

PROGRAMA DE LABORATORIO

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS



LABORATORIO DE INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN

2

CÓDIGO:	0771	PUNTEO NETO LABORATORIO:	28.75
ESCUELA DE INGENIERÍA EN:	CIENCIAS Y SISTEMAS	ÁREA A LA QUE PERTENECE:	DESARROLLO DE SOFTWARE
PRE REQUISITO:	[0107] MATEMÁTICA INTERMEDIA 1 [0770] INTROD A LA PROGRAMACIÓN Y COMPUTACIÓN 1 [0795] LÓGICA DE SISTEMAS [0960] MATEMÁTICA PARA COMPUTACIÓN 1	POST REQUISITO:	[0777] ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 1 [0964] ORGANIZACIÓN COMPUTACIONAL [0772] ESTRUCTURAS DE DATOS [0601] INVESTIGACIÓN DE OPERACIONES I
CATEGORÍA:	OBLIGATORIO	VIGENCIA:	SEGUNDO SEMESTRE 2026

Descripción del Laboratorio

El laboratorio de Introducción a la Programación y Computación II cubrirá conceptos y herramientas para el desarrollo de programas de software utilizando conceptos de POO y el lenguaje C#.

Resumen de Ponderaciones y Tiempo de Auto-Aprendizaje

TIPO	CANTIDAD	PONDERACIÓN	HORAS DE AUTO-APRENDIZAJE
Proyectos	3	27	360
Examen Final	1	3	0
TOTAL	4	30	360

Desglose de Actividades

TIPO DE ACTIVIDAD	PONDERACIÓN	HORAS DE AUTO-APRENDIZAJE
Proyecto No. I - Entregable 1 (Envío: 31/07/2026, Entrega: 27/08/2026)	9	120
Proyecto No. II - Entregable 2 (Envío: 29/08/2026, Entrega: 24/09/2026)	9	120
Proyecto No. III - Entregable 3 (Envío: 26/09/2026, Entrega: 22/10/2026)	9	120
Evaluación Final (los proyectos 1 y 2 aportarán 1.5 puntos y el proyecto 3, 1.75 puntos, a esta evaluación)	3	X
TOTAL	30	360

Equipo Académico

Coordinador del Área

Nombre: MARLON FRANCISCO ORELLANA LOPEZ	Correo electrónico: marlonorellana2005@gmail.com
---	--

Docente

Ing. Marlon Antonio Pérez Turk	2346363510101@ingenieria.usac.edu.gt
--------------------------------	--------------------------------------

	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Día				X	X	
Horario				7:10 - 8:50	7:10 - 8:50	
Lugar				Meet	Meet	

Tutor

Nombre del Tutor	Eddy Alejandro Murga Barillas	
Correo electrónico institucional	2552889210101@ingenieria.usac.edu.gt	

Tipo		Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Clase	Día		X				
	Horario		17:20 - 19:00				
	Lugar		Meet				
Atención al Estudiante	Día	X			X		
	Horario	17:20 - 19:00			17:20 - 19:00		
	Lugar						

Índice

Descripción del Laboratorio	1
Resumen de Ponderaciones y Tiempo de Auto-Aprendizaje	1
Desglose de Actividades	1
Equipo Académico	2
Coordinador del Área	2
Docente	2
Tutor (Encargado de la Sección Principal)	3
Competencias Vinculadas al Perfil del Egresado	4
Competencias Específicas	4
Competencias Generales	5
Competencias del Laboratorio	5
Competencia(s) Específica(s)	5
Competencia(s) General(es)	6
Diseño Didáctico	6
Sesión de Diagnóstico	6
Sesión No. 2, Unidad 1 - Introducción a C#	7
Sesión No. 3, Unidad 2 - Manejo de Versiones	8
Sesión No. 4, Unidad 3 - Procesamiento de Datos XML	9
Sesión No. 5, Unidad 3 - Procesamiento de Datos JSON	10
Sesión No. 6, Unidad 4 - Estructuras de Datos Lineales	11
Sesión No. 7, Unidad 4 - Estructuras de Datos No Lineales (Árboles)	12
Sesión No. 8, Unidad 5 - HTML5 y CSS	13
Sesión No. 9, Unidad 6 - Entorno para Desarrollo Web en C#	14
Sesión No. 10, Unidad 7 - Web Development	15
Sesión No. 11, Unidad 8 - Acceso a Datos Web	16
Rúbrica de Evaluación	17
Normativa Académica y Ética del Curso	17
Bibliografía	18
E-Grafía	18

