



## I. INFORMACIÓN GENERAL

<b>CURSO</b>	:	Software para Ingeniería
<b>CÓDIGO</b>	:	EL227
<b>CICLO</b>	:	201802
<b>CUERPO ACADÉMICO</b>	:	<b>Cieza Dávila, Javier Eduardo</b> <b>Puerta Arce, Juan Alberto</b>
<b>CRÉDITOS</b>	:	3
<b>SEMANAS</b>	:	16
<b>HORAS</b>	:	1 H (Laboratorio) Semanal /1 H (Práctica) Semanal /2 H (Teoría) Semanal
<b>ÁREA O CARRERA</b>	:	Ingeniería Electrónica

## II. MISIÓN Y VISIÓN DE LA UPC

Misión: Formar líderes íntegros e innovadores con visión global para que transformen el Perú.

Visión: Ser líder en la educación superior por su excelencia académica y su capacidad de innovación.

## III. INTRODUCCIÓN

El curso Software para Ingeniería permite a los alumnos introducirse en el mundo de la programación, cuyo ámbito de aplicación lo tenemos en la matemática, ciencias e ingeniería. Hoy en día existen muchas aplicaciones y equipos electrónicos que son controlados por un software, el cual a su vez está desarrollado en base a algoritmos computacionales. Por lo que aprender a ¿programar¿ juega un rol muy importante en la formación de un ingeniero electrónico y mecatrónico.

El curso Software para Ingeniería ha sido diseñado con el propósito de permitir al futuro ingeniero desarrollar su capacidad de resolver problemas de ingeniería utilizando las técnicas algorítmicas de la programación, desde los primeros ciclos de estudio. El curso contribuye directamente desarrollar la competencia de Pensamiento Crítico y Razonamiento Cuantitativo (general-UPC) y específica de diseño de procesos y sistemas. (ambas nivel 1)

## IV. LOGRO (S) DEL CURSO

Al finalizar el curso el estudiante construye algoritmos computacionales a través de las técnicas de un lenguaje de programación, de manera creativa y razonada.

Competencia 1: Pensamiento Crítico

Nivel de logro: 1

Definición: Capacidad para explorar de manera exhaustiva problemas, idea o eventos para formular conclusiones u opiniones sólidamente justificadas.

Competencia 2: Razonamiento Cuantitativo

Nivel de logro: 1

Definición: Capacidad para interpretar, representar, comunicar y utilizar información cuantitativa diversa en situaciones de contexto real. Incluye calcular, razonar, emitir juicios y tomar decisiones con base en esta información cuantitativa.

Competencia 3: Diseño de procesos y sistemas

Nivel de logro: 1

Definición: Diseña un sistema, componente o proceso para satisfacer requerimientos solicitados, considerando restricciones realistas de tipo económico, ambiental, social, político, ético, de salud y seguridad, de manufacturabilidad y sostenibilidad.

## V. UNIDADES DE APRENDIZAJE

### UNIDAD N°: 1 Programación Estructurada y modular

#### LOGRO

Competencias(s):

Pensamiento Crítico, Razonamiento Cuantitativo

Logro:

Al finalizar la unidad el alumno construye programas en base a técnicas algorítmicas secuenciales, selectivas y repetitivas haciendo uso de las herramientas del lenguaje de programación C++.

#### TEMARIO

Tema 1:

- Manejo de datos en la programación.
- Algoritmos de programación secuenciales.
- Codificación de algoritmos secuenciales en el lenguaje C++

Actividad de aprendizaje:

- Relatos de experiencias.
- Exposición participativa.
- Ejercicios en PC
- Actividad y cuestionario en línea.

Evidencias de aprendizaje:

- Cuestionarios y tareas en línea para que el estudiante vaya reforzando los conocimientos adquiridos durante la semana.

