

Programa del curso CE5508

SOA4ID: Arquitectura Orientada a Servicios aplicada a sistemas emergentes

Área Ingeniería en Computadores
Licenciatura Ingeniería en Computadores

1 Datos generales

Nombre del curso:	SOA4ID: Arquitectura orientada a servicios aplicada a sistemas emergentes
Código:	CE5508
Tipo de curso:	Teórico – Práctico
Electivo o no:	Si
Nº de créditos:	4
Nº horas de clase por semana:	4
Nº horas extraclase por semana:	12
Áreas curriculares	Ciencias de Ingeniería (40%) Diseño en Ingeniería (60%)
Ubicación en el plan de estudios:	Curso de 8 ^{avo} o 9 ^{eno} semestre de la carrera de Ingeniería en Computadores
Requisitos:	CE-3103: Bases de Datos
Co-requisitos:	No
El curso es requisito de:	No
Asistencia:	Obligatorio
Suficiencia:	No

Posibilidad de reconocimiento: Sí

Vigencia del programa: II Semestre 2021

2 Descripción general

Este curso comprende una introducción a los conceptos básicos para modelar, diseñar y crear soluciones orientadas a sistemas emergentes. Le da al estudiante una visión general para poder seleccionar y utilizar lenguajes de programación, y tecnologías más avanzadas para resolver problemas computacionales

Se espera que el estudiante haya desarrollado las habilidades y conocimientos necesarios en el dominio de las arquitecturas de computadores, de cursos predecesores como Arquitectura de Computadores I (CE-4301), Taller de Diseño Digital (CE-3201), Especificación y Diseño de software (CE-4101). De igual manera, este curso toma como base los conocimientos adquiridos en el área de bases de datos, provenientes del curso Bases de Datos (CE-3101). Los conocimientos adquiridos en este curso servirán de gran aporte para el desarrollo de los cursos de Proyecto de Diseño de Ingeniería en Computadores (CE-5302) y Trabajo Final de Graduación (CE5600).

Adicionalmente se estimula la participación en clase y el trabajo en grupo para la presentación de propuestas de solución a problemas planteados durante la clase como fuera de ella. El desarrollo de estas habilidades permitirá un mejor desempeño del estudiante en proyectos de cualquier tipo. En caso de ser requerido se brindará atención de necesidades educativas especiales en cumplimiento con la ley 7600. Los atributos que se ejercitan a lo largo de las diferentes actividades de este y que se proponen que el estudiante pueda alcanzar al final de curso son:

- Análisis de Problemas (AP)
- Uso de herramientas de ingeniería (HI)
- Trabajo Individual y de Equipo (TE)

3 Objetivos

General:

Al finalizar el curso el estudiante deberá ser capaz de construir soluciones tecnológicas completas con base en componentes avanzados de un sistema computacional, aplicando conceptos y modelos de seguridad para proteger toda la solución.

Objetivos específicos	Atributos correspondientes
Emplear las herramientas de programación más utilizadas en la implementación de sistemas que permitan demostrar la utilidad de las tecnologías involucradas en el curso.	Análisis de Problemas (AP).
Diseñar y crear sistemas tomando como base tecnologías emergentes que permitan ofrecer las soluciones más adecuadas a cada problema particular	Análisis de Problemas (AP).
Elaborar normas grupales para el desarrollo de proyectos de programación en equipo, así como documentación técnica, de manera colaborativa, ordenada y concisa, haciendo uso de principios y valores como responsabilidad, respeto y tolerancia.	Trabajo Individual y en Equipo
Aplicar los principales elementos semánticos y sintácticos de un modelo de programación concreto para el diseño de soluciones con un fin determinado	Uso de Herramientas de Ingeniería (HI)

4 Contenidos

→ **SERVICIOS WEB (3 SEMANAS)**

- ◆ Características de los Servicios Web
- ◆ Tipos de servicios
- ◆ Protocolos de red
- ◆ SOAP, WSDL, RESTful
- ◆ JSON/XML
- ◆ Implementación de servicios web
- ◆ Validación y pruebas en servicios

→ **CLOUD COMPUTING (2 SEMANAS)**

- ◆ Conceptos básicos y terminologías de Cloud Computing
- ◆ Tecnologías Cloud-enabling
- ◆ Virtualización y Contenedores
- ◆ Mecanismos de administración de soluciones en el cloud:
Balanceo de carga, escalabilidad, elasticidad, replicación,
monitoreo, SLA
- ◆ Conceptos de CI/CD
- ◆ Almacenamiento en la nube

