



UTEC  
Universidad  
de Ingeniería  
y Tecnología

**UTEC**  
UNIVERSIDAD DE INGENIERÍA  
Y TECNOLOGÍA



**CIENCIA DE LA  
CE2A**

**COMPUTACIÓN**

# ÍNDICE

|   |          |
|---|----------|
| <b>1. Asignatura</b>                            | <b>3</b> |
| <b>2. Datos generales</b>                       | <b>3</b> |
| <b>3. Profesores</b>                            | <b>3</b> |
| 3.1 Profesor coordinador del curso              | 3        |
| 3.2 Profesor(es) instructor(es) del curso       | 3        |
| <b>4. Introducción al curso</b>                 | <b>3</b> |
| <b>5. Objetivos</b>                             | <b>4</b> |
| <b>6. Competencias</b>                          | <b>4</b> |
| <b>7. Resultados de aprendizaje</b>             | <b>4</b> |
| <b>8. Temas</b>                                 | <b>5</b> |
| <b>9. Plan de trabajo</b>                       | <b>5</b> |
| 9.1 Metodología                                 | 5        |
| 9.2 Sesiones de teoría                          | 5        |
| 9.3 Sesiones de práctica (laboratorio o taller) | 5        |
| <b>10. Sistema de evaluación</b>                | <b>7</b> |
| <b>11. Sesiones de apoyo o tutorías</b>         | <b>8</b> |
| <b>12. Referencias Bibliográficas</b>           | <b>8</b> |

# UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

## SILABO 2020-2

### 1. ASIGNATURA

CS2201 – Arquitectura de Computadores

### 2. DATOS GENERALES

**2.1 Créditos:** Tres (3) créditos

**2.2 Horas de teoría:** Dos (2) semanales

**2.3 Horas de práctica:** Dos (2) semanales

**2.4 Duración del período:** Dieciséis (16) semanas

**2.5 Condición:**

- Obligatorio para Ciencia de la Computación.

**2.6 Modalidad:** Presencial

**2.7 Requisitos:**

- CS1D02 – Estructuras Discretas II

### 3. PROFESORES

#### 3.1 Profesor coordinador del curso

Jorge Luis González Reaño (jgonzalez@utec.edu.pe)

Horario de atención: Miércoles 10 am, previa coordinación con el profesor.

#### 3.2 Profesor(es) instructor(es) del curso

Jorge Luis González Reaño (jgonzalez@utec.edu.pe)

Horario de atención: Miércoles 10 am, previa coordinación con el profesor.

### 4. INTRODUCCIÓN AL CURSO

El curso introduce los conceptos fundamentales de organización y diseño de procesadores modernos a través de un enfoque teórico-práctico. Son estudiados conceptos fundamentales de la arquitectura del procesador, lo cual permitirá al estudiante entender las limitaciones de un sistema de cómputo. El tener un conocimiento detallado del funcionamiento del sistema desde la perspectiva de *hardware*, permitirá al estudiante realizar implementaciones eficientes de *software*. El curso incluye actividades de implementación de módulos usando lenguajes de descripción de *hardware* los cuales son probados en simulación. También se realiza la implementación de un microprocesador basado en los conceptos teóricos estudiados. Los temas principales desarrollados en el curso son: diseño lógico, *Instruction Set Architecture*, microarquitectura y sistemas de memoria.

## 5. OBJETIVOS

Sesión 1: Definir los conceptos y principios fundamentales de Arquitectura de Computadores. Diferenciar los niveles de abstracción desde dispositivos electrónicos hasta algoritmos. Revisar conceptos de lógica y representación numérica en computadores estudiados en cursos previos.

Sesión 2: Detallar los principios de lógica combinacional, así como el comportamiento de compuertas y otros circuitos relacionados. Relacionar la lógica combinacional al funcionamiento del procesador.

Sesión 3: Detallar los principios de lógica secuencial, así como el comportamiento de latches, flip-flops, registros y circuitos relacionados. Relacionar la lógica secuencial al funcionamiento del procesador.

Sesión 4: Analizar y diseñar máquinas de estado finitas a partir de los conceptos de lógica combinacional y secuencial estudiados en las sesiones anteriores.

Sesión 5: Introducir y estudiar el modelo de Von Neumann presente en procesadores modernos. Detallar y estudiar un *Instruction Set Architecture* (ISA), así como presentar su relevancia en los procesadores.

Sesión 6: Detallar los principios de microarquitectura relacionados al ISA analizado en la sesión previa. Detallar el diseño de un procesador *single-cycle*.

