



# T10922 Lenguajes de Última Generación - programa

Lenguajes De Última Generación (Universidad Abierta Interamericana)

## Programa de Asignatura

00 Código:	F	C	Asign.
	T	109	22
	T	108	22

**01. Facultad:** Tecnología Informática /Carrera: Ingeniería en Sistemas Informáticos

**02. Asignatura:** T109 – 22 - LENGUAJES DE ÚLTIMA GENERACIÓN  
T108 – 22 - LENGUAJES DE ÚLTIMA GENERACIÓN

**03. Año lectivo:** 2021

**04. Año de Cursada:** 2°

**05. Cuatrimestre:** 2°

**06. Hs. Semanales:** 6

**07.**

<b>Profesor Titular:</b> Vilaboa, Pablo	
<b>Buenos Aires</b>	<b>Rosario</b>
<b>Titular a cargo:</b> Vilaboa, Pablo	<b>Asociado a cargo:</b> Dorigo, Gastón
<b>Adjuntos:</b> Cardacci Dario Ghigliani Leonardo Martinez Jorge Prinzo Mauricio Parkinson Daniel Roda Jorge	<b>Auxiliar:</b> García Gustavo
<b>Eje socioprofesional: Los Modelos como representación abstracta de la realidad (Análisis y Lenguajes)</b>	
<b>Coordinador de eje:</b> Ing. Vilaboa, Pablo	<b>Coordinador de eje:</b> Ing. Vaquero, Marcelo
<b>Eje Epistémico: Teoría de Sistemas y Lenguajes de Programación.</b>	

### 08. Fundamentación:

Los sistemas de información administran grandes volúmenes de datos y deben generar procesos eficientes. Estos datos deben persistir en el tiempo de manera íntegra y los sistemas son los responsables de depositarlos y obtenerlos cuando hace falta. Esta asignatura aborda los problemas con los que nos enfrentamos y las soluciones a aplicar para llevar a cabo esa tarea.

También trabaja los aspectos necesarios para realizar el acceso a datos desde entornos centralizados y distribuidos.

Además, aporta técnicas de programación avanzadas que permiten de manera muy eficiente capturar los datos con los que se va a trabajar, validarlos y manipularlos.

### **09 Ítems del perfil que se desarrollarán:**

Los administradores de los sistemas administrativos demandan un perfil profesional que permita manejar una gran cantidad de información, lo que implica una especialización en el almacenamiento de la información sobre diferentes soportes, grandes bancos de datos, software específico para el procesamiento de información y relaciones de datos.

Por otra parte, dominar la utilización de herramientas y técnicas de programación avanzadas para el desarrollo, permite manejar nuevos niveles de abstracción, lo cual le otorga al futuro profesional la posibilidad de satisfacer las demandas de la sociedad, las que deben ser desarrolladas con pericia y practicidad.

El conocer cómo integrar los componentes funcionales para generar soluciones computacionales es una necesidad actual de las organizaciones, por lo cual esta asignatura le brinda al futuro graduado la posibilidad de incorporar los conocimientos necesarios para satisfacer estas consideraciones.

En consecuencia, esta asignatura aporta al desarrollo de las siguientes competencias definidas en el perfil de Ingeniero en Sistemas Informáticos de la UAI:

- 1) Competencia para identificar y resolver problemas.
- 2) Competencia para desarrollar proyectos (sistemas, componentes, productos o procesos) en distintos ámbitos de aplicación.
- 3) Competencia para utilizar de manera efectiva las técnicas y herramientas de aplicación.
- 4) Competencia para contribuir a la generación de desarrollos tecnológicos.
- 5) Competencia para desempeñarse de manera efectiva en espacios de intercambio.
- 6) Competencia para comunicarse con efectividad.
- 7) Competencia para aprender en forma continua y autónoma.

**10. Correlativas previas:** Programación Orientada a Objetos.

**Posteriores:** Trabajo de Campo I y Lenguajes de Programación para la Administración

### **11. Articulación con asignaturas:**

Articulación Vertical:

Las asignaturas Programación Estructurada y Programación I integran los contenidos sobre el concepto de programa, la introducción a la programación, las estructuras de datos y los algoritmos más importantes para el ordenamiento, clasificación y búsqueda de datos, los que son utilizados por esta asignatura.

Metodología de Desarrollo de Sistemas I y Programación Orientada a Objetos proveen a Lenguajes de Última Generación lo necesario para que se puedan trabajar los aspectos referidos a la documentación, el uso y la forma de desarrollar software.

