

Programa del curso CE1102

Taller de Programación

Área Ingeniería en Computadores
Licenciatura Ingeniería en Computadores

I parte: Aspectos relativos al plan de estudios

1 Datos generales

Nombre del curso:	Taller de Programación
Código:	CE1102
Tipo de curso:	Teórico – Práctico
Electivo o no:	Obligatorio
Nº de créditos:	2
Nº horas de clase por semana:	4
Nº horas extraclase por semana:	2
% de las áreas curriculares:	Ciencias de la Ingeniería (75%) Diseño de Ingeniería (25%)
Ubicación en el plan de estudios:	Curso del 1 ^{er} semestre del Programa de Licenciatura en Ingeniería en Computadores.
Requisitos:	No hay
Correquisitos:	Introducción a la Programación (CE-1101)
El curso es requisito de:	Algoritmos y Estructuras de Datos I (CE-1103)
Asistencia:	Obligatoria
Suficiencia:	El curso no puede ser presentado por suficiencia
Posibilidad de reconocimiento:	Sí
Vigencia del programa:	II Semestre 2022

2 Descripción general

“Try a thing you haven't tried before three times:- once to get over the fear, once to find out how to do it, and a third time to find out whether you like it or not”.

[Virgil Thomson](#) ([American Composer](#), [1896-1989](#))

Este curso introduce los conceptos de organización de un sistema computacional y herramientas necesarias para el entender la operación de un sistema computacional y para el desarrollo de programas. Asimismo, ejercita las destrezas y las habilidades adquiridas en el curso de "Introducción a la Programación" (CE-1101) con el fin de desarrollar experiencia en el análisis, diseño, implementación, desarrollo y validación de programas, cimentando las bases para el curso “Algoritmos y Estructuras de Datos” (CE-1103).

Adicionalmente se estimula la participación en clase y el trabajo en grupo para la presentación de propuestas de solución a problemas planteados en el aula, proyectos grupales y exposiciones.

La metodología del curso busca además reforzar ciertas habilidades tales como el trabajo en equipo y el pensamiento creativo. El desarrollo de estas habilidades permitirá el desarrollo fluido de los diferentes proyectos y tareas. Adicionalmente se espera reforzar la creación de documentación técnica como informes y bitácoras.

Los atributos que se ejercitan a lo largo de las diferentes actividades de este y que se proponen que el estudiante pueda alcanzar al final del curso son:

- Conocimiento de Ingeniería (CI) en nivel inicial.
- Ética y Equidad (EE) en nivel inicial.
- Trabajo Individual y de Equipo (TE) en nivel inicial.

El curso es adaptable a necesidades educativas especiales que puedan tener los estudiantes.

3 Objetivos

Objetivo General:

Al finalizar este curso el estudiante debe ser capaz de:

- Construir programas con base en los elementos básicos de un ambiente de programación, analizar los conceptos de las ciencias de la computación relacionados con los sistemas computacionales que inciden en el desarrollo de programas y en su desempeño.

Objetivos Específicos:

Objetivo(s) del curso	Atributo(s) correspondiente(s)	Nivel de desarrollo de cada atributo que se planea alcanzar: Inicial - I, intermedio - M o avanzado - A
1. Manejar los conceptos básicos de la organización de computadores.	Conocimiento Ingeniería (CI).	Inicial(I)
2. Manejar las herramientas más utilizadas en un ambiente de desarrollo de programación.	Conocimiento de Ingeniería (CI)	Inicial(I)
3. Analizar, diseñar, implementar, validar y documentar programas.	Conocimiento de Ingeniería (CI)	Inicial(I)
4. Manejar conceptos y terminología fundamentales en las ciencias de la computación.	Conocimiento de Ingeniería (CI)	Inicial (I)
5. Elaborar, organizar y cumplir con normas grupales para el desarrollo de proyectos de programación en equipo, así como de elaboración de documentación técnica, de manera colaborativa, ordenada y concisa, haciendo uso de principios y valores como responsabilidad, respeto y tolerancia.	Trabajo Individual y en Equipo (TE)	Inicial (I)
6. Conocer aspectos importantes del papel de la programación dentro de la computación, en el contexto de la realidad nacional e internacional.	Ética y Equidad (EE)	Inicial (I)

