

---

## Universidad Hispanoamericana

### Bachillerato en Ingeniería Informática

---

**Nombre del curso:** Programación II

**Código:** INFO-104

**Modalidad:** Cuatrimestral

**Naturaleza del curso:** Teórico-práctico

**Horas lectivas teóricas semanales:** 1

**Horas lectivas laboratorio semanales:** 2

**Horas de estudio independiente semanales:** 9

**Créditos:** 4

**Requisitos:** Programación I

**Correquisitos:** Ninguno

**Ciclo lectivo:** III Cuatrimestre

**Profesor:**

#### Justificación del Curso:

La principal finalidad del curso de programación en C# es proporcionar los conocimientos y aptitudes necesarias para la programación orientada a objetos y la realización de aplicaciones para Windows. Para ello, trabajaremos dentro del entorno de programación Visual utilizando diversos asistentes y los elementos más comunes del lenguaje de programación C++, incluyendo clases y técnicas de programación orientada a objetos.

#### Descripción del curso:

Este es un curso teórico-práctico, el cual pretende introducir al estudiante al mundo de la programación orientada a objetos, por medio del lenguaje de C#, enfrentando problemas comunes de los desarrolladores de software, permitiendo al estudiante evolucionar su creatividad en la programación

#### Objetivo General:

1. Brindar al estudiante los fundamentos de la Programación Orientada a Objetos (POO).
2. Aprovechar el lenguaje de programación C#, contemplando sus facilidades y características a bajo nivel.
3. Brindar al estudiante el conocimiento suficiente de un lenguaje de programación versátil y estandarizada que sirve de base a una amplia variedad de programas de alto nivel como lo es C#

#### Objetivos Específicos:

1. Dominar los conceptos básicos del C#
2. Ampliar y dominar los conceptos generales de C#
3. Dominar el entorno de programación en C#
4. Hacer uso de la Programación Orientada a Objetos (POO) por medio de la programación del lenguaje C#
5. Dominar las estructuras básicas de manejo de punteros, clases (Abstracción, Encapsulamiento, principio de ocultación), herencia y polimorfismo.

## **Contenido Temático:**

### **Tema 1: Introducción al curso**

- Repaso de Programación Estructurada
- Programación Estructurada en lenguaje C#
  - a. Ciclos
  - b. Funciones
  - c. Sobre carga de funciones
  - d. Condicionales
  - e. Variables
    - i. Tipos de Variables
    - ii. Notación Decimal, Binaria y Hexadecimal

### **Tema 2: Introducción a la POO**

- Qué es POO
- Qué es un Objeto / Clase
- Librerías C#

### **Tema 3: Introducción al ambiente Visual**

- Entorno de desarrollo de C#
- Utilización de controles.
- Creación de menús.
- MDI Forms

### **Tema 4: Almacenamiento de información**

- Como acceder a los archivos.

### **Tema 5. Clases**

- Abstracción de Datos
- Definición e Implementación de un Objeto
- Constructores / Destructores
- Alcance de funciones y propiedades miembro
- Objetos como miembros de una clase
- Funciones friend
- Funciones / Miembro Static

### **Tema 6. Herencia**

- Herencia simple (Clase base – Clase derivada)
- Miembros Protected
- Herencia: Public, Private, Protected

### **Tema 7. Polimorfismo**

- Relación en la jerarquía de herencia
- Funciones Virtuales
- Clases abstractas
- Vinculación dinámica

## Tema 8. Bases de Datos

- Uso de Bases de datos con C#

### Metodología de enseñanza

El profesor ubicará a los estudiantes en el contexto del desarrollo de aplicaciones, su importancia y la diferencia con el desarrollo de otro tipo de aplicaciones. Se utilizarán aplicaciones existentes para demostrar las funcionalidades, características y ventajas de las aplicaciones desarrolladas.

Se brindará a los estudiantes todos los conocimientos técnicos necesarios para para diseñar e implementar la mejor solución posible según los criterios dados.

Las tareas y proyectos estarán diseñados para que el alumno investigue y se vea enfrentado a situaciones retadoras, en las cuales deba hacer uso de los conceptos básicos y de los más avanzados que se hayan visto en clase.

### Estrategias de aprendizaje

Para un mayor aprendizaje, se harán discusiones en clase, donde el estudiante participe activamente y pueda haber un intercambio de ideas.

Se utilizarán casos de estudio para fomentar el pensamiento analítico y crítico en los estudiantes y así determinar bajo qué circunstancias es mejor escoger un lenguaje u otro para el desarrollo de aplicaciones web.

Los proyectos se utilizarán para que los estudiantes puedan experimentar y llegar a conocer los distintos lenguajes de programación que se utilizan en el desarrollo de aplicaciones web y se requiere de un esfuerzo especial por parte de ellos, donde deberá integrar todos los conocimientos adquiridos en el curso para diseñar e implementar una aplicaciones web completamente funcionales.

### Recursos Educativos:

Los temas principales son desarrollados con la ayuda de los equipos multimedia existente en las aulas, esto se realiza por medio de Power Point.

Se hará uso de los libros y manuales que están en la biblioteca para consulta de los estudiantes.

Laboratorios de cómputo.