Esta asignatura le provee a Trabajo de Campo I y Trabajo de Diploma las técnicas de programación avanzadas para el óptimo manejo de datos desde las aplicaciones, la validación de los mismos, el almacenamiento y su posterior recuperación.

#### Articulación Horizontal:

La articulación con Metodología de Desarrollo de Sistemas II le da una visión práctica y teórica sobre la forma de analizar, diseñar y modelar soluciones orientadas a objetos de una manera iterativa e incremental, lo cual facilita la tarea de desarrollo realizadas en esta asignatura.

#### 1. Objetivos:

Que los alumnos logren:

- Construir programas orientados a objetos con interfaz de aplicación visual para acceder y procesar datos existentes en una base de datos.
- Elaborar hipótesis de indagación sobre el desarrollo de sistemas de información.
- Asociar los conocimientos de metodologías de construcción de software vistas con anterioridad, a la metodología orientada a objetos.
- Discriminar procesos de desarrollo de acuerdo con el problema a solucionar.
- Elaborar componentes de software.
- Crear programas de instalación y distribución de aplicaciones.
- Resolver situaciones complejas a partir de la simplificación que otorga la aplicación de tecnología y desarrollar los mecanismos que permitan su transferencia al ámbito social.
- Comprender los paradigmas de las nuevas plataformas de desarrollo de software.

#### 13. Unidades de desarrollo de los contenidos:

##### Unidad 1: ENSAMBLADOS.

Módulos. Encabezados. Metadatos. El proceso de enlace. Ensamblados. El Manifiesto. Ensamblados privados y compartidos. Nombres seguros. Contenedores de claves y de firmas parciales. Directivas de versiones en archivos de configuración de aplicaciones. Ensamblados previamente cargados y búsquedas en el GAC. Versión del runtime. El vinculador de ensamblado (AL). La utilidad de nombre seguro (SN). La utilidad Caché de ensamblados global (GACUTIL). El desensamblador MSIL (ILDASM).

Tiempo: 18 horas

##### Unidad 2: ARQUITECTURAS

Definición de arquitectura, Elección de la arquitectura en el desarrollo de sistemas de información. Arquitectura lógica del SI. Desarrollo en 3 capas. Desarrollo en N capas. Patrones aplicables a la solución y modelado de Arquitecturas. Framework de persistencia. Sincronización con el diagrama de clases. Mapeo de objetos. Exportación y acceso a la base de datos

Tiempo: 24 horas

### **Unidad 3: ACCESO A DATOS – MODO CONECTADO Y DESCONECTADO**

Constructor de comandos SQL para la actualización de la base de datos. Utilización de los contadores de performance. Administración del log de eventos. Servicios del sistema. Introducción a las interfaces. Almacenamiento de fragmentos en caché. Introducción al acceso a datos conectado y desconectado. Proveedores de datos. Introducción a un modelo de acceso a datos orientado a objetos. El objeto Connection. Configuración delConnectionString. Apertura y cierre de la conexión. Manejo de transacciones. El objeto Command. Lectura de datos. Manejo de parámetros y procedimientos almacenados. El objeto DataReader. El objeto DataSet. Construcción de un DataSet. La clase DataAdapter. Lectura de datos de una base de datos. Actualización de la base de datos. Lectura y escritura de archivo de texto. System.IO. File y Stream. Lectura y escritura de archivos XML. La clase XmlTextReader. La clase XmlTextWriter. Manejo del DOM de XML. La clase XmlNode. La clase XmlDocument. LinqTOXML. Trabajando con elementos. Manipulando Objetos en Memoria

Tiempo: 24 horas

### **Unidad 4: CONTROLES DE USUARIO Y REPORTES.**

Controles de usuario. Controles personalizados. Mejora de controles personalizados. Controles compuestos. Expresiones regulares. El lenguaje de las expresiones regulares. Opciones de las expresiones regulares. Clases de expresiones regulares. La clase Regex. Las clases MatchCollection y Match. La clase Group. Las clases CaptureCollection y Capture. Expresiones regulares en funcionamiento. Herramientas para generar reportes. Proyectos de reportes. Diseño de un reporte. Estructura del reporte. Sincronización de datos en el reporte.

Tiempo: 12 horas

### **Unidad 5: PROGRAMACIÓN GRÁFICA GDI+**

Gráficos vectoriales en 2-D. Objetos gráficos. Líneas, rectángulos, polígonos, elipses y arcos. Splines de Bézier y Cardinales. El objeto Pen. Formas con relleno. Pinceles. Regiones. Combinaciones alfa. Transformaciones. Imaging. Carga y almacenamiento de imágenes. Tratamiento de una imagen. Mapas de bits transparentes y semitransparentes. Iconos. Metarchivos. Tipografía. Familias de fuentes. Dibujo de texto. Texto alineado. Variaciones textuales. Anti-Aliasing.

