



**1. Datos Generales de la asignatura**

<b>Nombre de la asignatura:</b>	<b>Arquitectura y Diseño de Software</b>
<b>Clave de la asignatura:</b>	<b>ISC-2102</b>
<b>SATCA<sup>1</sup>:</b>	<b>2-2-4</b>
<b>Carrera:</b>	<b>Ingeniería en Sistemas Computacionales.</b>

**2. Presentación**

<b>Caracterización de la asignatura</b>
<p>La asignatura Arquitectura y Diseño de Software se encuentra en el VIII semestre y forma parte de la especialidad Ingeniería de software, junto con el grupo de asignaturas: Programación de Móviles; Ingeniería de Requerimientos; Verificación y Validación; Modelo de Desarrollo Integral (CMMI) y Proceso Personal para el Desarrollo de Software.</p> <p>El desarrollo de la competencia de la asignatura por parte del estudiante aporta al perfil del egresado al integrar las diferentes arquitecturas de software, establecer los lenguajes y estilo correspondientes para el desarrollo de este, para incrementar la productividad en las organizaciones, así como utilizar tecnologías y herramientas actuales y emergentes acordes a las necesidades del entorno en el ámbito de desarrollo de software. En esta asignatura se tiene la intención de formar profesionistas capaces de planear e implementar herramientas que permitan analizar el comportamiento de una aplicación y buscar el máximo desempeño, incrementando la utilidad y calidad en desarrollos.</p> <p>Esta asignatura aporta al perfil del ingeniero en tecnologías de la información y comunicaciones la capacidad de diseñar, desarrollar e integrar tecnologías actuales de arquitectura de software aplicando un modelo estratégico de seguridad integrada al mismo.</p> <p>En el contexto de la ingeniería de software el desarrollo de la arquitectura tiene que ver con la estructuración de un sistema para satisfacer los requerimientos de clientes y otros involucrados, en especial los requerimientos de atributos de calidad.</p> <p>En esta materia se abordan aspectos relacionados con un desarrollo de la arquitectura con un énfasis importante hacia las bases teóricas, pero también realizando ejercicios prácticos que permiten relacionar la teoría con la realidad.</p>
<b>Intención didáctica</b>
<p>Este programa de estudios es teórico-práctico, es decir, el docente incursionara en la teoría que fundamentara la parte aplicativa de los sistemas computacionales y propondrá la ejecución de casos de estudio aplicado a las organizaciones integrando tecnologías y sistemas propios de la ingeniería de software, con un enfoque dirigido a mejorar e innovar los procesos para proporcionar servicios diferenciados; además</p>

<sup>1</sup> Sistema de Asignación y Transferencia de Créditos Académicos



de propiciar la creación de valor para el cliente y como resultado logre ventajas competitivas sustanciales y sostenibles

El papel del docente tendrá una intervención mediadora entre los contenidos disciplinarios de la asignatura y las herramientas utilizadas para facilitar el aprendizaje significativo, autónomo y colaborativo, por lo que se requiere que el perfil del docente que imparta esta asignatura cuente con: conocimientos de administración de software; manejo de lenguajes de programación; conocimiento y aplicación de lenguajes de modelado de software; conocimientos sobre el análisis y diseño de sistemas de información.

La asignatura cubre la necesidad que tiene un ingeniero al enfrentarse a la implementación de estándares de modelado, así como el uso de herramientas CASE en el proceso.

Se organiza en seis unidades; la primera unidad, hace referencia a la introducción de las diferentes técnicas del modelado. La unidad dos, presenta las técnicas y modelado del diseño. En la unidad tres, introduce al desarrollo de software basado en arquitecturas. La unidad cuatro ostenta las notaciones de la documentación y evaluación de arquitecturas de software. En la unidad cinco, conduce a conocer las nuevas propuestas arquitectónicas.

