Saber ser)

Nombre: Disciplina
La disciplina es clave en la automatización de procesos en la nube, ya que requiere seguir procedimientos, mantener estructuras ordenadas y ejecutar tareas con constancia y precisión. Al utilizar herramientas como Ansible, la disciplina permite desarrollar playbooks claros, aplicar buenas prácticas y asegurar la correcta implementación de la infraestructura, evitando errores que podrían afectar entornos críticos.

Conocimiento (Saber)

Competencia	
Aplica y adapta modelos de despliegue en la nube mediante arquitecturas por capas, microservicios y plataformas como Docker y Terraform optimizando la disponibilidad y escalabilidad de los sistemas.	
Tema	Subtema
Despliegue en la nube (cloud)	¿Qué es Ansible y cómo funciona?
Despliegue en la nube (cloud)	Beneficios de Ansible para la automatización
Despliegue en la nube (cloud)	Casos de uso de Ansible en diferentes industrias
Despliegue en la nube (cloud)	Conceptos clave (Playbooks, Módulos e Inventarios)
Despliegue en la nube (cloud)	Configuración e implementación de Ansible
Despliegue en la nube (cloud)	Orquestación de infraestructura con Ansible

Habilidades (Saber Hacer)

Competencia	Tipo de Actividad	Ponderación
Aplica y adapta modelos de despliegue en la nube mediante arquitecturas por capas, microservicios y plataformas como Docker, Terraform y Ansible optimizando la disponibilidad y escalabilidad de los sistemas.	Cuestionario corto	0

Tiempo de Auto-aprendizaje

Tipo	Horas de Auto-aprendizaje
Proyectos	225
Prácticas	60
Total	285

Rúbrica de Evaluación

Cada una de las actividades del laboratorio (proyectos, prácticas, tareas y otras) cuenta con una rúbrica de evaluación específica, la cual está detallada en el documento que se entrega al estudiante al momento de asignar la actividad. Estas rúbricas describen los criterios de evaluación, niveles de desempeño esperados y la ponderación correspondiente de cada aspecto evaluado.

Es **responsabilidad del estudiante** leer detenidamente la rúbrica asignada antes de iniciar el desarrollo de la actividad. Comprender los criterios de evaluación no solo permite orientar adecuadamente el trabajo, sino también mejorar el desempeño académico y fomentar la autorregulación del aprendizaje.

En caso de no recibir la rúbrica al momento de la asignación, el estudiante **debe solicitarla directamente al tutor académico**, ya que constituye una herramienta esencial para el cumplimiento de los objetivos de aprendizaje y la evaluación transparente.

Resumen de Ponderaciones

Tipo	Valor
Proyectos	90
Prácticas	10
Total	100

Normativa Académica y Ética del Curso

En concordancia con el perfil del estudiante de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, se espera un alto nivel de compromiso con la excelencia académica y la ética profesional. Por ello, que se establece los siguientes lineamientos de carácter obligatorio que regulan el comportamiento académico del estudiante:

Plagio y copias

- Todo proyecto será sometido a verificación para confirmar su autoría y originalidad, con la finalidad de evitar cualquier plagio, copia o que la actividad no haya sido realizada por el estudiante.
- Cualquier evidencia de lo antes descrito en las distintas actividades será sancionada con una calificación de 0 (cero) y el caso será reportado al Docente quien a su vez informará a la Escuela de Ciencias y Sistemas para su seguimiento institucional.

Prórrogas y reposiciones

- No se otorgarán prórrogas para entregas de actividades.
- No se permitirá la reposición de proyectos bajo ninguna circunstancia.

Requisitos para evaluación final del curso

- Es obligatorio aprobar el laboratorio para tener derecho a la evaluación final del curso.
- La calificación de prácticas, proyectos y otras actividades que se indique será asignada de forma presencial, en la fecha y hora establecidas por el tutor académico.

Asistencia

- Para obtener la nota del laboratorio, se requiere un mínimo del 80% de asistencia a las sesiones de laboratorio.
- En caso de inasistencia, sólo se aceptarán justificaciones válidas respaldadas por constancia oficial.

Entregas

- No se aceptarán entregas tardías de tareas, prácticas, exámenes cortos, exámenes finales o proyectos sin justificación.

Medio oficial de entrega

- La plataforma UEDI de la Facultad será el único medio oficial para la entrega de actividades del curso.

Equipo Académico

Coordinador del Área

Nombre: Ing. Marlon Francisco Orellana López	Correo electrónico:
--	---------------------

Sección A

Docente

CLAUDIA LICETH ROJAS MORALES	claudiarojasmorales@gmail.com
------------------------------	-------------------------------

	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Día	X	X				
Horario	17:20 – 19:00	17:20 – 19:00				
Lugar	MEET	MEET				

Tutor(es)

Nombre del Tutor	Mynor Francisco Morán García	
Correo electrónico institucional	2992105610101@ingenieria.usac.edut.gt	

Tipo		Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Clase	Día				X		
	Horario				17:20 – 19:00		
	Lugar				MEET		
Atención al	Día						

Estudiante	Horario						
	Lugar						

Sección B

Docente

MIRNA IVONNE ALDANA LARRAZABAL	ivonnealdanal@gmail.com
--------------------------------	-------------------------

	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Día		X		X		
Horario		07:10 - 8:50		07:10 - 8:50		
Lugar		MEET		MEET		

Tutor(es)

Nombre del Tutor	Melyza Alejandra Rodriguez Contreras	
Correo electrónico institucional	2593602600101@ingenieria.usac.edu.gt	

Tipo		Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Clase	Día				X		
	Horario				17:20 - 19:00		
	Lugar				MEET		
Atención al Estudiante	Día						
	Horario						

	Lugar						
--	-------	--	--	--	--	--	--

Bibliografía

- Essential software architecture, Ian Gorton
- Just enough software architecture, George Fairbanks
- Head first design patterns, Elisabeth Freeman, Eric Freeman, Bert Bates, Kathy Sierra
- Software systems architecture, working with stakeholders using viewpoints and perspectives, Nick Rozanski, Eoin Woods
- Software performance and scalability: a quantitative approach, Henry H. Liu
- Software Architecture Patterns. Second Edition. Mark Richards
- Software Quality Engineering, Jeff Tian
- Software Architecture in Practice. Len Bass, Paul Clements, Rick

E-Grafía

- Refactoring Guru. (s.f.) Patrones de diseño. <https://refactoring.guru/design-patterns/>