Bibliografía:

- Joyanes Aguilar, Luis (2013). Fundamentos generales de programación.  
Harvey Deitel, Paul J Deitel. Como programar en C/C++

Tema 2:

- Algoritmos de programación selectivos
- Generación de números aleatorios.
- Codificación de algoritmos selectivos en el lenguaje C++

Actividad de aprendizaje:

- Relatos de experiencias.
- Exposición participativa.
- Ejercicios en PC
- Actividad y cuestionario en línea.

Evidencias de aprendizaje:

- Cuestionarios y tareas en línea para que el estudiante vaya reforzando los conocimientos adquiridos durante la semana.

Bibliografía:

Joyanes Aguilar, Luis (2013). Fundamentos generales de programación. Harvey Deitel, Paul J Deitel. Como programar en C/C++

Tema 3:

- Algoritmos de programación repetitivos
- Codificación de algoritmos repetitivos en el lenguaje C++

Actividad de aprendizaje:

- Relatos de experiencias.
- Exposición participativa.
- Ejercicios en PC
- Actividad y cuestionario en línea.

Evidencias de aprendizaje:

- Cuestionarios y tareas en línea para que el estudiante vaya reforzando los conocimientos adquiridos durante la semana.

Bibliografía:

Joyanes Aguilar, Luis (2013). Fundamentos generales de programación. Harvey Deitel, Paul J Deitel. Como programar en C/C++

Tema 4:

- Primera Evaluación

Actividad de aprendizaje:

- Refuerzo de lo visto en semanas anteriores.

Evidencias de aprendizaje:

PC1.

Desarrollar algoritmos de programación estructurada y codificarlos en el lenguaje de programación C++.

Bibliografía:

Tema 5:

- Concepto de programación modular en programación.
- Definición de funciones por el programador

Actividad de aprendizaje:

- Relatos de experiencias.
- Exposición participativa.
- Video complementario.
- Actividad y Cuestionario en línea.

Evidencias de aprendizaje:

- Cuestionarios y tareas en línea para que el estudiante vaya reforzando los conocimientos adquiridos durante la semana.

Bibliografía:

Joyanes Aguilar, Luis (2013). Fundamentos generales de programación. Harvey Deitel, Paul J Deitel. Como programar en C/C++

Tema 6:

- Codificación de algoritmos modulares en el lenguaje C++

Actividad de aprendizaje:

- Relatos de experiencias.
- Exposición participativa.

-Ejercicios en PC

-Actividad y Cuestionario en línea.

Evidencias de aprendizaje:

-Cuestionarios y tareas en línea para que el estudiante vaya reforzando los conocimientos adquiridos durante la semana.

Bibliografía:

Joyanes Aguilar, Luis (2013). Fundamentos generales de programación. Harvey Deitel, Paul J Deitel. Como programar en C/C++

Tema 7:

-Introducción a la programación gráfica.

-Uso de la librería Miniwin

-Planeamiento del trabajo Final.

Actividad de aprendizaje:

-Relatos de experiencias.

-Exposición participativa.

-Ejercicios en PC

Evidencias de aprendizaje:

-Cuestionarios y tareas en línea para que el estudiante vaya reforzando los conocimientos adquiridos durante la semana.

Bibliografía:

Web:

<http://miniwin.readthedocs.io/en/latest/Instalacion.html>

Tema 8:

-Interacción con el teclado y mouse (miniwin)

Actividad de aprendizaje:

-Actividad y cuestionario en línea.

Evidencias de aprendizaje:

-Cuestionarios y tareas en línea para que el estudiante vaya reforzando los conocimientos adquiridos durante la semana.

#### **HORA(S) / SEMANA(S)**

Semana(s): 1,2,3,4,5,6,7 y 8

### **UNIDAD N°: 2 Herramientas y Técnicas de procesamiento de cadenas y arreglos en la programación**

#### **LOGRO**

Competencias(s):

Pensamiento Crítico, Razonamiento Cuantitativo

Logro:

Al finalizar la unidad el alumno construye programas con técnicas algorítmicas y la herramientas del lenguaje C++ para el procesamiento de cadenas y arreglos

#### **TEMARIO**

Tema 1:

-Concepto de cadena en la computación.