El estudiante es el responsable de su aprendizaje y de la adquisición de competencias y se encuentra en constante evaluación para determinar sus fortalezas, por ello la evaluación es formativa centrada en el proceso de mejora continua a través del uso de instrumentos de evaluación basado en indicadores, criterios y descriptores de logro.

### 3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Instituto Tecnológico Superior de Comalcalco., Julio de 2021	Lic. María Teresa Santiago Lerma	Análisis y Enriquecimiento de los programas por competencias.

### 4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
Aplica diseño, patrones y estilos arquitectónicos para la construcción de software.

### 5. Competencias previas

*Competencias específicas:*

- Comprende y describe los conceptos principales del paradigma de programación orientada a objetos para modelar situaciones reales.
- Conoce, comprende y aplica las estructuras de datos, métodos de ordenamiento y búsqueda para la optimización del rendimiento de soluciones de problemas del contexto.

*Competencias genéricas:*

- Capacidad de abstracción en el modelado de problemas.
- Capacidad de análisis y síntesis.
- Capacidad de organizar y planificar.
- Habilidad para buscar y analizar información proveniente de fuentes diversas.
- Solución de problemas.
- Toma de decisiones.
- Trabajo en equipo.
- Capacidad de aplicar los conocimientos.
- Habilidades de investigación.
- Capacidad de generar nuevas ideas.

### 6. Temario

No.	Temas	Subtemas
1	Fundamentación teórica	1.1 Qué es Análisis y Diseño Orientado a Objetos 1.2 Referencias históricas. 1.2.1 OMT (Técnica de Modelado de Objetos). 1.2.2 Metodología Booch. 1.3 Metodología RUP (Rational Unified Process). 1.4 Diseño de Alto Nivel (HLD) y Bajo Nivel (LLD).



		<ul style="list-style-type: none"> <li>1.5 Comprensión de los requerimientos.</li> <li>1.6 Casos de uso.               <ul style="list-style-type: none"> <li>1.6.1 Introducción.</li> <li>1.6.2 Elementos de casos de uso (diagrama, Relaciones, Especificaciones, Identificación de Casos de Uso).</li> </ul> </li> <li>1.7 Modelo del Dominio.               <ul style="list-style-type: none"> <li>1.7.1 Visualización de conceptos.</li> <li>1.7.2 Añadir asociaciones.</li> <li>1.7.3 Añadir atributos.</li> </ul> </li> </ul>
2	Análisis y Diseño	<ul style="list-style-type: none"> <li>2.1 Representación de los Eventos del Sistema usando Diagramas de Secuencia del Sistema.</li> <li>2.2 Contratos de las operaciones.</li> <li>2.3 Diseño basado en responsabilidades.</li> <li>2.4 Modelo de diseño.</li> <li>2.6 Modelo de comportamiento.</li> <li>2.7 Diagramas de diseño.               <ul style="list-style-type: none"> <li>2.7.1 Diagramas de secuencia.</li> <li>2.7.2 Diagramas de colaboración.</li> <li>2.7.3 Diagrama de clases de diseño.</li> </ul> </li> <li>2.8 Patrones Grasp.</li> <li>2.9 Patrones GoF.</li> </ul>
3	Diseño de Arquitecturas de software.	<ul style="list-style-type: none"> <li>3.1 Desarrollo de Software Basado en Arquitecturas               <ul style="list-style-type: none"> <li>3.1.1 Historia</li> </ul> </li> <li>3.2 Atributos Funcionales</li> <li>3.3 Atributos de Calidad</li> <li>3.4 Proceso de elaboración de Arquitecturas de Software               <ul style="list-style-type: none"> <li>3.4.1 Forward Engineering</li> <li>3.4.2 Reverse Engineering</li> <li>3.4.3 Estilos</li> <li>3.4.4 Patrones</li> </ul> </li> </ul>
4	Documentación y métodos de evaluación de Arquitecturas de Software	<ul style="list-style-type: none"> <li>4.1 Documentación de Arquitecturas de Software</li> <li>4.2 Modelo 4+1</li> <li>4.3 ATAM</li> <li>4.4 SAAM</li> <li>4.5 ARID</li> <li>4.6 ALMA</li> <li>4.7 SNA</li> </ul>
5	Tendencias	<ul style="list-style-type: none"> <li>5.1 Líneas de Productos de Software.</li> <li>5.2 Arquitecturas Orientadas a Servicios.</li> <li>5.3 Arquitecto de Software</li> </ul>