### Políticas de Evaluación:

Descripción	Porcentaje
Parcial 1 Semana 6	15%
Trabajo Extra Semana 11	15%
Examen Final (Acumulativo) Temas del 1 al 5 emana 15	30%
Proyecto 1 Semana 14	25%
Tareas (7.5%), Quices(7.5%)	15%
Total	100%

### **Normativas del Curso:**

#### **Reglas Generales del Curso:**

1. Los quices NO se avisarán, y podrán ser realizados en cualquier momento.
2. NO hay reposición de quices.
3. Las tareas deberán ser entregadas en la fecha solicitada en forma individual. Se recibirán únicamente al inicio de la lección.
4. NO se aceptarán tareas en fechas posteriores.
5. Los proyectos se realizará en parejas, y es requisito para pasar el curso.
6. Tanto las tareas como proyectos se deben entregar en sobre de manila, con su nombre completo, carnet, curso. Impresión de comprobación y fuentes.
7. Las copias de tareas o proyectos se eliminaran y se tomara como nota un 1 sobre 100.

#### **Ausencias:**

1. Se consideran como persona ausente a aquella que no este en la clase después de haber transcurrido los primeros 20 minutos de la clase.
2. El profesor podrá poner ausente a un estudiante cuando este al menos no haya estado en la clase un 40% de la duración de la misma.
3. La justificación de ausencias la tiene que realizar de forma escrita. No se aceptaran justificaciones de manera verbal.
4. El estudiante tendrá un máximo de tres días hábiles para justificar la ausencia.
5. Si un estudiante llegara en el transcurso de los primeros 20 minutos a clase, no quedara como ausente sino como llegada tardía.
6. Es deber del estudiante mantenerse al día con la materia del curso, en caso de ausencia.

#### **Quices:**

1. Los Quices no se reponen y son sin previo aviso.
2. Los quices deben estar en un ZIP (no RAR por favor) y con solo los archivos que se hacen mención en el quiz o que se necesitan para la revisión del mismo.
3. Si un archivo presenta un virus dicho quiz, tarea o examen no se revisara y obtendrá como nota un 1.
4. Si el estudiante no presenta los archivos necesarios para las revisiones del quiz, no se calificara.
5. No es responsabilidad del profesor o de la Universidad si en la realización de un quiz, examen o proyecto, el estudiante pierde información ya sea por cortes de corriente, mal manejo de información, problemas en las maquinas o por alguna otra circunstancia. El estudiante debe siempre mantener un respaldo de lo que hace.

#### **Exámenes:**

1. Es deber del estudiante siempre respaldar la información en disco o cualquier otro medio, Si el estudiante no presenta los archivos necesarios para revisiones del examen, no calificara (nota1).

2. Para seguridad del estudiante debería de presentar los exámenes en dos discos, uno para que sirva como respaldo de la información. (exámenes normales y de ampliación).
3. Se deben de entregar en ZIP (No rar)
4. El estudiante tendrá derecho a realizar consultas acerca del examen, durante los primeros 20 minutos del mismo, y con relación al no entendimiento de la redacción de alguna pregunta.

### Proyectos

1. Son requisitos para pasar el curso. Si un estudiante no presenta uno, pierde el curso.
2. Se debe de entregar al inicio de la clase, el día de la entrega. Si un estudiante lo entrega tarde se rebajaran 20pts de 100 por la entrega tardía del proyecto.
3. Por día de retraso tendrá una multa de 10pts.
4. No se aceptaran copias o trabajo en conjunto.
5. Los proyectos se deben entregar en sobre de Manila, con su nombre completo, carnet, curso y además de las pantallas de comprobación.
6. Es deber del estudiante siempre respaldar la información en el disco o cualquier otro medio. Si el estudiante no presenta los archivos necesarios para las revisiones del proyecto, no se calificara (Nota 1).
7. Los proyectos se deben de defender y la defensa tendrá una evaluación. Si una persona no puede defender el proyecto perderá un porcentaje del proyecto.

### Modalidad:

Las clases se impartirán magistralmente y se realizarán prácticas semanales en laboratorio.

El profesor asignará prácticas programadas para reforzar los contenidos desde una perspectiva programática.

La asistencia a clase y a la práctica semanal es obligatoria, para un aprovechamiento óptimo del curso.

Los exámenes se realizarán en el horario de clase, en las semanas 6, 11 y 15.

Para alcanzar los objetivos del curso eficientemente, el profesor se valdrá de:

Actividad	Descripción
<b>Proyectos Programados</b>	Tienen como objetivo que los estudiantes sigan y apliquen las fases respectivas, para la resolución de problemas informáticos. Los proyectos van incrementando el nivel de complejidad de acuerdo a los temas vistos en clase.
<b>Tareas Cortas</b>	Se utilizan para guiar al estudiante conocer el entorno de programación y dar énfasis a los temas vistos en clase. Estimula la práctica fuera del horario de clase.
<b>Quices</b>	Permiten medir el nivel de conocimiento de la materia que tienen los estudiantes.

<b>Laboratorios</b>	Los estudiantes ponen en práctica la materia vista en clase mediante las prácticas semanales en el laboratorio. Estas prácticas son dirigidas por el profesor del curso, siguiendo una guía semanal. También permiten que los estudiantes se familiaricen con el entorno de programación.
---------------------	---

**Bibliografía del Curso:**

Joyanes, Luis. Programación en C#. Mc Graw Hill  
Joyanes, Luis. Programación C, C# Java y UML. Mc Graw Hill  
Balagurusamy B. Programación Orientada a objetos C++. Mc Graw Hill  
Corona, María Adriana Diseño de algoritmos y su codificación en lenguaje C. Mc Graw Hill  
Deitel & Deitel. Como programar en C#. Person-Prentice Hall  
Adams. Programación avanzada de gráficos en C para Windows. Mc Graw Hill  
Cox, Brand. Programación Orientada a Objetos. Addison Wesley  
Date. El lenguaje de Programación C. Prentice Hall