-Algoritmos para el procesamiento de cadenas.

-Codificación de algoritmos en el lenguaje C++ para el manejo de cadenas.

Actividad de aprendizaje:

-Relatos de experiencias.

-Exposición participativa.

-Ejercicios en PC

-Actividad y Cuestionario en línea.

Evidencias de aprendizaje:

-Cuestionarios y tareas en línea para que el estudiante vaya reforzando los conocimientos adquiridos durante la semana.

Bibliografía:

Joyanes Aguilar, Luis (2013). Fundamentos generales de programación. Harvey Deitel, Paul J Deitel. Como programar en C/C++

Tema 2:

-Concepto de arreglo en la computación.

-Algoritmos para el procesamiento de arreglos de 1 y 2 dimensiones.

-Codificación de algoritmos en el lenguaje C++ para el manejo arreglos de 1 y 2 dimensiones.

Actividad de aprendizaje:

-Relatos de experiencias.

-Exposición participativa.

-Ejercicios en PC

-Actividad y Cuestionario en línea.

Evidencias de aprendizaje:

-Cuestionarios y tareas en línea para que el estudiante vaya reforzando los conocimientos adquiridos durante la semana.

Bibliografía:

Joyanes Aguilar, Luis (2013). Fundamentos generales de programación. Harvey Deitel, Paul J Deitel. Como programar en C/C++

Tema 3:

Clase Práctica:

-Desarrollo de algoritmos para el procesamiento de datos en arreglos y cadenas.

-Codificación de algoritmos para el procesamiento de datos en arreglos y cadenas en lenguaje C++

Actividad de aprendizaje:

-Relatos de experiencias.

-Exposición participativa.

-Ejercicios en PC

-Actividad y Cuestionario en línea.

Evidencias de aprendizaje:

-Cuestionarios y tareas en línea para que el estudiante vaya reforzando los conocimientos adquiridos durante la semana.

Bibliografía:

Joyanes Aguilar, Luis (2013). Fundamentos generales de programación. Harvey Deitel, Paul J Deitel. Como programar en C/C++

Tema 4:

-Segunda Evaluación

Actividad de aprendizaje:

-Refuerzo de lo visto en semanas anteriores.

Evidencias de aprendizaje:

PC2.

Desarrollar algoritmos de programación para procesamiento de cadenas y arreglo y codificarlos en el lenguaje de programación C++.

Bibliografía:

**HORA(S) / SEMANA(S)**

Semana(s): 9,10,11 y 12

**UNIDAD N°: 3 Trabajo Final**

## **LOGRO**

Competencias(s):

Pensamiento Crítico, Razonamiento Cuantitativo y diseño de procesos y sistemas

Logro:

Al finalizar la unidad, el alumno elabora un proyecto software (C++ - Librería Miniwin) a partir de un problema de ciencias o ingeniería.

## **TEMARIO**

Tema 1:

-Elaboración de algoritmos de animación codificados en base a la librería miniwin.

Actividad de aprendizaje:

-Relatos de experiencias.

-Exposición participativa.

-Actividad en línea.

Evidencias de aprendizaje:

-Cuestionarios y tareas en línea para que el estudiante vaya reforzando los conocimientos adquiridos durante la semana.

Bibliografía:

Joyanes Aguilar, Luis (2013). Fundamentos generales de programación. Harvey Deitel, Paul J Deitel. Como programar en C/C++

Tema 2:

-Elaboración de algoritmos de animación codificados en base a la librería miniwin.

Actividad de aprendizaje:

-Relatos de experiencias.

-Exposición participativa.

-Actividad en línea.

Evidencias de aprendizaje:

TA.

Cuestionarios y tareas en línea para que el estudiante vaya reforzando los conocimientos adquiridos durante la semana.

Bibliografía:

Joyanes Aguilar, Luis (2013). Fundamentos generales de programación. Harvey Deitel, Paul J Deitel. Como programar en C/C++

Tema 3:

-Presentación del Trabajo Final del curso.