## 7. Actividades de aprendizaje de los temas

1. Fundamentación teórica	
Competencias	Actividades de aprendizaje
<p><i>Específica:</i></p> <ul style="list-style-type: none"> <li>Identifica y comprende metodologías y técnicas aplicables al diseño para realizar los componentes del proceso de un sistema concreto.</li> </ul> <p><i>Genéricas:</i></p> <ul style="list-style-type: none"> <li>Capacidad de análisis y síntesis.</li> <li>Capacidad de organizar y planificar.</li> <li>Comunicación oral y escrita.</li> <li>Solución de problemas.</li> <li>Toma de decisiones.</li> <li>Habilidad para buscar y analizar información proveniente de fuentes diversas.</li> </ul>	<ul style="list-style-type: none"> <li>Comunicar y presentar la asignatura: encuadre, empleo de recursos educativos abiertos (REA), entrega de productos y evidencias de aprendizaje en la plataforma, netiqueta y tiempo de acompañamiento docente.</li> <li>Organizar al grupo en equipos para investigar y discutir las fases que implica el modelado del negocio. Elaborar el informe de investigación y subirlo a plataforma en formato pdf.</li> <li>Organizar al grupo para realizar las siguientes tareas: <ul style="list-style-type: none"> <li>_Aplicar el Lenguaje Unificado.</li> <li>_Modelado (UML) realizando prácticas sobre el modelado del negocio para un caso concreto.</li> <li>_Documentar los requerimientos denotados por UML para un sistema concreto.</li> </ul> </li> <li>Organizar al grupo para realizar la práctica de realizar todos los componentes del proceso que involucra un caso de uso para el sistema considerado en el punto anterior.</li> <li>Retroalimentación de los temas por parte de los estudiantes y el facilitador. Valoración de la práctica mediante una lista de cotejo.</li> </ul>
2. Análisis y diseño	
Competencias	Actividades de aprendizaje
<p><i>Específica:</i></p> <ul style="list-style-type: none"> <li>Identifica y comprende metodologías y técnicas aplicables al diseño para realizar los componentes del proceso de un sistema concreto.</li> </ul> <p><i>Genéricas:</i></p> <ul style="list-style-type: none"> <li>Capacidad de análisis y síntesis.</li> <li>Capacidad de organizar y planificar.</li> <li>Comunicación oral y escrita.</li> <li>Solución de problemas.</li> <li>Toma de decisiones.</li> <li>Habilidad para buscar y analizar información proveniente de fuentes diversas.</li> </ul>	<ul style="list-style-type: none"> <li>Organizar al grupo en equipos para realizar la representación de los eventos del sistema, así como los contratos de las operaciones.</li> <li>Retroalimentar y valorar la utilización de los patrones de diseño GRASP y GoF, realizar los modelos correspondientes al sistema.</li> <li>Elaborar los diagramas denotados por la metodología que correspondan a los modelos anteriores. Evaluar el avance alcanzado por los equipos.</li> <li>Organizar los tiempos para la entrega de los avances del proyecto y</li> </ul>

