

**SEMINARIO DE LENGUAJES
(OPCION .NET)**

Año 2019

Carrera/ Plan:

Licenciatura en Informática Plan 2015/Plan 2012/Plan 2003-07
Licenciatura en Sistemas Plan 2015/Plan 2012/Plan 2003-07
Analista Programador Universitario Plan 2015/Plan 2007
Analista en TIC Plan 2017

Año: 2°**Régimen de Cursada:** Semestral**Carácter:** Electiva**Correlativas:** Taller de Programación (Plan 2015 o 2017) o
Algoritmos, Datos y Programas**Profesor/es:** Leonardo Corbalan**Hs. semanales:** 6**FUNDAMENTACIÓN**

La plataforma .NET proporciona un modelo de programación coherente e independiente del lenguaje para todas las capas o niveles de una aplicación. El lenguaje de programación C#, diseñado especialmente para la plataforma .NET, 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).

COMPETENCIAS

- CGS2- Comunicarse con efectividad en forma oral y escrita.
- CGS4- Aprender en forma continua y autónoma, con capacidad de planificar este aprendizaje.
- CGS6- Capacidad para interpretar la evolución de la Informática con una visión de las tendencias tecnológicas futuras.
- CGT1- Identificar, formular y resolver problemas de Informática.
- CGT5- Utilizar de manera efectiva las técnicas y herramientas de aplicación de la Informática.
- LI- CE4 – Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real, especificación formal de los mismos, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software/sistemas de información que se ejecuten sobre equipos de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.
- LS- CE1 – Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real. Especificación formal, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software que se ejecuten sobre sistemas de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.

CONTENIDOS MINIMOS (de acuerdo al Plan de Estudios)

Durante el desarrollo del curso se estudian las características generales de la plataforma .NET y del lenguaje de programación C# para desarrollar aplicaciones de escritorio con interfaces de usuario en modo texto (consola) y modo gráfico (formulario Windows). También se adquieren las habilidades necesarias en el manejo del SharpDevelop, un completo entorno de desarrollo integrado (IDE) open source para la plataforma .NET.

PROGRAMA ANALÍTICO

Unidad 1: Conceptos básicos sobre la plataforma .Net y el lenguaje C#

Introducción a la Plataforma .NET: Características. Common Language Runtime. Microsoft Intermediate Language. Compilador Just-In-Time. Common Type System (CTS). Base Classes Library.

Introducción al lenguaje C#: Características del lenguaje. Estructuras de control. Sistema de tipos. Conversiones de tipos. Operadores. Ámbito de las variables. Métodos y parámetros. Excepciones. Manejo de strings, arreglos y colecciones. Utilización de un entorno de desarrollo integrado open source.

Unidad 2: Conceptos básicos de programación orientada a objetos con C#. Aplicaciones de consola

Conceptos introductorios a la programación orientada a objetos. Clases. Ocultación. Definición de clases. Creación de objetos. Campos y métodos. Sobrecarga de métodos. Constructores y destructores. Modificadores de acceso. Herencia. Redefinición de métodos. Concepto de polimorfismo. Propiedades e Índices. Miembros estáticos (de clase), diferencia entre miembros estáticos y de instancia. Delegados. Pasaje de métodos como parámetro. Utilización de delegados como mecanismo para implementar eventos. Convenciones de nomenclatura para delegados y métodos involucrados en el lanzamiento y manejo de un evento. Eventos.

Unidad 3: Aplicaciones Windows (interfaz gráfica)

Introducción al desarrollo de aplicaciones gráficas basadas en formularios Windows. Controles clásicos, acceso a sus propiedades y utilización de sus principales eventos. Contenedores. Propiedad Controls. Creación de formularios, incorporación, manipulación y eliminación de controles por código. Derivación de controles. Cuadros de diálogos, utilización de método ShowDialog() y propiedad DialogResult.

Unidad 4: ADO.NET. Persistencia de datos con XML

Conexión de aplicaciones con orígenes de datos. ActiveX Data Objects (ADO.NET) Clases, propiedades y métodos más importantes. Visualización en formulario Windows, controles DataGridView y BindingSource. Relación Maestro/Detalle. Filtrado y ordenamiento de filas. Persistencia de datos. Introducción a XML, elementos y atributo, sintaxis, XML bien formado y XML válido. Introducción a XSD. Persistencia de objetos DataTable y DataSet en archivos XML.

