

Taller de Lenguaje .NET**Carrera/ Plan:***Ingeniería en Computación Plan 2008 / Plan 2011 / Plan 2024***Año 2025****Año:** 5to**Régimen de Cursada:** Semestral**Carácter (Obligatoria/Optativa):** Optativa**Correlativas:** Taller de Lenguajes II - Programación III**Profesor/es:** Leonardo Corbalán**Hs. semanales:** 6 hs.**FUNDAMENTACIÓN**

La plataforma .NET proporciona un modelo de programación coherente e independiente del lenguaje para todas las capas o niveles de una aplicación. Se utiliza para la creación de todo tipo de aplicaciones: web, para dispositivos móviles, escritorio, juegos, IoT entre otras. Se admite en Windows, Linux y macOS. El lenguaje de programación C#, diseñado especialmente para la plataforma .NET, es un lenguaje moderno que se actualiza constantemente y se encuentra entre los más utilizados por la comunidad de desarrollo de software actual.

OBJETIVOS GENERALES

Profundizar los conocimientos obtenidos por el alumno en los primeros cursos vinculados con Algoritmos y Programación, permitiéndole desarrollar un estudio teórico-práctico de un lenguaje de programación soportado por la plataforma .NET, poniendo énfasis en el análisis formal de las características del lenguaje y su comparación con los que el alumno conociera a ese momento (típicamente Pascal).

CONTENIDOS MINIMOS (de acuerdo al Plan de Estudios)

Estudio de un lenguaje de programación en el que se desarrollen aplicaciones concretas. En lo posible la oferta de lenguajes será variable y actualizada con el cambio tecnológico.

PROGRAMA ANALÍTICO

Unidad 1: Plataforma .NET: Características. Common Language Runtime (CLR). Microsoft Intermediate Language (MSIL). Compilador Just-In-Time (JIT). Common Type System (CTS). Base Classes Library (BCL).

Unidad 2: Lenguaje C#: Características del lenguaje: Estructuras de control. Tipos básicos. Operadores. Ámbito de las variables. Métodos y parámetros. Sistema unificado de tipos: Tipos de valor y tipos de referencia. Conversiones de tipo explícitas e implícitas. Conversiones *boxing* y *unboxing*. Manejo de *strings*, arreglos y colecciones. Formatos compuestos e interpolación de strings. Manejo estructurado de excepciones.

Unidad 3: Programación orientada a objetos con C#: Clases y objetos. Campos y métodos de instancia. Constructores y finalizadores. Sobrecarga de métodos y de constructores. Miembros estáticos (de clase). Encapsulamiento. Propiedades e Indizadores. Herencia. Invalidación de métodos. Modificadores de acceso. Polimorfismo. Principio Open/Close. Interfaces. Implementación de múltiples

interfaces. Implementación explícita de miembros de interfaces. Uso de interfaces conocidas: IComparable, IComparer, IEnumerable e IEnumerator de la plataforma .NET. Iteradores.

Unidad 4: Conceptos avanzados del lenguaje C# y la plataforma .NET: Delegados. Métodos anónimos y expresiones lambda. Uso de delegados para implementar el pasaje de métodos por parámetro y la generación de eventos. Convenciones de nomenclatura para delegados y métodos involucrados en el lanzamiento y manejo de un evento. Métodos genéricos. Tipos genéricos (clases, interfaces y delegados genéricos). Programación asincrónica. Patrón asincrónico basado en Tareas. Métodos asincrónicos. Modificador async y operador await

Unidad 5. Persistencia de datos: Introducción a Entity Framework Core. LINQ. Operaciones CRUD sobre base de datos relacionales.

Unidad 6. Aplicaciones Web: Introducción al desarrollo de aplicaciones web con ASP.NET Core. Inyección de dependencias. Uso de contenedores DI. Desarrollo de aplicación web de página única (SPA) con ASP.NET Core Blazor. Componentes Razor. Conceptos de Arquitectura Limpia.