	retroalimentar a los equipos.
<b>3. Diseño de Arquitecturas de software</b>	
<b>Competencias</b>	<b>Actividades de aprendizaje</b>
<p><b>Específica(s):</b></p> <ul style="list-style-type: none"> <li>Identifica los conceptos fundamentales de arquitectura de software y su relevancia para realizar el diseño de arquitectura del sistema que se está realizando.</li> </ul> <p><b>Genéricas:</b></p> <ul style="list-style-type: none"> <li>Capacidad de análisis y síntesis.</li> <li>Capacidad de organizar y planificar.</li> <li>Comunicación oral y escrita.</li> <li>Solución de problemas.</li> <li>Toma de decisiones.</li> <li>Habilidad para buscar y analizar información proveniente de fuentes diversas.</li> </ul>	<ul style="list-style-type: none"> <li>Realizar un ejercicio que muestre la aplicación del concepto de desarrollo de software basado en arquitecturas.</li> <li>Organizar al grupo en equipos para buscar, seleccionar y evaluar información de atributos funcionales del negocio, de usuario, de sistema, entre otros, presentar el informe de investigación y discutirlo en plenaria.</li> <li>Organizar al grupo en equipos para buscar, seleccionar y evaluar información de atributos de calidad como: desempeño, confiabilidad, seguridad, facilidad de modificación, facilidad de uso, robustez, portabilidad, escalabilidad, reutilización, disponibilidad, etcétera. Presentar el informe de investigación y discutirlo en plenaria.</li> <li>Realizar el diseño de la arquitectura del sistema que se está realizando, realizar ejercicios que permitan modelar la arquitectura de sistemas existentes.</li> <li>Evaluar los avances del proyecto y retroalimentar por parte de los compañeros y el facilitador en plenaria.</li> </ul>
<b>4. Documentación y métodos de evaluación de Arquitecturas de Software</b>	
<b>Competencias</b>	<b>Actividades de aprendizaje</b>
<p><b>Específica(s):</b></p> <ul style="list-style-type: none"> <li>Documenta y evalúa una arquitectura de software para aplicar uno de los métodos al diseño de la arquitectura del sistema.</li> </ul> <p><b>Genéricas:</b></p> <ul style="list-style-type: none"> <li>Capacidad de análisis y síntesis.</li> <li>Capacidad de organizar y planificar.</li> <li>Comunicación oral y escrita.</li> <li>Solución de problemas.</li> <li>Toma de decisiones.</li> <li>Habilidad para buscar y analizar información proveniente de fuentes diversas.</li> <li>Capacidad de generar nuevas ideas (creatividad).</li> </ul>	<ul style="list-style-type: none"> <li>Organizar al grupo en equipo para buscar, seleccionar y evaluar información de documentación de arquitecturas de Software. Subir el informe de investigación en formato pdf a la plataforma para su valoración.</li> <li>Realizar ejercicios con el lenguaje UML.</li> <li>Analizar en equipos, el Modelo 4+1 e identificar sus vistas y aplicaciones. Valorar la actividad con la rúbrica de discusión en equipos.</li> <li>Aplicar el método de evaluación ATAM a un ejemplode diseño de arquitectura</li> <li>Aplicar el método de evaluación SAAM a un ejemplode diseño de arquitectura</li> <li>Aplicar el método de evaluación ARID</li> </ul>

	<p>a un ejemplode diseño de arquitectura</p> <ul style="list-style-type: none"> <li>• Aplicar el método de evaluación ALMA a un ejemplo de diseño de arquitectura.</li> <li>• Aplicar el método de evaluación SNA a un ejemplo de diseño de arquitectura.</li> <li>• Realizar una comparación de los diferentes métodos de evaluación y aplicar uno de los métodos al diseño de la arquitectura del sistema.</li> </ul>
<b>5. Tendencias</b>	
Competencias	Actividades de aprendizaje
<p><i>Específica:</i></p> <ul style="list-style-type: none"> <li>• Identifica nuevos paradigmas y perspectivas de desarrollo de software para implementar servicios Web que conformen una Arquitectura Orientada a Servicios.</li> </ul> <p><i>Genéricas:</i></p> <ul style="list-style-type: none"> <li>• Capacidad de análisis y síntesis.</li> <li>• Capacidad de organizar y planificar</li> <li>• Solución de problemas.</li> <li>• Toma de decisiones</li> <li>• Trabajo en equipo.</li> <li>• Capacidad de aplicar los conocimientos en la práctica.</li> <li>• Capacidad de generar nuevas ideas (creatividad).</li> </ul>	<ul style="list-style-type: none"> <li>• Analizar la aplicación del paradigma de Líneas de Productos de Software para conocer sus ventajas y beneficios.</li> <li>• Implementar Servicios Web que conformen unaArquitectura Orientada a Servicios.</li> <li>• Realizar un análisis para identificar las capacidadesdel ingeniero de software.</li> </ul>