BIBLIOGRAFÍA

No se utiliza bibliografía obligatoria. Los contenidos publicados por la cátedra en la plataforma IDEAS cubren las necesidades surgidas de las actividades del curso. Sin embargo se aconseja adquirir el hábito de consultar el material de referencia publicado en el sitio web MSDN library (url: <http://msdn.microsoft.com/es-es/library/ms123401.aspx>) en relación a la plataforma .NET y al lenguaje C#.

BIBLIOGRAFÍA COMPLEMENTARIA

- Illustrated C# 2010, Daniel M. Solis. Apress 2010
- .NET Framework Essentials, Thuan L. Thai, Hoang Q. Lam, O'Reilly, 2003.
- Como Programar en C#, H. Deitel, Pearson. Prentice Hall, Segunda Edición, 2007.
- Dissecting a C# Application Inside SharpDevelop, C. Holm, M. Krüger, B. Spuida, APress, 2004.
- C# al Descubierto, Joseph Mayo, ed. Prentice Hall, ISBN 84-205-3477-3
- C# Essentials. Beb Albahari, Peter Drayton y Brand Merrill, ed. O'Reilly, ISBN 0596003153
- Inside C#, Tom Archer, ed. Microsoft Press, ISBN 0735616485
- Learning XML, Second Edition, E. Ray, O'Reilly, 2003
- Extensible Markup Language (XML) <http://www.w3.org/XML/>

METODOLOGÍA DE ENSEÑANZA

La asignatura se organiza en clases teóricas y prácticas (una teoría y una consulta de práctica por semana). Tanto las teorías como las prácticas 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 12 clases teóricas, 11 trabajos prácticos (uno por cada teoría a excepción de la última) y varios trabajos de programación obligatorios que deben presentarse acompañados de un informe escrito y expuestos oralmente en un coloquio hacia el final del curso. Una vez concluidas las clases teóricas, el horario reservado para las mismas es utilizado para consulta y para la toma de exámenes.

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.

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 y la cartelera de novedades.

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 (detalladas en "CRONOGRAMA DE CLASES Y EVALUACIONES"). 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.

A continuación se ofrecen precisiones sobre la forma en que se alcanzan y evalúan las competencias enunciadas previamente:

CGS2- Comunicarse con efectividad en forma oral y escrita

El informe escrito que los alumnos deben presentar junto con los ejercicios de programación obligatorios son utilizados para evaluar el contenido técnico, la estructura, organización, sintaxis y claridad conceptual. Además, estos trabajos deben exponerse en un coloquio ante la cátedra que evalúa los conocimientos

adquiridos, forma de presentación y expresión de los alumnos. Los resultados se reflejan en planillas escritas que conforman documentación de evaluación del coloquio.

CGS4- Aprender en forma continua y autónoma, con capacidad de planificar este aprendizaje

Se incluyen actividades planificadas para los alumnos proponiendo “desafíos” que deben convertirse en “ideas proyecto” y posteriormente en potenciales desarrollos del alumno. El objetivo es que el alumno logre abstraer una serie de pasos que respondan a una metodología clásica de investigación y lo ayuden a formarse en esta competencia:

- Búsqueda de bibliografía actualizada sobre el tema.
- Abstracción del desafío/problema como una “idea proyecto a resolver”.
- Expresión sintética de la especificación del proyecto, con recursos humanos requeridos y plan de tareas.
- Implementación y defensa oral/escrita de la solución al desafío.

La cátedra acompaña el proceso del alumno, para consolidar sus habilidades para esta competencia. La evaluación de esta competencia se refleja en una planilla detallada, donde se indica la capacidad del alumno para desarrollar su aprendizaje y la formulación de la solución al desafío en forma autónoma.

CGS6- Capacidad para interpretar la evolución de la Informática con una visión de las tendencias tecnológicas futuras

Se plantean actividades planificadas para los alumnos proponiendo el estudio de la tecnología existente y prevista para un tipo de problema y se los “desafía” a presentar la posible evolución de la solución para ese tipo de problema y en QUE podría mejorarse la solución/soluciones actuales.