BIBLIOGRAFÍA

No se utiliza bibliografía obligatoria. Los contenidos publicados por la cátedra en la plataforma IDEAS cubren los requerimientos del curso. Sin embargo, se aconseja adquirir el hábito de consultar la documentación de Microsoft para usuarios finales, desarrolladores y profesionales de TI (<https://docs.microsoft.com/es-es/>) en relación con la plataforma .NET y al lenguaje C#.

BIBLIOGRAFÍA COMPLEMENTARIA

- Professional C# 7 and .NET Core 2.0, Christian Nagel, Wrox; Edición: 7. (2018).
- Illustrated C# 7: The C# Language Presented Clearly, Concisely, and Visually. Daniel M. Solis. Apress 5th ed. (2018).
- Dependency Injection in .NET Core 2.0, Marino Posadas y Tadish Dash. Packt Publishing (2017)
- ASP.NET Core in Action. Andrew Lock. Manning Publications; Edición: 1st (2018)
- Microsoft Blazor Building Web Applications in .NET. Second Edition. Peter Himschoot
- Microsoft documentation for end users, developers, and IT professionals. <https://docs.microsoft.com>

METODOLOGÍA DE ENSEÑANZA

La asignatura se organiza en clases teóricas, prácticas y de explicaciones de práctica. Estas actividades se desarrollan íntegramente en la sala de PC. La razón de ello es maximizar la interacción del alumno con el lenguaje, la plataforma y el ambiente de desarrollo, aún en las clases teóricas donde abundan ejercicios de codificación propuestos a los estudiantes con la intención de discutir y asimilar conceptos teóricos a partir de los resultados obtenidos. Durante las consultas de práctica, además de atender las dudas individuales de los alumnos, se ofrecen explicaciones generales sobre algunos de los ejercicios más representativos.

El material del curso se compone de 13 clases teóricas y 11 trabajos prácticos no entregables. Cada uno de los ejercicios publicados en los trabajos prácticos incluye algún concepto concreto que la cátedra pretende ilustrar. En consecuencia, es sumamente importante que los alumnos resuelvan todos los ejercicios y participen activamente en las consultas de práctica para despejar cualquier duda que les pueda sobrevenir.

Durante el desarrollo del curso, se plantea la realización de un trabajo grupal de desarrollo que constará de dos entregas: una parcial y una final. Para ambas entregas, es requisito que se acompañe un informe

escrito que detalle los aspectos relevantes del trabajo. Se requiere que todos los integrantes del grupo conozcan los detalles del trabajo, los cuales serán evaluados en un coloquio individual hacia el final del curso. Una vez concluidas las clases teóricas, el horario reservado para las mismas es utilizado para consulta y para la realización de los coloquios.

Se utiliza la plataforma IDEAS para la publicación del material (contenido teórico, trabajos prácticos y trabajos de programación obligatorios) y la comunicación con los alumnos a través de la mensajería, cartelera de novedades y foro.

Aunque la orientación del curso es predominantemente práctica, no se desatienden aspectos teóricos importantes que el alumno debe conocer para comprender con claridad los conceptos inherentes a un lenguaje orientado a objetos como es el caso de C#.

El curso promueve el trabajo constante y la participación de los alumnos en clase. A partir de la segunda teoría, una vez concluidas las actividades relacionadas con los nuevos conceptos presentados, se discuten con la intervención activa de los estudiantes, los detalles más significativos de la práctica resuelta la semana anterior. Por lo tanto, es importante que los alumnos trabajen todas las semanas llevando al día la realización de las prácticas para así poder aprovechar esta instancia de fijación de conceptos.

Completan estas instancias de fijación y clarificación de conceptos una serie de autoevaluaciones realizadas a lo largo del curso. Estas autoevaluaciones son de carácter formativo, no se utilizan para examinar a los alumnos sino para que ellos mismos puedan valorar de qué forma están transitando el proceso de aprendizaje. Una autoevaluación consiste en una serie de ejercicios con preguntas y opciones de respuestas presentados a la clase de a uno por vez utilizando un proyector multimedia. Concluido el tiempo otorgado en cada ejercicio para que el alumno piense su respuesta se señala la opción correcta. Durante la ejecución de la prueba los estudiantes calculan su propio puntaje, información mantenida sólo para sí. Este espacio es utilizado también por la cátedra para despejar dudas y clarificar conceptos relacionados con los ejercicios presentados.