## 8. Práctica(s)

<ul style="list-style-type: none"> <li>• Dado un sistema concreto real o ficticio realizar representación del modelado del negocio siguiendo las especificaciones denotadas por UML para este proceso. Realizar el análisis de requerimientos del sistema dado, contemplando: recolección, representación y validación de los datos.</li> <li>• Representar los elementos que conforman un caso de uso para el sistema especificado.</li> <li>• Realizar el análisis y diseño para el sistema determinado siguiendo los patrones especificados por la notación UML.</li> <li>• Desarrollo de un proyecto dosificado durante el semestre, involucrando la elaboración, evaluación y documentación de una arquitectura de software, donde se aplique al menos uno de los métodos de evaluación y al menos dosde las vistas del modelo 4+1 utilizando el Lenguaje Unificado de Modelado.</li> </ul>
--



## 9. Proyecto de asignatura

**Proyecto:** Elaboración, evaluación y documentación de una arquitectura de software, donde se aplique al menos uno de los métodos de evaluación y al menos dos de las vistas del modelo 4+1 utilizando el Lenguaje Unificado de Modelado.

**Objetivo:** Aplicar el diseño de una arquitectura de software para emplear al menos uno de los métodos de evaluación utilizando el Lenguaje Unificado de Modelado.

### Fundamentación:

El plan de pruebas de software se elabora para atender los objetivos de calidad en un desarrollo de sistemas, encargándose de definir aspectos como por ejemplo los módulos o funcionalidades sujeto de verificación, tipos de pruebas, entornos, recursos asignados, entre otros aspectos. Para su desarrollo se requiere el desarrollo de las habilidades genéricas como son liderazgo, herramientas de investigación y el uso de la tecnología para verificar y validar software.

### Planeación:

Se recomienda un plan de pruebas de software, considerando los siguientes pasos:

- 1.- Analizar los requerimientos de desarrollo de software
- 2.- Identificar las funcionalidades nuevas a probar
- 3.- Identificar las funcionalidades de sistemas existentes que deben probarse
- 4.- Definir la estrategia de pruebas
- 5.- Definir los criterios de inicio, aceptación y suspensión de pruebas
- 6.- Identificar los entornos (ambientes) requeridos.
- 7.- Determinar necesidades de personal y entrenamiento
- 8.- Establecer la metodología y procedimientos de prueba

### Ejecución:

Consiste en el desarrollo plan de pruebas de software, realizada por parte de los estudiantes con asesoría del docente, se inicia con la solicitud de Prueba", que se da inicio al proceso de pruebas definido. Es responsabilidad del líder de proyecto elaborar esta solicitud facilitando toda la documentación que se requiere, los requisitos asociados, los ejecutables de la aplicación o módulo, los manuales de instalación, configuración y usuario.

Continúa con "Crear Plan de Prueba"- El cual tiene el objetivo fundamental de establecer la planificación en general del proceso de pruebas en el laboratorio de calidad. En este momento se registran los datos iniciales del plan, se asigna el miembro del equipo que asumirá el rol de Especialista de Pruebas.

Por último, ejecución del plan de prueba el responsable de su ejecución en este caso es el probador, o los probadores que integren el equipo de pruebas. La primera actividad que se debe realizar es la definición de las opciones de ejecución del caso de prueba, que establecen cuál será el entorno en el que se ejecuten y las configuraciones. Del segundo juego de datos en adelante se reproduce la grabación y se evalúan todas las combinaciones restantes. Si se detecta algún error durante la realización de las comprobaciones se debe entonces generar una No Conformidad (NC) y son estas precisamente el principal artefacto que se obtiene de este subproceso.

