



## PROGRAMA DE CURSO – Primer Semestre 2020

Curso:	Informática 2
Carrera:	Ingeniería en Sistemas
Catedrático:	Ernesto Rodríguez

No. de  
períodos a la  
semana:

Horario:

Área:

Requisito:

Requisito para:

4	Lunes, Martes 6:50 – 8:20 / Jueves 8:40-10:00	Troncal Profesional	Informática 1	
---	--	---------------------	---------------	--

### Objetivos:

- Aprender a desarrollar software de calidad profesional utilizando los métodos y herramientas predominantes en la industria de software.
- Desarrollar y practicar las destrezas de análisis y razonamiento necesarias para convertir una idea o problema en una solución de software.
- Aprender principios importantes de la programación como: Única responsabilidad, programación con interfaces, abstracción, trabajo en equipo, desacoplamiento, etc.
- Familiarizar al estudiante con los mecanismos utilizados por las computadoras digitales/Von Neuman para representar, ordenar y procesar datos

Fecha	Objetivos comunes	Contenido	Actividad	Evaluación
<b>Semana 1.</b> 20 - 24 de Enero	Introducir al estudiante al lenguaje de programación C++ y familiarizar al estudiante con los conceptos y filosofía del lenguaje	Capítulos 2.1 - 2.2	<i>Catedrático: Clase magistral</i>  <i>Alumno: hoja de trabajo</i>	Hoja de trabajo #1
<b>Semana 2</b> 27 - 31 de Enero	Que el estudiante se familiarice con las estructuras para control de flujo en C++.	Capítulo 2.1 - 2.2	<i>Catedrático: Clase magistral</i>	Hoja de trabajo #2
<b>Semana 3</b> 3 – 7 de Febrero	Que el estudiante se familiarice con los tipos de C++ y entienda la	Capítulo 6.1 - 6.2	<i>Catedrático: Clase magistral</i>	Hoja de trabajo #2

	relación que estos tipos tienen con el hardware de una computadora.			
<b>Semana 4</b> 10 – 14 de Febrero	Familiarizar al estudiante con el concepto de una declaración y que el estudiante entienda lo que implica en la memoria de una computadora declarar una variable.	Capítulo 6.3	<i>Catedrático: Clase magistral</i>  <i>Alumno: hoja de trabajo</i>	Hoja de trabajo #3
<b>Semana 4</b> 10 – 14 de Febrero	<b>PRIMERA EVALUACIÓN PARCIAL</b> <b>Nota: en la semana de exámenes parciales se impartirán las clases normalmente.</b> <b>Último día para presentar nota de Primer Parcial: 21 de Febrero</b>			
<b>Semana 5</b> 17 – 21 de Febrero	Que el estudiante comprenda la diferencia entre un objeto y un valor. El estudiante podrá entender cómo el compilador reserva memoria para estos dos tipos de datos.	Capítulo 6.4-6.6	<i>Catedrático: Clase magistral</i>  <i>Alumno: hoja de trabajo</i>	Hoja de trabajo #3
<b>Semana 6</b> 24 -28 de Febrero	Que el estudiante entienda el concepto de un puntero y en general entienda como la memoria está organizada y es accesada por un ordenador.	Capítulo 7.1-7.3	<i>Catedrático: Clase magistral</i>  <i>Alumno: hoja de trabajo</i>	Hoja de trabajo #4
<b>Semana 7</b> 2 de Marzo - 6 de Marzo	Que el estudiante entienda el concepto de un puntero y en general entienda como la memoria está organizada y es accesada por un ordenador.	Capítulos 7.4-7.8	<i>Catedrático: Clase magistral</i>  <i>Alumno: hoja de trabajo</i>	Hoja de trabajo #5
<b>Semana 8</b> 9 de Marzo - 13 de Marzo	Repasar el concepto de tipos en un lenguaje de programación. Que el estudiante entienda la	Capítulos 8	<i>Catedrático: Clase magistral</i>  <i>Alumno: hoja de trabajo</i>	Hoja de trabajo #6