Esto lleva al alumno a buscar bibliografía relacionada con el cambio tecnológico y formarse un criterio sobre las tendencias (por ejemplo, en los procesadores a utilizar, el tipo de topología de red o la migración de aplicaciones a móviles o el modo de extraer conocimiento de grandes volúmenes de datos).

La cátedra acompaña el proceso del alumno, para contrastar las conclusiones del alumno y validar su habilidad para esta competencia. La evaluación de esta competencia se refleja en una planilla detallada, donde se indican los resultados del estudio bibliográfico del alumno y su capacidad para formular las ventajas potenciales del cambio tecnológico en el problema/área de conocimiento planteada.

CGT1- Identificar, formular y resolver problemas de Informática.

En la cátedra se pone énfasis en el proceso de identificación de problemas del mundo real, especificación de los mismos como problemas resolubles desde la informática y en el desarrollo de soluciones verificables para los mismos.

La evaluación de esta competencia se realiza por medio de las evaluaciones de los informes escritos, trabajos de programación obligatorios, coloquios y examen final de la asignatura y se refleja en la corrección de las pruebas escritas del alumno.

CGT5- Utilizar de manera efectiva las técnicas y herramientas de aplicación de la Informática

En la cátedra se pone énfasis en la capacidad del alumno para conocer técnicas y herramientas de aplicación en Informática (en lo posible siguiendo las tendencias marcadas por el cambio tecnológico) y en la aplicación efectiva de las mismas.

La cátedra acompaña el proceso con materiales para que el alumno estudie casos y valore la selección y empleo eficiente de herramientas y técnicas determinadas para cada problema.

La evaluación de esta competencia se realiza por medio de las evaluaciones de los informes escritos, trabajos de programación obligatorios, coloquios y examen final de la asignatura y se refleja en la corrección de las pruebas escritas del alumno.

EVALUACIÓN

Para aprobar la cursada el alumno deberá:

- Rendir un examen y obtener una calificación mayor o igual a 6 (seis). En caso de desaprobado se tomarán hasta 2 recuperatorios.
- Aprobar el informe escrito y el coloquio sobre los trabajos de programación obligatorios. En caso de desaprobado, el alumno contará con una fecha más de recuperatorio.

Régimen de promoción:

- El alumno que apruebe la cursada también obtendrá la promoción.

CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Fecha	Contenidos/Actividades
1	08/03/2019	Teoría 1. Generalidades de la plataforma .NET. Common Language Runtime (CLR). Microsoft Intermediate Language (MSIL). Compilador Just-In-Time (JIT). Common Type System (CTS). Base Classes Library (BCL). Introducción al lenguaje C#. Características. Generalidades del sistema de tipos. Constantes y variables. Conversiones de tipos implícitas y explícitas. Operadores. Espacios de nombres. Estructuras de control. Ámbito de las variables. Compilación línea de comandos. Presentación del entorno de desarrollo open source SharpDevelop.
2	15/03/2019	Teoría 2. Sistema de tipos. Diferencias entre tipos valor y tipos referencia. Pila de ejecución y memoria Heap. Conversiones boxing y unboxing. Arreglos. Clases String y StringBuilder. Conversiones de tipo. Operadores de conversión explícita. Conversiones con clases auxiliares. Tipos enumerativos. Métodos. Parámetros por valor, por referencia y de salida. Repaso de puntos claves de la práctica 1
3	22/03/2019	Teoría 3. String - Formatos compuestos. Arreglos de varias dimensiones. Arreglos de arreglos. Colecciones: Espacio de nombres System.Collection, clases ArrayList, BitArray, Stack, Queue y HashTable. Manejo de Excepciones: sentencia try, propagación de excepciones. Repaso de puntos claves de la práctica 2. Autoevaluación sobre teorías y prácticas 1 y 2.
4	29/03/2019	Teoría 4. Conceptos introductorios de programación orientada a objetos. Clases, características y comportamiento. Ocultación. Definición de clases en C#. Creación de objetos (operador new). Referencia this. Miembros de una clase: campos y métodos. Sobrecarga de métodos. Constructores. Sobrecarga de constructores. Repaso de puntos claves de la práctica 3.
5	05/04/2019	Teoría 5. Herencia. Especialización de clases. Redefinición de métodos.