Competencias Vinculadas al Perfil del Egresado

Competencias Específicas

No.	Competencia
1	Demuestra destreza y habilidad en la selección, uso y adaptación de herramientas metodológicas, tecnológicas, equipos especializados y en la lectura e interpretación de datos, pertinentes al contexto de su ejercicio profesional.
2	Aplica los conocimientos de su disciplina en la elaboración, fundamentación y defensa de argumentos para prevenir y resolver problemas complejos en su campo profesional, identificando y aplicando innovaciones.
3	Toma decisiones profesionales con base en fundamentos teóricos, datos e información pertinente, válida y confiable.

Competencias Generales

No.	Competencia
1	Aplica principios básicos de ingeniería, ciencias de computación y sistemas de información y comunicación, en la formulación y resolución adecuada de problemas complejos.
2	Maneja e interpreta adecuadamente datos masivos, sean estos estructurados o no estructurados, facilitando su visualización e interpretación de forma eficaz en apoyo a la toma de decisiones.
3	Aplica conocimientos tecnológicos con ética profesional y respetando y cuidando los recursos naturales, humanos y financieros.

Competencias del Laboratorio

Competencia(s) Específica(s)

No.	Competencia	Nivel de Aprendizaje
1	Aplica los fundamentos de la Programación Orientada a Objetos mediante el lenguaje C#, empleando estructuras de control, funciones y los pilares del paradigma (herencia, abstracción, polimorfismo y encapsulamiento), para resolver problemas de programación.	Aplicar
2	Aplica prácticas de control de versiones mediante Git, incluyendo la configuración de repositorios y la implementación de troncales y ramas, para mantener un historial de cambios ordenado en proyectos de software.	Aplicar
3	Procesa información estructurada en formato XML y JSON mediante las librerías nativas de C#, aplicando serialización, deserialización, lectura y escritura de datos, para integrar fuentes de datos externas en aplicaciones de software.	Aplicar

4	Crea estructuras de datos lineales y no lineales (listas enlazadas, árboles binarios de búsqueda y árboles AVL) mediante Programación Orientada a Objetos en C#, para implementar soluciones eficientes de almacenamiento y organización de datos.	Crear
5	Crea páginas web utilizando HTML5 y CSS, aplicando una estructura semántica y estilos visuales coherentes, para el desarrollo de interfaces web funcionales.	Crear
6	Desarrolla aplicaciones web full-stack con el framework ASP.NET Core MVC y JavaScript, integrando vistas, modelos y templates con Razor en el backend y el frontend, con consumo de APIs mediante HTTP para el acceso a datos externos.	Crear

Competencia(s) General(es)

No.	Competencia	Nivel de Aprendizaje
1	Analiza los fundamentos del desarrollo de software full-stack, integrando Programación Orientada a Objetos, control de versiones, procesamiento de información estructurada, estructuras de datos y desarrollo web, para construir soluciones de software completas y bien organizadas.	Analizar

Diseño Didáctico

Sesión de Diagnóstico

Evaluación de conocimientos previos

Se aplicará una actividad diagnóstica con el objetivo de identificar el nivel de conocimientos y habilidades que los estudiantes poseen al inicio del curso. No influye en la nota final, pero es obligatoria para todos los estudiantes.

Tipo de Actividad	Descripción
Cuestionario interactivo	Se utilizarán plataformas como kahoot o Wayground para evaluar el conocimiento previo de los estudiantes.