	semántica que usa C++ para reservar y administrar memoria para los diferentes tipos.			
<b>Semana 8</b> 9 de Marzo - 13 de Marzo	<b>SEGUNDA EVALUACIÓN PARCIAL</b> <b>Nota: en la semana de exámenes parciales se impartirán las clases normalmente.</b> <b>Último día para presentar nota de Segundo Parcial: 15 de Marzo</b>			
<b>Semana 9</b> 16-20 de Marzo	Familiarizarse con el concepto de clases y objetos y aprender a utilizar correctamente estos mecanismos de abstracción. Introducir el concepto de modularidad y encapsulación para mejor organización de código.	Capítulos 2.3.1-2.3.3 y 2.4 - 2.4.2 <ul style="list-style-type: none"> <li>• Clases</li> <li>• Modularidad</li> </ul> Capítulo 16	<i>Catedrático: Clase magistral</i>  <i>Alumno: hoja de trabajo</i>	Hoja de trabajo #6
<b>Semana 10</b> 23-27 de Marzo	Familiarizarse con el concepto de clases y objetos y aprender a utilizar correctamente estos mecanismos de abstracción. Introducir el concepto de modularidad y encapsulación para mejor organización de código.	Capítulos 2.3.1-2.3.3 y 2.4 - 2.4.2 <ul style="list-style-type: none"> <li>• Clases</li> <li>• Modularidad</li> </ul> Capítulo 16	<i>Catedrático: Clase magistral</i>  <i>Alumno: hoja de trabajo</i> <i>Develación del proyecto final</i>	Hoja de trabajo #7
<b>Semana 11</b> 30 de Marzo – 3 de Abril	Que el estudiante entienda el concepto de un constructor y un destructor y las implicaciones que estos tienen en cuanto al manejo de memoria. Que el estudiante entienda como una computadora estructura los datos de una clase en memoria.	Capítulo 17	<i>Catedrático: Clase magistral</i>  <i>Alumno: hoja de trabajo</i>	Hoja de trabajo #8
<b>Semana 12</b> 13 – 17 de Abril	Introducir el concepto de operadores y	Capítulo 18	<i>Catedrático: Clase magistral</i> <i>Alumno: hoja de trabajo</i>	Hoja de trabajo #9



	overloading de los mismos.			
<b>Semana 12</b> 13 – 17 de Abril	<b>TERCERA EVALUACIÓN PARCIAL</b> <b>Nota: en la semana de exámenes parciales se impartirán las clases normalmente.</b> <b>Último día para presentar nota de Tercer Parcial: 24 de Abril</b>			
<b>Semana 13</b> 20 – 24 de Abril	Que el estudiante entienda el concepto de herencia y su importancia en la reutilización de código.	Capítulo 20.1 - 20.3	<i>Catedrático: Clase magistral</i> <hr/> <i>Alumno: hoja de trabajo</i>	Hoja de trabajo #10
<b>Semana 14</b> 27 de Abril – 1 de Mayo	Que el estudiante entienda el concepto de herencia y su importancia en la reutilización de código.	Capítulo 20.3-20.7	<i>Catedrático: Clase magistral</i> <hr/> <i>Alumno: hoja de trabajo</i>	Hoja de trabajo #11
<b>Semana 15</b> 4 – 8 de Mayo	Que el estudiante entienda el concepto de plantillas y su importancia en la creación de código genérico y reutilizable.	Capítulo 3.4.2 <ul style="list-style-type: none"> <li>● Plantillas</li> <li>● Funciones genéricas</li> </ul> Capítulo 23.1 - 23.4	<i>Catedrático: Clase magistral</i> <hr/> <i>Alumno: hoja de trabajo</i>	Hoja de trabajo #12
<b>Semana 16</b> 11 – 15 de Mayo	<b>EXÁMENES FINALES</b> <b>22 de mayo. Último día de ingreso de notas FINALES al Blackboard.</b>			

**Evaluación:**

**Zona 60 puntos**  
**Examen Final 40 puntos**  
**TOTAL 100 PUNTOS**

- Para tener derecho a Examen Final se requiere zona mínima de 30 puntos y 80% de asistencia
- Nota mínima para aprobar el curso 61 puntos
- Examen de Recuperación es sobre 40 puntos (conserva la zona)

Artículo	Detalles	Valor unitario	Total
Hoja de trabajo semanal	El estudiante trabajara en una hoja de trabajo		

	semanal. Esta hoja ayudará al estudiante a repasar el contenido aprendido en clase. Durante el semestre habrán 12 hojas de trabajo, sin embargo solo las 10 mejores notas se tomarán en cuenta.	2%	18%
Participación en clase	Durante el semestre, se llevarán a cabo actividades dentro del salón de clase las cuales serán calificadas y entregadas de inmediato.	0.5%	12%
Examen parcial	El contenido aprendido en clase se evaluará mediante tres exámenes parciales.	10%	30%
Examen final	Todo el material estudiado en el curso será evaluado mediante un examen final. Dicho examen tendrá una parte <b>teórica</b> que se elaborará individualmente y una parte <b>práctica</b> que será la entrega de un proyecto final. Este proyecto se puede realizar de forma individual o en parejas pero la nota es individual por alumno.	20% Teórico 20% Práctico	40%

#### Bibliografía:

- B. Stroustrup, The C++ Programming Language, 4ta Edicion, Addison-Wesley (**Texto primario**)
- J. Smiley, Learn to Program with C++, 1ra Edicion, McGraw Hill
- <http://www.cplusplus.com/>, Referencia del lenguaje C++ y sus bibliotecas