### Evaluación:

El plan de prueba de desempeño debe aplicar al código fuente de software, considerando los siguientes elementos.

- Métodos.
- Atributos.
- Variables. Conexión a la base de datos.



- Componentes.
  - Excepciones.
- Las pruebas unitarias se deben aplicar:
- Diferentes escenarios de pruebas.
  - Criterios de aceptación.
  - Resultados de las pruebas.

## 10. Evaluación por competencias

### Evaluación formativa:

Producto o Evidencia de aprendizaje	Instrumento de evaluación
Reporte de investigación documental	<p>Indicadores de evaluación de un reporte de investigación:</p> <ul style="list-style-type: none"> <li>– Tiene carátula con datos de la institución, nombre del alumno, materia, nombre del autor, título del libro y capítulo, nombre del docente, grupo y fecha.</li> <li>– El documento contiene índice en el que se han incluido todos los subtemas.</li> <li>– El contenido se encuentra en arial 12, justificados y espacio y medio; los subtítulos en arial 14, centrados y a espacio y medio.</li> <li>– El documento contiene introducción, apartados y conclusión.</li> <li>– El documento emplea citas bibliográficas y referencias ambas emplean las normativas de APA.</li> <li>– El documento se subió a la plataforma en tiene tiempo y formato pdf.</li> </ul>
Rúbrica de discusión de temas en equipo.	<p>Rúbrica de discusión de temas:</p> <ul style="list-style-type: none"> <li>– Expongo y argumento mis ideas perfectamente, valorando y respetando las intervenciones y opiniones de mis compañeros y compañeras.</li> <li>– Utilizo sin dificultad las tecnologías de la información y la comunicación para obtener información.</li> <li>– Asumo mi rol sin interferir en el trabajo de los demás y apporto ideas al grupo.</li> </ul>
Aplicación de las prácticas.	<p>Lista de cotejo de reporte de practica:</p> <ul style="list-style-type: none"> <li>– Presenta los elementos indispensables: nombre del equipo, institución, nombre de la práctica, datos generales, nombre del curso, nombre del profesor y fecha.</li> <li>– Se evalúan las características del modelo de arquitectura de software.</li> <li>– Se evalúan críticamente los métodos de evaluación de arquitectura de software.</li> </ul>

	<ul style="list-style-type: none"> <li>- Se observa el procedimiento realizado en cada práctica.</li> <li>- Se subió el reporte de lectura a la plataforma para su evaluación.</li> </ul>
<p>Proyecto: Elaboración, evaluación y documentación de una arquitectura de software, donde se aplique al menos uno de los métodos de evaluación y al menos dos de las vistas del modelo 4+1 utilizando el Lenguaje Unificado de Modelado.</p>	<p>Lista de cotejo de</p> <ol style="list-style-type: none"> <li>1.- Analizar los requerimientos de desarrollo de arquitectura de software</li> <li>2.- Seleccionar uno de los métodos de evaluación</li> <li>3.- Elaborar la documentación.</li> <li>4.- Definir la estrategia de pruebas</li> <li>5.- Definir los criterios de inicio, aceptación y suspensión de pruebas</li> <li>6.- Identificar los entornos (ambientes) requeridos.</li> <li>7.- Determinar necesidades de personal y entrenamiento.</li> <li>8.- Establecer la metodología y procedimientos de prueba.</li> </ol>
<p><b>Evaluación sumativa:</b></p> <ol style="list-style-type: none"> <li>1. Participación en clases</li> <li>2. Evidencias de productos y desempeños en la plataforma</li> <li>3. Portafolio de aprendizaje personal</li> <li>4. Evaluación del proyecto</li> </ol>	