Presentación del tutor

El tutor se presenta formalmente al grupo, compartiendo su formación académica, experiencia profesional y educativa, así como sus expectativas sobre el curso. También se abordan aspectos como normas de convivencia, canales de comunicación, disponibilidad para consultas y métodos de acompañamiento.

Presentación de los estudiantes

Se escogen un grupo de estudiantes al azar. En su presentación, se les pedirá que compartan información básica como su nombre, intereses personales o profesionales, experiencias previas relacionadas con el curso y sus expectativas. Esta actividad busca promover la interacción, el reconocimiento entre pares y la construcción de un entorno participativo y respetuoso.

Presentación del programa del curso

Se presenta el contenido del programa del curso, se aclaran dudas y se fomenta el compromiso del estudiante con su aprendizaje.

Evaluación de conocimientos del laboratorio actual

Se realiza una evaluación o práctica que permite conocer el grado de familiaridad de los estudiantes con las herramientas, entornos o competencias técnicas necesarias para el laboratorio actual.

Tipo de Actividad	Descripción
Cuestionario interactivo	Se utilizarán plataformas como kahoot o Wayground para evaluar el conocimiento previo de los estudiantes.

Sesión No. 2, Unidad 1 - Introducción a C#

Área Actitudinal (Saber ser)

Nombre del valor:
Descripción

Área de Conocimiento (Saber)

Competencia(s)	
Aplica los fundamentos de la Programación Orientada a Objetos mediante el lenguaje C#, empleando estructuras de control, funciones y los pilares del paradigma (herencia, abstracción, polimorfismo y encapsulamiento), para resolver problemas de programación.	
Tema	Subtema
Introducción a C#	Instalación de C#14 (.NET SDK 10) y entorno de desarrollo Visual Studio
Introducción a C#	Variables, expresiones y sentencias
Introducción a C#	Ejecución condicional
Introducción a C#	Funciones
Introducción a C#	Iteración
Introducción a C#	Strings

Programación Orientada a Objetos	Clases, Objetos, Métodos y Atributos
Programación Orientada a Objetos	Pilares: Herencia, Abstracción, Polimorfismo, Encapsulamiento

Área de Habilidades (Saber Hacer)

Competencia	
Aplica los fundamentos de la Programación Orientada a Objetos mediante el lenguaje C#, empleando estructuras de control, funciones y los pilares del paradigma (herencia, abstracción, polimorfismo y encapsulamiento), para resolver problemas de programación.	
Tipo de Actividad	Ponderación
Ejercicio práctico: programa de consola en C# aplicando estructuras de control y POO básica	0

Sesión No. 3, Unidad 2 - Manejo de Versiones

Área Actitudinal (Saber ser)

Nombre del valor:
Descripción

Área de Conocimiento (Saber)

Competencia(s)	
Aplica prácticas de control de versiones mediante Git, incluyendo la configuración de repositorios y la implementación de troncales y ramas, para mantener un historial de cambios ordenado en proyectos de software.	
Tema	Subtema
Manejo de versiones	Conceptos y fundamentos
Manejo de versiones	Configuración de software para versionar
Manejo de versiones	Implementación de troncales y ramas

Área de Habilidades (Saber Hacer)

Competencia
Aplica prácticas de control de versiones mediante Git, incluyendo la configuración de repositorios y la implementación de troncales y ramas, para mantener un historial de cambios ordenado en proyectos de

software.	
Tipo de Actividad	Ponderación
Ejercicio práctico: configuración de un repositorio Git con ramas y control de versiones	0

Sesión No. 4, Unidad 3 - Procesamiento de Datos XML

Área Actitudinal (Saber ser)

Nombre del valor:
Descripción

Área de Conocimiento (Saber)

Competencia(s)	
Procesa información estructurada en formato XML y JSON mediante las librerías nativas de C#, aplicando serialización, deserialización, lectura y escritura de datos, para integrar fuentes de datos externas en aplicaciones de software.	
Tema	Subtema
Procesamiento de datos XML	Introducción a XML
Procesamiento de datos XML	Sintaxis de XML
Procesamiento de datos XML	Procesamiento de datos XML con XDocument C#
Procesamiento de datos XML	Lectura y Escritura XML