Sesión 7: Extender el diseño de microarquitectura en un procesador *multi-cycle*.

Sesión 8: Evaluar el aprendizaje a través del examen parcial del curso.

Sesión 9: Explicar la técnica de *pipelining* en procesadores. Definir problemas (*hazards*) y soluciones durante la ejecución de programas.

Sesión 10: Analizar y detallar técnicas presentes en procesadores modernos como: renaming, reorder buffer, ejecución fuera de orden, superscalares.

Sesión 11: Presentar el sistema de memoria como una parte fundamental del sistema de cómputo. Explicar los fundamentos básicos, definir la jerarquía y detallar el funcionamiento de *cache*.

Sesión 12: Complementar la visión sobre el sistema de memoria a través de la abstracción de memoria virtual.

Sesión 13: Analizar el funcionamiento de procesadores vectoriales/array y relacionados, tales como las unidades de procesamiento gráfico (GPU).

Sesión 14: Introducir otras arquitecturas de procesamiento actuales, así detallar sus características más relevantes, tales como los *systolic arrays*.

Sesión 15: Detallar la interconexión intra- y on-chip de procesadores actuales.

Sesión 16: Evaluar el aprendizaje a través del examen final del curso.

## 6. COMPETENCIAS

- a2: aplicar conocimientos de ciencias (nivel 2)  
El estudiante resuelve una hoja de problemas donde describe, selecciona y aplica lo aprendido de los fundamentos de arquitectura de computadores.
- b2: analizar información (nivel 3)  
El estudiante analiza y propone el método de desarrollo solicitado, a partir de la lectura de problemas en arquitectura de computadores .
- b3: interpretar información (nivel 3)  
El estudiante evalúa y crea a partir de la interpretación de los problemas propuestos, el diseño más adecuado para la solución de este.
- c1: diseñar un sistema, un componente o un proceso para satisfacer las necesidades deseadas dentro de restricciones realistas (nivel 2).  
El estudiante aplica y analiza los criterios de diseño de un microprocesador de acuerdo a lo planteado en clase.
- d1: trabajar en equipo (nivel 2)  
El estudiante desarrolla soluciones a los problemas propuestos a partir de la discusión de ideas y descripción de *hardware* con los miembros de grupo.

## 7. RESULTADOS DE APRENDIZAJE

Al final del curso el estudiante de Arquitectura de Computadores:

**RA1.** El estudiante será capaz de aplicar los fundamentos sobre arquitectura de computadores; así como de comprender, diseñar y controlar el sistema computacional.

**RA 2.** El estudiante será capaz de identificar problemas en el uso y diseño de procesadores.

**RA 3.** El estudiante será capaz de realizar implementaciones eficientes de *software*, al tener entendimiento detallado del funcionamiento de *hardware*.

**RA 4.** El estudiante será capaz de implementar un microprocesador siguiendo los principios fundamentales de estos impartidos en el curso.

Fecha de actualización: 27/08/2020

Revisado y aprobado por el Centro de Excelencia en Enseñanza y Aprendizaje y la Dirección de Ciencia de la Computación



## 8. TEMAS

1. Introducción: motivación y revisión.
2. Diseño digital: lógica combinacional.
3. Diseño digital: lógica secuencial.
4. Diseño de *Finite State Machines* (FSM).
5. *Instruction Set Architecture* (ISA).
6. Microarquitectura: procesadores *single-cycle*.
7. Microarquitectura: procesadores *multi-cycle*.
8. *Pipelining* en microprocesadores.
  - 8.1 Problemas de ejecución causados por dependencias.
9. Técnicas de microarquitectura: desde *reorder buffer* a ejecución *superscalar*.
10. Sistemas de memoria: jerarquía y cache.
11. Memoria virtual.
12. Procesadores tipo array y vectoriales: GPU
13. Introducción a arquitecturas alternativas: procesadores sistólicos.
14. Introducción a interconexión en procesadores.

## 9. PLAN DE TRABAJO

### 9.1 Metodología

La metodología del curso corresponde al aprendizaje basado en problemas y proyectos. Esta metodología promueve el desarrollo del pensamiento crítico y el interés del estudiante en resolver problemas teóricos y aplicados.

### 9.2 Sesiones de teoría

Las sesiones teóricas serán desarrolladas bajo la estructura de clase magistral. Durante las sesiones teóricas, se promueve la participación activa de los estudiantes, fomentando preguntas y discusiones sobre el estudio de arquitectura de computadores. Después de cada sesión, los estudiantes tendrán una lectura sugerida (indicada por el docente). Este contenido permitirá al estudiante reforzar los conceptos estudiados.