4 Contenidos

Teoría de Sistemas (4 semanas)

- Modelado general de problemas (modelo de caja negra)
- Máquinas.
- Máquina virtual, multinivel.
- Máquinas de Estados Finitos.
- Máquinas de Turing.

Sistemas numéricos (3 semanas)

- El sistema binario
- Otros sistemas numéricos
- Conversiones de base
- Aritmética en sistemas numéricos
- Representación de los números enteros
- Representación de fracciones
- Representación de otros tipos de datos

Organización de un sistema computacional (3 semanas)

- Antecedentes históricos
- Arquitectura de un computador
- Memoria
- Unidad Central de Procesamiento
- Almacenamiento secundario
- Codificación de la información
- Otras arquitecturas

Sistema Operativo (1 semana)

- Funciones del sistema operativo
- Características virtuales
- Evolución del sistema operativo
- Arquitectura del sistema operativo.

Programación de bajo nivel de un computador (3 semanas)

- Antecedentes históricos
- El programa almacenado
- Ejecución de un programa
- Lenguaje de máquina, lenguaje ensamblador

El ambiente de desarrollo (14 semanas)

- El ambiente de Python
- Entornos de programación en equipo
- El ambiente Java

II parte: Aspectos operativos

5 Metodología de enseñanza y aprendizaje

Este curso es complementario al curso "Introducción a la Programación" (CE-1101), cada semana consiste de 4 horas en una clase magistral, conferencias, videos, etc. Se hará énfasis en las buenas prácticas de ingeniería en análisis, diseño, programación y pruebas. Además, se combinará con una participación activa por parte de los estudiantes durante el transcurso de las lecciones, por medio de llamadas orales, respuestas a casos en la pizarra y de trabajos en grupo.

Se presupone que el alumno profundiza los temas abordados en la clase en tanto con las lecturas recomendadas por el profesor como las que el estudiante busca por su cuenta y que el estudiante será responsable de desarrollar los proyectos programados y prácticas que se asignen en el curso. Para completar esta labor el estudiante debe dedicar, en promedio, dos horas trabajo extraclase.

También requiere que el estudiante integre un equipo de programación de dos personas, elabore actividades individuales dentro del trabajo en equipo, organice el trabajo, cumpla con los acuerdos y normas del trabajo en equipo. El estudiante podría realizar exposiciones individuales cortas sobre temas asignados por el profesor.

Adicionalmente, con el fin de evaluar los conocimientos adquiridos, los estudiantes elaboraran pruebas cortas, evaluaciones, tareas, proyectos programados, así como participación en clase.

El profesor incluirá en los proyectos y tareas los rubros de evaluación de los mismos. Revisará los proyectos en una cita sincrónica con el o los estudiantes. La asistencia, cámara por parte de los integrantes durante estas revisiones es de carácter obligatorio.

El estudiante requiere acceso a un computador con servicio de Internet. El computador puede ser personal o de los laboratorios institucionales.

6 Evaluación

Los siguientes son los rubros que se evaluarán en el curso:

Rubro	Valor (%)	Tiempo estimado de realización
Proyectos programados (2 o 3)	55% (25%, 30 % si son 2 o 15%, 20%, 20% si son 3 proyectos)	Semana 7, Semana 11,(Si 3) Semana 17
Primer Examen Parcial	15%	Semana 9
Segundo Examen Parcial	10%	Semana 16
Exámenes cortos, tareas y exposiciones	20%	1 por semana
Total	100	

Los exámenes se realizarán en las semanas 9 y 16 respectivamente

No es posible eximirse de ninguna evaluación del curso.