Área de Habilidades (Saber Hacer)

Competencia	
Procesa información estructurada en formato XML y JSON mediante las librerías nativas de C#, aplicando serialización, deserialización, lectura y escritura de datos, para integrar fuentes de datos externas en aplicaciones de software.	
Tipo de Actividad	Ponderación
Ejercicio práctico: lectura y escritura de datos en formato XML con C#	0

Sesión No. 5, Unidad 3 - Procesamiento de Datos JSON

Área Actitudinal (Saber ser)

Nombre del valor:
Descripción

Área de Conocimiento (Saber)

Competencia(s)	
Procesa información estructurada en formato XML y JSON mediante las librerías nativas de C#, aplicando serialización, deserialización, lectura y escritura de datos, para integrar fuentes de datos externas en aplicaciones de software.	
Tema	Subtema
Procesamiento de datos JSON	Introducción a JSON
Procesamiento de datos JSON	Sintaxis de JSON
Procesamiento de datos JSON	Tipos de datos JSON
Procesamiento de datos JSON	Serialización y Deserialización en C#
Procesamiento de datos JSON	Lectura y Escritura de JSON
Procesamiento de datos JSON	JSON vs XML

Área de Habilidades (Saber Hacer)

Competencia	
Procesa información estructurada en formato XML y JSON mediante las librerías nativas de C#, aplicando serialización, deserialización, lectura y escritura de datos, para integrar fuentes de datos externas en aplicaciones de software.	
Tipo de Actividad	Ponderación
Ejercicio práctico: serialización y deserialización de datos en formato JSON con C#	0

Sesión No. 6, Unidad 4 - Estructuras de Datos Lineales

Área Actitudinal (Saber ser)

Nombre del valor:
Descripción

Área de Conocimiento (Saber)

Competencia(s)	
Crea estructuras de datos lineales y no lineales (listas enlazadas, árboles binarios de búsqueda y árboles AVL) mediante Programación Orientada a Objetos en C#, para implementar soluciones eficientes de almacenamiento y organización de datos.	
Tema	Subtema
Estructuras de Datos Lineales	Tipos de datos abstractos (TDA's)
Estructuras de Datos Lineales	Lista enlazada simple
Estructuras de Datos Lineales	Lista doblemente enlazada
Estructuras de Datos Lineales	Lista circular

Área de Habilidades (Saber Hacer)

Competencia	
Crea estructuras de datos lineales y no lineales (listas enlazadas, árboles binarios de búsqueda y árboles AVL) mediante Programación Orientada a Objetos en C#, para implementar soluciones eficientes de almacenamiento y organización de datos.	
Tipo de Actividad	Ponderación
Ejercicio práctico: implementación de listas enlazadas simples y dobles en C#	0

Sesión No. 7, Unidad 4 - Estructuras de Datos No Lineales (Árboles)

Área Actitudinal (Saber ser)

Nombre del valor:
Descripción

Área de Conocimiento (Saber)

Competencia(s)	
Crea estructuras de datos lineales y no lineales (listas enlazadas, árboles binarios de búsqueda y árboles AVL) mediante Programación Orientada a Objetos en C#, para implementar soluciones eficientes de almacenamiento y organización de datos.	
Tema	Subtema
Árboles	Definición y terminología

Árboles binarios de búsqueda	Operaciones (inserción, búsqueda, eliminación)
Árboles binarios de búsqueda	Recorridos (Inorden, preorden, posorden)
Árboles AVL	Definición
Árboles AVL	Operaciones
Árboles AVL	Rotaciones (simples, dobles)

Área de Habilidades (Saber Hacer)

Competencia	
Crea estructuras de datos lineales y no lineales (listas enlazadas, árboles binarios de búsqueda y árboles AVL) mediante Programación Orientada a Objetos en C#, para implementar soluciones eficientes de almacenamiento y organización de datos.	
Tipo de Actividad	Ponderación
Ejercicio práctico: implementación de árboles binarios de búsqueda y árboles AVL en C#	0