		Concepto de polimorfismo. Destruyores. Referencia base. Operador <code>is</code> , utilización junto con el operador <code>as</code> . Modificadores de acceso: <code>public</code> , <code>protected</code> , <code>private</code> , e <code>internal</code> . Repaso de puntos claves de la práctica 4.
6	12/04/2019	Teoría 6. Propiedades: control de acceso a campos privados por medio de propiedades. Propiedades de sólo lectura, de sólo escritura y de lectura/escritura. Indizadores. Control de acceso a la representación interna de los objetos por medio de indizadores. Indizadores de sólo lectura, sólo escritura y lectura/escritura. Miembros estáticos (de clase), diferencia entre miembros estáticos y de instancia. Repaso de puntos claves de la práctica 5. Autoevaluación sobre teorías y prácticas 1 a 5.
7	26/04/2019	Teoría 7. Delegados: concepto. Utilización de delegados para implementar el pasaje de métodos como parámetro. Utilización de delegados como mecanismo para implementar eventos. Convenciones de nomenclatura para delegados y métodos involucrados en el lanzamiento y manejo de un evento. Parámetro <code>sender</code> de tipo <code>object</code> y parámetro <code>e</code> de tipo <code>EventArgs</code> . Repaso de puntos claves de la práctica 6.
8	03/05/2019	Teoría 8. Eventos: Control de acceso a los delegados por medio de eventos (construcción sintáctica <code>Event</code>). Construcción <code>add</code> y <code>remove</code> . Operadores <code>+=</code> y <code>-=</code> para el alta y baja de las suscripciones a eventos. Introducción al desarrollo de aplicaciones gráficas basadas en formularios Windows (Winform). Diseñador de formularios del IDE SharpDevelop. Utilización de los principales controles gráficos, acceso a sus propiedades y utilización de sus principales eventos. Repaso de puntos claves de la práctica 7.
9	10/05/2019	Teoría 9. Formularios Windows (Continuación). Análisis de código generado por el IDE SharpDevelop. Contenedores: Propiedad <code>Controls</code> . Creación de formularios, incorporación, manipulación y eliminación de controles por código. Derivación de controles (controles especializados). Cuadros de diálogos. Método <code>ShowDialog()</code> y propiedad <code>DialogResult</code> . Implementación de una calculadora, parámetro <code>sender</code> para manejo de múltiples eventos con un único método manejador. Repaso de puntos claves de la práctica 8. Autoevaluación sobre teorías y prácticas 1 a 8.
10	17/05/2019	Teoría 10. Conexión de aplicaciones con orígenes de datos. ActiveX Data Objects (ADO.NET) Espacio de nombres <code>System.Data</code> . Utilización de las clases <code>DataTable</code> , <code>DataRow</code> , <code> DataColumn</code> , <code>DataSet</code> y <code>DataRelation</code> . Sus métodos y propiedades más importantes. Visualización en formulario Windows, control <code>DataGridView</code> y <code>BindingSource</code> . Relación Maestro/Detalle entre tablas. Filtrado y ordenamiento de filas. Repaso de puntos claves de la práctica 9.

11	24/05/2019	Teoría 11. Persistencia de datos. Introducción a XML, elementos y atributo, sintaxis, XML bien formado, XML válido (esquema). Introducción a XSD. Persistencia de objetos <code>DataTable</code> y <code>DataSet</code> en archivos XML con y sin información de esquema. Repaso de puntos claves de la práctica 10
12	31/05/2019	Teoría 12. Repaso de puntos claves de la práctica 11. Autoevaluación general.

Evaluaciones previstas	Fecha
Examen escrito (1ra. fecha)	14/06/2019
Examen escrito (2da. fecha)	21/06/2019
Examen escrito (3ra. fecha)	28/06/2019
Coloquios	A partir del 10/06/2019 en los horarios de práctica

CONTACTO DE LA CÁTEDRA:

Plataforma IDEAS (<https://ideas.info.unlp.edu.ar>) **Curso:** "Seminario de Lenguajes - Opción .NET 1er. Semestre 2019"

Firma del/los profesor/es