Los proyectos programados están divididos en uno individual y uno o dos en grupos de dos estudiantes. Los proyectos se deben defender previa cita con el profesor. Dudas de los proyectos se pueden resolver con el profesor previa cita o los asistentes a través del grupo de Telegram, Facebook o el grupo de Google.

Las exposiciones son de seis a siete minutos de duración. La duración exacta la especificará el profesor. Los rubros de evaluación de la misma serán presentados por el profesor con antelación. Se considerarán entre otros aspectos como: Medios Utilizados, Introducción, Desarrollo del Tema, Conexión con de la programación con la realidad Nacional e Internacional y manejo del tiempo.

En los Exámenes habrá preguntas de los diferentes conceptos relacionándolos con la realidad nacional e internacional en cuanto a desarrollo de proyectos de software.

Las tareas son de una hoja por los dos lados.

7 Bibliografía

Obligatoria:

- Downey, A.; Elkner, J.; Meyers, C. How to think like a computer scientist. Green Tea Press. 2002.
- Deitel, H. et all. Python: How to program. Prentice Hall, 2002.
- Levine, G. Computación y programación moderna. Pearson Education, 2002.

Complementaria:

- Abelson, H. Sussman, G. Structure and Interpretation of Computer Programs. The MIT Press, McGraw-Hil Company, Massachusetts, 1985.
- Peterson, J.L.; Silberschatz, A. Operating Systems Concepts. Addison-Wesley, 1983.
- PTL Group. DrScheme Reference Manual.
- PTL Group. DrScheme User's Manual.

8 Profesores **Jeff Schmidt Peralta**

- Grupo 01
- Aula F2-07(Caso presencial)
- Consultas: grupo de Telegram o al correo electrónico del profesor.
- Clases sincrónicas vía Grupo Teams
- Consulta Teams: Miércoles de 1:30 pm a 3:30 pm.
- Correo electrónico: jschmidtcr@gmail.com

Milton Villegas Lemus

- Grupo: 02.
- Aula F2-10 (Caso presencial)
- Clases Sincrónicas vía Grupo de Microsoft Teams
- Muro de Consultas: Google Classroom Taller G2S222
- Consulta Teams Equipo Intro/Taller, Martes de 4:30 pm a 6:00 pm.
- Correo electrónico: mvilem@gmail.com
- Asistente: Marcelo Truque: mtruquem@gmail.com

9 Normativa del curso

1. El curso se aprueba con nota de 70. No hay examen de reposición.
2. Las actividades o trabajos académicos deben cumplir con las normas dictadas por el Consejo de Escuela. Si alguna actividad implica la entrega de un medio físico, debe estar libre de virus. Si tuviese virus tendrá nota de cero.
3. No se aceptarán trabajos después de la fecha y hora indicadas. Por lo tanto, trabajos entregados tardíamente tendrán una nota de cero. Los quices no se reponen.
4. Para proyectos, el profesor especificará la penalización por entrega extemporánea según su criterio.
5. Los exámenes escritos presenciales han de entregarse en cuadernillos, debidamente engrapados y presentados. Para los exámenes sincrónicos, éstos son a mano alzada, se deben digitalizar (scan), enviarse a la hora y fecha indicada a la asignación abierta en el Classroom para ello. El contenido académico de las actividades, ya sean estas tareas, exposiciones, quices, exámenes parciales o proyectos, son acumulativos.
6. El contenido académico de las actividades, ya sean estas tareas, exposiciones, quices, exámenes parciales o proyectos, son acumulativos.
7. Los fraudes en cualquier actividad o asignación llevada a cabo durante el semestre, implicará que se reportará la nota mínima y se aplicará la reglamentación establecida. Además, si aplica se reportará una carta al expediente del estudiante.
9. El profesor se reserva el derecho de calificar forma y fondo de las actividades tomando como referencia la mejor actividad presentada.