Sesión No. 8, Unidad 5 - HTML5 y CSS

Área Actitudinal (Saber ser)

Nombre del valor:
Descripción

Área de Conocimiento (Saber)

Competencia(s)	
Crea páginas web utilizando HTML5 y CSS, aplicando una estructura semántica y estilos visuales coherentes, para el desarrollo de interfaces web funcionales.	
Tema	Subtema
HTML5	Introducción a HTML5 y estructura básica
HTML5	Elementos, atributos y párrafos
HTML5	Estilos, tablas e imágenes
HTML5	Listas y elementos de un formulario
HTML5	Tipos y atributos del elemento "input"
CSS	Módulos CSS y conexión con HTML
CSS	Sintaxis, variables y selectores CSS

Área de Habilidades (Saber Hacer)

Competencia	
Crea páginas web utilizando HTML5 y CSS, aplicando una estructura semántica y estilos visuales coherentes, para el desarrollo de interfaces web funcionales.	
Tipo de Actividad	Ponderación
Ejercicio práctico: desarrollo de una página web con HTML5 y CSS	0

Sesión No. 9, Unidad 6 - Entorno para Desarrollo Web en C#

Área Actitudinal (Saber ser)

Nombre del valor:
Descripción

Área de Conocimiento (Saber)

Competencia(s)	
Desarrolla aplicaciones web full-stack con el framework ASP.NET Core MVC y JavaScript, integrando vistas, modelos y templates con Razor en el backend y el frontend, con consumo de APIs mediante HTTP para el acceso a datos externos.	
Tema	Subtema
Framework ASP.NET Core	Back End Development y programación del lado del servidor
Framework ASP.NET Core MVC	Vistas, URLs y Apps MVC
Framework ASP.NET Core MVC	Creando una nueva App
Framework ASP.NET Core MVC	Modelos y creación de módulos
Framework ASP.NET Core MVC	Creando instancias de Modelos y Vistas

Área de Habilidades (Saber Hacer)

Competencia
Desarrolla aplicaciones web full-stack con el framework ASP.NET Core MVC y JavaScript, integrando vistas, modelos y templates con Razor en el backend y el frontend, con consumo de APIs mediante HTTP para el acceso a datos externos.

Tipo de Actividad	Ponderación
Ejercicio práctico: creación de un proyecto ASP.NET Core MVC con vistas y modelos	0

Sesión No. 10, Unidad 7 - Web Development

Área Actitudinal (Saber ser)

Nombre del valor:
Descripción

Área de Conocimiento (Saber)

Competencia(s)	
Desarrolla aplicaciones web full-stack con el framework ASP.NET Core MVC y JavaScript, integrando vistas, modelos y templates con Razor en el backend y el frontend, con consumo de APIs mediante HTTP para el acceso a datos externos.	
Tema	Subtema
Web Development	Front End Development
Web Development	Programación del lado del cliente
Web Development	JavaScript
Templates con Razor	Layouts y herencia de templates
Templates con Razor	Archivos estáticos e inlines
Templates con Razor	ViewModels

Área de Habilidades (Saber Hacer)

Competencia	
Desarrolla aplicaciones web full-stack con el framework ASP.NET Core MVC y JavaScript, integrando vistas, modelos y templates con Razor en el backend y el frontend, con consumo de APIs mediante HTTP para el acceso a datos externos.	
Tipo de Actividad	Ponderación
Ejercicio práctico: desarrollo de templates con Razor y lógica de cliente en JavaScript	0

Sesión No. 11, Unidad 8 - Acceso a Datos Web

Área Actitudinal (Saber ser)

Nombre del valor:
Descripción

Área de Conocimiento (Saber)