→ **INTRODUCCIÓN A LA ARQUITECTURA DE SOFTWARE (1.5 SEMANAS)**

- ◆ Definición y responsabilidades de la Arquitectura de software
- ◆ Herramientas para el modelado de la arquitectura de software
- ◆ Atributos de calidad como Rendimiento, Seguridad, Verificación, Interoperabilidad, Usabilidad, etc en el diseño de arquitectura.
- ◆ Modelos de documentación de la arquitectura de software

→ **ESTILOS DE ARQUITECTURAS COMUNES (2 SEMANAS)**

- ◆ Arquitecturas en capas
- ◆ Arquitecturas de pipeline
- ◆ Arquitecturas de servicios
- ◆ Arquitecturas basadas en eventos
- ◆ Arquitecturas basadas en espacio

- ◆ Arquitecturas orientadas a servicios orquestados
- ◆ Arquitecturas de Microservicios

→ **ARQUITECTURAS ORIENTADAS A SERVICIOS (2 SEMANAS)**

- ◆ Pasos para la implementación de SOA
- ◆ Arquitecturas REST para SOA
- ◆ Patrones para la implementación de SOA
- ◆ Integración y orquestación de Servicios
- ◆ CI/CD en arquitecturas de servicios

→ **ARQUITECTURAS DE MICROSERVICIOS (3 SEMANAS)**

- ◆ Domain-driven design (DDD)
- ◆ Pasos para la implementación de arquitectura microservicios
- ◆ Patrones para la implementación de arquitectura de microservicios
- ◆ Despliegue y Mantenimiento de soluciones con microservicios
- ◆ Gobernabilidad de Microservicios

→ **MODELO DE SEGURIDAD (1 SEMANAS)**

- ◆ Modelo de seguridad.
- ◆ Ataque por escalamiento de servicios
- ◆ Mecanismos de seguridad para servicios web
- ◆ Análisis de tráfico

→ **APLICACIONES DE SOA EN SISTEMAS EMERGENTES (1 SEMANAS)**

- ◆ Aplicaciones móviles
- ◆ Integración de interfaces de recolección de datos
- ◆ Reconocimiento facial y voz
- ◆ Análisis de sentimiento en texto
- ◆ Extractor de texto en imágenes
- ◆ Detección de objetos en imágenes
- ◆ Reconocimiento de atributos faciales

5 Metodología de enseñanza y aprendizaje

Se emplearán técnicas de clases magistrales por parte del profesor, en donde se desarrollarán los aspectos teóricos y prácticos más relevantes de los diferentes temas. Además, se combinarán con una alta participación por parte de los estudiantes durante el transcurso de las lecciones, por medio de llamadas orales, respuestas a casos en la pizarra y de trabajos individuales y en grupo.

Se presupone que el alumno profundiza los temas abordados en la clase en las lecturas recomendadas por el profesor y que el estudiante será responsable de desarrollar los diferentes entregables que se asignen en el curso.

El profesor asumirá el papel de facilitador y el estudiante tendrá la mayor responsabilidad de su progreso.

6 Evaluación

Pruebas cortas	10%
Investigaciones / Exposiciones	15%
Laboratorios y Casos de Estudio	15%
Proyectos programados	60%
TOTAL	100%

La evaluación del curso presenta un balance entre el trabajo individual y en equipo de los estudiantes.

Las pruebas cortas se efectuarán en el momento de la clase que el profesor considere más apropiado. Por ningún motivo se realizarán reposiciones de pruebas cortas.

Se asignarán tareas extraclase que pueden estar relacionadas con análisis, diseño, programación, investigación y presentaciones orales. Se establecerán las reglas para cada una cuando se asignen. Estas tareas extraclase deberán ser desarrolladas por los estudiantes fuera del horario normal de clases.

Para el cálculo del porcentaje de pruebas cortas y tareas extraclase, se corresponderá a calcular la proporción de todas las calificaciones obtenidas, donde cada entregable tendrá un peso idéntico para el cálculo de dicho rubro. Para los proyectos programados no necesariamente aplicará la misma regla. En el caso de las tareas extraclase, algunas podrían dividirse en pequeñas tareas que al final sumarán el valor de una tarea extraclase. Cuando esto suceda se avisará con anticipación a los estudiantes.