Tiempo: 12 horas

### **Unidad 6: ESTRATEGIAS PARA EL DISEÑO DE ALGORITMOS**

Estrategias de diseño de algoritmos. Comparación de algoritmos. Principio de Invarianza. Eficiencia, legibilidad y reusabilidad. Cálculo de la eficiencia de un

algoritmo. Algoritmos secuenciales, paralelos y distribuidos. Verificación y Depuración de Algoritmos.

Tiempo: 12 horas

#### **14. Metodología de trabajo:**

La dinámica de las clases está orientada a vincular los conceptos teóricos con la práctica. Por tal motivo el desarrollo de los conceptos teóricos y metodológicos está acompañado de la resolución de casos prácticos en los laboratorios de computación. Se utilizarán métodos pedagógicos activos, que prevén un papel protagónico de los alumnos, en lugar de limitarlos al de meros espectadores pasivos de la enseñanza. Ello da lugar a que un alto espacio del tiempo total de la clase se dedique a actividades que incluyan la participación de los alumnos.

Los métodos pedagógicos a aplicar serán los siguientes:

Clases teóricas, explicativas y dialogadas, para la internalización de conceptos nuevos, revisión de temas (apoyo) de la materia correlativa y para la profundización de los mismos.

Clases prácticas: la realización de trabajos prácticos de indagación grupales con exposición y debate, en las fechas indicadas, constituirá la labor indispensable para lograr los objetivos propuestos. Lecturas de textos. Clases especiales a cargo de los alumnos. Prácticos en laboratorio y la confección de un plan de desarrollo tecnológico que integre los conocimientos previos.

Los acuerdos metodológicos de evaluación centran la atención en la profundización de indagación, análisis y procesamiento de la información, la transferencia de contenidos trabajados en sus producciones. Como capacidad de detectar y solucionar problemas tecnológicos.

El docente cumplirá el rol de guía, conductor de la actividad.

Se trata de que el curso se vuelva un verdadero taller de trabajo, para estimular la cooperación, la participación y la responsabilidad de resolver las tareas asignadas. También se trabajará con audiovisuales, se realizarán análisis de lecturas especializadas, salidas didácticas y la participación de invitados especiales.

La constitución de la clase fijará el andamiaje cognitivo del alumno sostenido en el conocimiento teórico y práctico.

El tiempo disponible se dividirá en 4 espacios:

Presentación del tema (10%): El docente expondrá una introducción para enmarcar el dominio del tema a tratar preparando al alumno en su camino de aprendizaje.

Desarrollo (20%): Exposición docente de los conceptos importantes remarcando las ideas fuerza valorizando la integración de los conocimientos nuevos con la experiencia previa del alumno.

Práctica: 50%. Para afianzar los conocimientos nuevos el docente guiará el aprendizaje del alumno fortaleciendo el dialogo con ejercitación por medios de guías de trabajo practico y abordajes bibliográficos.

Reflexión: 20%. Se fomentará el debate individual y grupal para asegurar la asimilación de los conocimientos nuevos por el alumno.

Se utilizarán PC preparadas para el desarrollo de software y un entorno de desarrollo integrado (IDE de desarrollo) a tal fin. También se utilizarán base de datos relacionales. Los textos de actualidad y las notas periodísticas sobre tecnología de punta serán propuestos para ser leídos y tratados en clase.

Se fomentará la participación de los alumnos en la formulación de soluciones a los problemas de la práctica profesional.

Se desarrollarán ejercicios que denoten un alto grado de relación grupal para aprovechar la sinergia de la relación interpersonal.

Herramientas:

Lenguaje de programación visual orientado a objetos.

Laboratorios.

- Lecciones
- Preguntas de auto evaluación
- Ejercicios resueltos
- Ejercicios para resolver
- Presentaciones animadas aclaratorias
- Demostraciones y ejemplificación
- Videos

#### Actividades extra áulicas

Complementariamente a la actividad áulica los estudiantes realizarán actividades en ámbitos extra áulicos planificadas y supervisadas por el profesor. El propósito de estas actividades es promover oportunidades de aprendizaje en contextos reales o simulados, en modalidad presencial o virtual, donde los estudiantes puedan ampliar/ profundizar los conocimientos, habilidades y actitudes desarrollados en cada asignatura. Estas propuestas extienden el aula tradicional a ámbitos más atractivos y motivadores para el alumno.