Competencia(s)	
Desarrolla aplicaciones web full-stack con el framework ASP.NET Core MVC y JavaScript, integrando vistas, modelos y templates con Razor en el backend y el frontend, con consumo de APIs mediante HTTP para el acceso a datos externos.	
Tema	Subtema
Acceso a datos web	Internet como origen de datos
Acceso a datos web	Protocolo HTTP y códigos de error
Acceso a datos web	Peticiones HTTP en C# y librerías
Acceso a datos web	Lectura de archivos binarios mediante peticiones HTTP
Acceso a datos web	Análisis de XML y JSON en peticiones
Acceso a datos web	Interfaces de programación de aplicaciones (API)
Acceso a datos web	Seguridad y uso de API's

Área de Habilidades (Saber Hacer)

Competencia	
Desarrolla aplicaciones web full-stack con el framework ASP.NET Core MVC y JavaScript, integrando vistas, modelos y templates con Razor en el backend y el frontend, con consumo de APIs mediante HTTP para el acceso a datos externos.	
Tipo de Actividad	Ponderación
Ejercicio práctico: consumo de una API mediante peticiones HTTP en C#	0

Rúbrica de Evaluación

Cada una de las actividades del laboratorio (proyectos, prácticas, tareas y otras) cuenta con una rúbrica de evaluación específica, la cual está detallada en el documento que se entrega al estudiante al momento de asignar la actividad. Estas rúbricas describen los criterios de evaluación, niveles de desempeño esperados y

la ponderación correspondiente de cada aspecto evaluado.

Es **responsabilidad del estudiante** leer detenidamente la rúbrica asignada antes de iniciar el desarrollo de la actividad. Comprender los criterios de evaluación no solo permite orientar adecuadamente el trabajo, sino también mejorar el desempeño académico y fomentar la autorregulación del aprendizaje.

En caso de no recibir la rúbrica al momento de la asignación, el estudiante **debe solicitarla directamente al tutor académico**, ya que constituye una herramienta esencial para el cumplimiento de los objetivos de aprendizaje y la evaluación transparente.

Normativa Académica y Ética del Curso

En concordancia con el perfil del estudiante de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, se espera un alto nivel de compromiso con la excelencia académica y la ética profesional. Por ello, que se establece los siguientes lineamientos de carácter obligatorio que regulan el comportamiento académico del estudiante:

Plagio y copias

- Todo proyecto será sometido a verificación para confirmar su autoría y originalidad, con la finalidad de evitar cualquier plagio, copia o que la actividad no haya sido realizada por el estudiante.
- Cualquier evidencia de lo antes descrito en las distintas actividades será sancionada con una calificación de 0 (cero) y el caso será reportado al Docente quien a su vez informará a la Escuela de Ciencias y Sistemas para su seguimiento institucional.

Prórrogas y reposiciones

- No se otorgarán prórrogas para entregas de actividades.
- No se permitirá la reposición de proyectos bajo ninguna circunstancia.

Requisitos para evaluación final del curso

- Es obligatorio aprobar el laboratorio para tener derecho a la evaluación final del curso.
- La calificación de prácticas, proyectos y otras actividades que se indique será asignada de forma presencial, en la fecha y hora establecidas por el tutor académico.

Asistencia

- Para obtener la nota del laboratorio, se requiere un mínimo del 80% de asistencia a las sesiones de laboratorio.
- En caso de inasistencia, sólo se aceptarán justificaciones válidas respaldadas por constancia oficial.

Entregas

- No se aceptarán entregas tardías de tareas, prácticas, exámenes cortos, exámenes finales o proyectos sin justificación.

Medio oficial de entrega

- La plataforma UEDI de la Facultad será el único medio oficial para la entrega de actividades del curso.

Bibliografía

Deitel, P., & Deitel, H. (2017). C# 6 for Programmers (6.a ed.). Pearson.

Chacon, S., & Straub, B. (2014). Pro Git (2.a ed.). Apress.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms (3.a ed.). MIT Press.

Freeman, A. (2022). Pro ASP.NET Core MVC (8.a ed.). Apress.

E-Grafía

Microsoft. (2025). Documentación de C# y .NET. Microsoft Learn.

<https://learn.microsoft.com/es-es/dotnet/csharp/>

World Wide Web Consortium. (2025). HTML5 y CSS: especificaciones y guías. <https://www.w3.org/standards/>