Actividad de aprendizaje:

Evidencias de aprendizaje:

TB.

Exponer las técnicas utilizadas en la elaboración del trabajo final, y documentarlas en un informe impreso

Bibliografía:

Tema 4:

- Examen Final

Actividad de aprendizaje:

Evidencias de aprendizaje:

EB.

Desarrollar algoritmos de programación con todas las técnicas vistas en el ciclo y codificarlos en lenguaje C++

Bibliografía:

## **HORA(S) / SEMANA(S)**

Semana(s): 13,14,15 y 16

## VI. METODOLOGÍA

El Modelo Educativo de la UPC asegura una formación integral, que tiene como pilar el desarrollo de competencias, las que se promueven a través de un proceso de enseñanza-aprendizaje donde el estudiante cumple un rol activo en su aprendizaje, construyéndolo a partir de la reflexión crítica, análisis, discusión, evaluación, exposición e interacción con sus pares, y conectándolo con sus experiencias y conocimientos previos. Por ello, cada sesión está diseñada para ofrecer al estudiante diversas maneras de apropiarse y poner en práctica el nuevo conocimiento en contextos reales o simulados, reconociendo la importancia que esto tiene para su éxito profesional.

Las actividades y productos desarrollados en el curso se sustentan sobre el lema ¿Se aprende haciendo¿ ya que tiene un enfoque teórico-práctico con el uso del laboratorio de cómputo y medios audio-visuales e Internet para la revisión de los conceptos, en donde los estudiantes (de manera individual) primero elaboran una propuesta a la solución a problemas propuestos por el profesor y luego la ejecutan en la PC. El cumplimiento del logro de cada una de las unidades responde a la base sobre la que se trabajará el siguiente, finalizando en una propuesta que engloba el trabajo realizado durante todo el ciclo. El curso combina sesiones presenciales en laboratorio con todo el grupo, y sesiones Blended donde el estudiante deberá dedicar al menos 2 horas a la semana para desarrollar las actividades planteadas.

## VII. EVALUACIÓN

### FÓRMULA

20% (PC1) + 22% (PC2) + 16% (TA1) + 18% (TB1) + 24% (EB1)

TIPO DE NOTA	PESO %
PC - PRÁCTICAS PC	20
PC - PRÁCTICAS PC	22
TA - TAREAS ACADÉMICAS	16
TB - TRABAJO	18
EB - EVALUACIÓN FINAL	24

## VIII. CRONOGRAMA

TIPO DE PRUEBA	DESCRIPCIÓN NOTA	NÚM. DE PRUEBA	FECHA	OBSERVACIÓN	RECUPERABLE
PC	PRÁCTICAS PC	1	SEMANA 6	SE EVALÚA LA UNIDAD 1. EVALUACIÓN INDIVIDUAL	SÍ
PC	PRÁCTICAS PC	2	SEMANA 14	SE EVALÚAN LAS UNIDADES 1 Y 2. EVALUACIÓN INDIVIDUAL	SÍ
TA	TAREAS ACADÉMICAS	1	SEMANA 15	SE EVALÚAN LAS UNIDADES 1 Y 2. EVALUACIÓN INDIVIDUAL	NO
TB	TRABAJO	1	SEMANA 15	SE EVALÚAN TODAS LAS UNIDADES. EVALUACIÓN GRUPAL	NO
EB	EVALUACIÓN FINAL	1	SEMANA 16	SE EVALÚAN TODAS LAS UNIDADES. EVALUACIÓN INDIVIDUAL	SÍ

## **IX. BIBLIOGRAFÍA DEL CURSO**

[https://upc.alma.exlibrisgroup.com/leganto/readinglist/lists/3188086130003391?institute=51UPC\\_INST  
&auth=LOCAL](https://upc.alma.exlibrisgroup.com/leganto/readinglist/lists/3188086130003391?institute=51UPC_INST&auth=LOCAL)