Los proyectos programados se podrán realizar en grupos. En la especificación de cada proyecto se establecerán estos lineamientos. Las evaluaciones programadas deberán tener una adecuada documentación y los resultados deberán ser presentados al profesor en una defensa preparada adecuadamente. La defensa es obligatoria para la revisión del proyecto programado, de no ser así, no se procederá a calificar el mismo. Las normas para la documentación se entregarán oportunamente para cada proyecto.

En caso de que se detecte un plagio o intento de fraude en cualquier trabajo, asignación o evaluación por parte de un estudiante, se procederá a anular el mismo y se enviará una carta al expediente del estudiante.

El orden, precisión, concisión, ortografía y redacción será evaluado en todos los documentos presentados. El profesor está en la libertad de revisar o no los documentos que no cumplan con dicho rubro.

7 Bibliografía

Software Architecture: Foundations, Theory and Practice

By Richard N. Taylor, Nenad Medvidovic.

Software Systems Architecture : Working with Stakeholders using Viewpoints and Perspectives. Second Edition

By Nick Rozanski

Documenting Software Architectures: Views and Beyond, Second Edition.

By Paul Clements.

Clean Architecture: A Craftsman's Guide to Software Structure and Design

By Robert C. Martin

Cloud Computing Design Patterns

By Robert Cope

8 Cuerpo docente

Ing. Isaac Ramírez, M.SI.

Egresado del instituto tecnológico de Costa Rica

Ha impartido el curso de Algoritmos y Estructuras de Datos I y II y Bases de Datos en el Área Académica Ingeniería en Computadores desde el 2014. Desde el 2008 ha ejercido en el área de computación en distintos roles como Programador, Ingeniero de Software y Supervisor de Desarrollo de Software.

Actualmente se desempeña como Ingeniero de Software en Microsoft.

Contacto

asistente.isaac.itcr@gmail.com

Teléfono: 83237466 (WhatsApp, Telegram)

Oficina: Edificio F2, Oficina 21

Consultas

Horario y lugar de consulta: a convenir con cada profesor.

Tutorías

Horario y lugar de tutorías se indicará una vez iniciado el curso lectivo debido a que debe ser definido por la institución.

9 Instrucciones generales

Google Classroom

Google classroom será la herramienta oficial para postear tareas, materiales, lecturas, entre otros. De igual forma, es la vía oficial para hacer entregas de las asignaciones. **Deberá crear una cuenta con su nombre completo y carné y vincularla con classroom.**

<https://classroom.google.com/c/MzcxOTE3NzQyNzU1?cjc=teall4m>

Teams

Utilizaremos teams para las video llamadas. Este será el enlace para las video llamadas:

https://teams.microsoft.com/l/meetup-join/19%3ameeting_ZTQyMWJkZDQtZGNiMC00NTliLWFmMDAtNzI2YjM1NWU5Mjhk%40thread.v2/0?context=%7b%22Tid%22%3a%22bfcf1d9d-93ea-43b1-b902-1daa68a64248%22%2c%220id%22%3a%229f6894d2-557a-47a5-a5c5-b3f2997d58ff%22%7d

Enlace para ingresar sus datos

Debe llenar el siguiente formulario con sus datos personales:
<https://forms.gle/qSNPtwa6jk6TWB8q6>

Medios de comunicación

Cada profesor establecerá los medios de comunicación con los estudiantes y los medios para hacer entrega de las evaluaciones y para la entrega de materiales.

Unirse al siguiente grupo de Telegram:
<https://t.me/joinchat/Vpa7W6WZbqU1MTdh>

Entregables

Cualquier documento que sea solicitado como entregable debe ser en formato PDF. En caso de ser necesario otro formato será indicado por el profesor.

Cada proyecto establece las reglas de entrega.

Sobre el proceso de acreditación de la carrera

Dado que la carrera Ingeniería en Computadores está en proceso de reacreditación, el profesor podrá dejarse algunas copias de los entregables y quices.

Otros

Está terminantemente prohibido realizar grabaciones de las lecciones sin autorización del profesor.

Cronograma del Curso

Puede acceder el cronograma del curso en este enlace:
https://docs.google.com/spreadsheets/d/1_ZnIL2rxNGj6318N0NfwH7jb56AflxC3DZ1vd2nFqF4/edit?usp=sharing