## 11. Fuentes de información

### Referencias

- Amórtegui Vargas, M. P., Zorro Jiménez, L. C., Loaiza Carvajal, V. C., González Rivera, R. A., & Torres Moreno, M. E. (2017). *Ingeniería de requerimientos*. Editorial Pontificia Universidad Javeriana.
- Baños Delgado, I. A. (2017). Metodología para evaluar la calidad de un producto software de una implementación de internet de las cosas. <https://repositorio.utb.edu.co/handle/20.500.12585/381>.
- Cabezas Rivera, K., & Mafla Sandoval, M. (2018). Sistema de gestión académica para la asociación cesar conto. [https://repository.unicatolica.edu.co/bitstream/handle/20.500.12237/993/SISTEMA\\_GESTION\\_ACADEMICA\\_PARA\\_ASOCIACION\\_CESAR\\_CONTO.pdf?sequence=1](https://repository.unicatolica.edu.co/bitstream/handle/20.500.12237/993/SISTEMA_GESTION_ACADEMICA_PARA_ASOCIACION_CESAR_CONTO.pdf?sequence=1).
- Cruz Segura, Y. (2018). *Método basado en ontología para representar decisiones de diseño de software en la actividad productiva de la UCI* (Master's thesis, Universidad de las Ciencias Informáticas).
- Frati, F. E., & Carmona, F. B. (2021). XXIII Edición del Workshop de Investigadores en Ciencias de la Computación. <http://sedici.unlp.edu.ar/handle/10915/119490>.
- Gamboa, J. Z. (2018). Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software. *INNOVA Research Journal*, 3(10), 20-33.
- García-Peñalvo, F. J. (2020). Procesos y Métodos de Modelado para la Ingeniería Web.



- <https://repositorio.grial.eu/handle/grial/1954>.
- Garrido, M., Braun, G., & Roger, S. (2017). Agentes inteligentes y web semántica: hacia la verbalización de un subconjunto de UML en una herramienta gráfica web. In *XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires)*.
- López Ruiz, L. I., & Gómez Aguilar, A. C. Laboratorio digital en arquitectura. <https://repository.usta.edu.co/handle/11634/32519?show=full>.
- Palmero, M. A. S., Martínez, N. S., & Grass, O. Y. R. (2019). Revisión de elementos conceptuales para la representación de las arquitecturas de referencias de software. *Revista Cubana de Ciencias Informáticas*, 13(1), 143-157.
- Peña, C. A. G. (2017). Líneas de Productos Software: Generando Código a Partir de Modelos y Patrones. *Scientia et technica*, 22(2), 175-182.
- Pisco Gomez, A. P., Regalado Jalca, J. J., Gutierrez Garcia, J., Quimis Sanchez, O., Marcillo Parrales, K., & Marcillo Merino, J. (2020). Fundamentos sobre la Gestion de Bases de Datos.
- Ramírez Chaparro, C. V. Diagramas esenciales del lenguaje unificado de modelado para los requisitos agiles en el desarrollo de software. <https://repository.unad.edu.co/bitstream/handle/10596/38052/cvramirezcz.pdf?sequence=1&isAllowed=y>.
- Ramos, D., Noriega, R., Laínez, J. R., & Durango, A. (2017). *Curso de Ingeniería de Software: 2ª Edición*. IT Campus academy.
- Rueda Rueda, J. S. (2018). Framework conceptual de ciberseguridad para aplicaciones de Internet de las cosas. <https://repository.unab.edu.co/handle/20.500.12749/3552>.
- Russo Ibañez, J. P. (2018). Mapeo sistemático y evaluación de arquitecturas de software para contextos de big data. <https://dspace.ort.edu.uy/handle/20.500.11968/3876>.
- Troya, J. M., Vallecillo, A., & Fuentes, L. (2017). Desarrollo de software basado en componentes. <http://biblioteca.udgvirtual.udg.mx/jspui/handle/123456789/1247>.
- Vázquez-Ingelmo, A., & García-Peñalvo, F. J. (2019). Análisis orientado a objetos. <https://repositorio.grial.eu/handle/grial/1563>.