EVALUACIÓN

Son requisitos para aprobar la cursada:

- Cumplir con las entregas parciales que se soliciten y aprobar con nota mayor o igual a 6 (seis) la entrega final del trabajo de desarrollo integrador.
- Aprobar un coloquio final.

El trabajo integrador puede presentarse de manera individual o grupal. El coloquio es individual. La entrega final del trabajo integrador y el coloquio contarán con una instancia de recuperación en caso de ser necesario.

Régimen de promoción: El alumno que apruebe la cursada también obtendrá la promoción. La calificación final se definirá en función del resultado del trabajo integrador y del coloquio.

CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Fecha	Contenidos/Actividades
-------	-------	------------------------

1	Semana del 09/03/2026	Teoría 1. Generalidades de la plataforma .NET. Introducción al lenguaje C#. Características. Tipos integrados. Variables, constantes, literales, operadores. Espacios de nombres. Estructuras de control.
2	Semana del 16/03/2026	Teoría 2. Sistema unificado de tipos. Tipos de valor y tipos de referencia. Conversiones de tipo explícitas e implícitas, <i>boxing</i> y <i>unboxing</i> . Métodos y parámetros. Pasaje de parámetros por la línea de comandos.
3	Semana del 30/03/2026	Teoría 3. Inferencia de tipos. Tipos anónimos. Tipo Dynamic. Formatos compuestos e interpolación de strings. Arreglos. Colecciones. Manejo de Excepciones.
4	Semana del 06/04/2026	Teoría 4. Conceptos introductorios de programación orientada a objetos. Clases. Miembros de instancia. Constructores. Sobrecarga de constructores y de métodos.
5	Semana del 13/04/2026	Teoría 5. Miembros estáticos. Encapsulamiento. Control de acceso a la representación interna de los objetos por medio de propiedades e indizadores.
6	Semana del 20/04/2026	Teoría 6. Herencia. Especialización de clases. Invalidación de métodos. Modificadores de acceso: public, protected, private, e internal. Polimorfismo. Principio Open/Close
7	Semana del 27/04/2026	Teoría 7. Interfaces. Implementación de múltiples interfaces. Implementación explícita de miembros de interfaces. Principio de inversión de dependencias (DIP) y Patrón de Inyección de Dependencias. Conceptos de Arquitectura Limpia.
8	Semana del 04/05/2026	Teoría 8. Uso de las interfaces IComparable, IComparer, IEnumerable e IEnumerator de la plataforma .NET. Iteradores. Delegados. Pasaje de métodos como parámetro. Métodos anónimos. Expresiones lambda.
9	Semana del 11/05/2026	Teoría 9. Métodos Genéricos. Tipos genéricos (Clases, interfaces y delegados).
10	Semana del 18/05/2026	Teoría 10. Persistencia de datos. SQLite. LINQ. ORM Entity Framework Core
11 y 12	Semanas del 01/06/2026 y 08/06/2026	Teoría 11 y 12. Introducción al desarrollo web en .NET: Evolución y arquitectura de las aplicaciones web modernas. Interfaz de Usuario con ASP.NET Core Blazor. Modelos de hospedaje (Blazor Server y Blazor WebAssembly). Introducción a la exposición y consumo de datos mediante servicios Web API. Integración de conceptos: Aplicación de los principios de Arquitectura Limpia, persistencia con Entity Framework Core e Inyección de Dependencias en el entorno web.
		El resto de las clases se utiliza para consulta, corrección y devolución de trabajos y para la toma de coloquios.

Evaluaciones previstas	Fecha
Entrega final del trabajo integrador	20/06/2026
Coloquios	A partir del 29/06/2026 se asignarán turnos para rendir coloquio en los horarios de práctica y teoría

CONTACTO DE LA CÁTEDRA:

Mail: slpnet@lidi.info.unlp.edu.ar

Plataforma IDEAS (<https://ideas.info.unlp.edu.ar>) **Curso:** "Seminario de Lenguajes - Opción .NET 1er. Semestre 2026"

Firma del/los profesor/es



Leonardo Corbalán
Profesor Adjunto