### 9.3 Sesiones de práctica (laboratorio o taller)

Las sesiones prácticas/laboratorio se desarrollarán a través de una metodología basada en problemas generando el aprendizaje práctico por parte del estudiante. Los laboratorios son de tipo individual cuando la actividad requiere el logro de implementar un concepto estudiado en una sesión de teoría.

El proyecto a desarrollar será planteado en la Sesión 11. El desarrollo es grupal y será realizado con la participación de máximo tres estudiantes.

## 10. SISTEMA DE EVALUACIÓN

La parte teórica del curso tiene dos módulos:

- 2 Prácticas calificadas (una por módulo).
- 2 Exámenes (uno por módulo)

La parte práctica y/o laboratorio tiene 2 pilares principales:

- 1 Proyecto.
- 8 Laboratorios.

| EVALUACIÓN   | TEORÍA  | PRÁCTICA Y/O LABORATORIO                       |
|--|---|--|
| *La ponderación de la evaluación se hará si ambas partes están aprobadas | 2 Prácticas Calificadas (20%) (C)<br>2 Exámenes (30%) (E) | 8 Laboratorios (30%)(L)<br>1 Proyecto (20%)(P) |
|  | 50%   | 50%  |
|  | <b>100%</b>   |  |

Se utilizarán las siguientes rúbricas para medir las competencias del curso, y evaluar las actividades más significativas del curso:

Laboratorios: [enlace](#)

Proyecto: [enlace](#)

Los Exámenes (E) y Prácticas Calificadas (C) evaluarán la competencia a2: aplicar conocimientos de ciencias (nivel 2).

## 11. SESIONES DE APOYO O TUTORÍAS

Este apartado permite formalizar los espacios de apoyo a los estudiantes y que éstos tengan la atención NECESARIA y el tiempo disponible para presentar sus dudas y consultas acerca del curso:

| Semana | Fecha/ Hora                     | Tema a tratar      | Objetivos de la sesión   |
|--------|---------------------------------|--------------------|--|
| 1      | 04/09/2020<br>4:00 PM – 5:00 PM | Recursos del curso | - Mostrar dónde se encuentran los recursos y materiales para cada actividad del curso. |
| 3      | 18/09/2020<br>4:00 PM – 5:00 PM | Laboratorio 1      | Clarificar los puntos de la rúbrica de laboratorio N° 1                                |
| 5      | 02/10/2020<br>4:00 PM – 5:00    | Soluciones Casos   | Conocer las soluciones de casos, quizes o lo que estime el                             |

Fecha de actualización: 27/08/2020

Revisado y aprobado por el Centro de Excelencia en Enseñanza y Aprendizaje y la Dirección de Ciencia de la Computación

|    |                                 |                  |   |
|----|---------------------------------|------------------|---|
|    | PM                              |                  | docente conveniente.  |
| 7  | 16/10/2020<br>4:00 PM – 5:00 PM | Examen Parcial   | Responder a las dudas sobre las preguntas de examen final, temas relacionados, contenidos, tipos de ejercicios. |
| 9  | 30/10/2020<br>4:00 PM – 5:00 PM | Soluciones Casos | Conocer las soluciones de casos, quizes o lo que estime el docente conveniente.                                 |
| 11 | 13/11/2020<br>4:00 PM – 5:00 PM | Proyecto         | Clarificar actividades del proyecto   |
| 13 | 27/11/2020<br>4:00 PM – 5:00 PM | Soluciones Casos | Conocer las soluciones de casos, quizes o lo que estime el docente conveniente.                                 |
| 15 | 11/12/2020<br>4:00 PM – 5:00 PM | Examen Final     | Responder a las dudas sobre las preguntas de examen final, temas relacionados, contenidos, tipos de ejercicios. |

## 12. REFERENCIAS BIBLIOGRÁFICAS

### Bibliografía Básica

Harris, S., & Harris, D. (2015). *Digital design and computer architecture: arm edition*. Morgan Kaufmann.

### Bibliografía Complementaria

Patterson, D. A., & Hennessy, J. L. (2016). *Computer Organization and Design ARM Edition: The Hardware Software Interface*. Morgan kaufmann.

LaMeres, B. J. (2019). *Quick Start Guide to Verilog* (pp. 13-22). Springer, Cham.

Readler, B. C. (2011). *Verilog by example: a concise introduction for FPGA design*. Full Arc Press.

Patel, S., & Patt, Y. (2004). *Introduction to Computing Systems: from bits & gates to C & beyond*. McGraw-Hill Professional.