Las actividades extra áulicas asumirán diferentes denominaciones y modalidades conforme la propuesta didáctica de cada asignatura. Serán propuestas y acordadas con los estudiantes al inicio de la cursada en cada comisión.

#### **15. Trabajos prácticos:**

a. Elaboración de una carpeta de ejercicios prácticos que será entregada al docente. La carpeta de trabajos prácticos consta de ejercicios, estos están agrupados de acuerdo con las unidades de la asignatura. Dentro de cada unidad el conjunto de ejercicios presenta problemas cuya resolución implica la aplicación de técnicas de programación y permiten abordar el problema tratado desde diferentes ópticas y con niveles graduales de dificultad. El docente podrá tomar algunos ejercicios emblemáticos del conjunto y desarrollarlos en el aula conjuntamente con los alumnos a modo de ejemplo. El docente sugerirá la resolución de cada conjunto de ejercicios en la medida que culmine con el desarrollo de la unidad correspondiente a la asignatura. La forma de entrega es impresa y en soporte magnético. La misma es de producción individual y se entregará en la

fecha indicada por el docente dentro del plazo en que se desarrolla el cuatrimestre.

b. Trabajo de indagación. Opcional. Los trabajos indagación estarán referidos a temas de la asignatura. Se realizarán en forma grupal y el número de sus integrantes será de cuatro a seis alumnos. La entrega de los mismos será fijada oportunamente por el docente dentro del plazo en que se desarrolla el cuatrimestre.

## **16. Bibliografía:**

### **Obligatoria:**

#### Teoría:

Deitel, Harvey M.; Deitel, Paul J.; Vidal Romero Elizondo, Alfonso (Traductor); y otros. Cómo programar en C#. 6a.ed.-- México, DF : Pearson.2016

Mayo, Joseph. C# al descubierto.-- Madrid: Pearson Educación, c2002. xxxi, 749 páginas

Brassard, G.; Bratley, P.. Fundamentos de algoritmia.-- Madrid: Prentice Hall, 1998. xiii, 579 páginas

### **Ampliatoria:**

Ampliatoria: “Programación Avanzada con Microsoft Visual Basic .NET”. Francesco Balena. Mc Graw Hill. 2003. España. Capítulo 25.

Ceballos Sierra, Francisco Javier. Microsoft C# lenguaje y aplicaciones. 2a. ed.-- México, DF: Alfaomega, c2008. xviii, 499 páginas, Wille, Christoph. C#.-- Madrid: Pearson Educación, c2001. xv, 208 páginas

Booch, Grady; Cardacci, Dario G. Orientación a objetos: Teoría y práctica.--Buenos Aires: Pearson Educación, 2104. 338 páginas.

Rosa Guerequeta, Antonio Vallecillo. (1998). Técnicas de Diseño de Algoritmos. Universidad de malaga. (<http://www.lcc.uma.es/~av/Libro/>)

Maria, Gopmez fuetes. Jorge, Cervantes. (2014). Introducción al análisis y diseño de algoritmos. Universidad autónoma metropolitana.

([http://www.cua.uam.mx/pdfs/conoce/libroselec/Notas\\_Analisis\\_AlgoritmosVF.pdf](http://www.cua.uam.mx/pdfs/conoce/libroselec/Notas_Analisis_AlgoritmosVF.pdf))

## **17. Procedimiento de evaluación y criterio de promoción:**

### **Parciales.**

Los exámenes parciales serán como mínimo dos, cada uno con teoría y práctica. Ambos son obligatorios, individuales y escritos. El primero de ellos se efectuará entre la quinta y séptima semana del cuatrimestre y el segundo entre la semana doce y catorce. Los parciales deberán tener un 60% de contenido correcto para estar aprobados. En este último caso la nota deberá quedar establecida entre 4 (cuatro) y 10 (diez) puntos.

### **Recuperatorios.**



Los exámenes recuperatorios se desarrollarán a razón de uno por cada parcial con teoría y práctica. Se podrá optar por tomar un recuperatorio integral en aquellos casos que se justifique y el mismo será equivalente al alcance de los recuperatorios individuales. Recuperarán los parciales que correspondan, aquellos alumnos que obtuvieron una calificación inferior a 4 (cuatro) puntos en alguno de sus exámenes parciales. Los recuperatorios son individuales y por escrito. El docente dispondrá la fecha para realizar esta actividad que podrá ser con posterioridad a cada uno de los parciales o en una fecha común para todos. Las notas de los exámenes recuperatorios no reemplazan las calificaciones insuficientes obtenidas en los exámenes regulares que se están recuperando, las mismas serán tomadas como parte de las calificaciones que intervienen en el promedio.

#### **Aspectos comunes a parciales y recuperatorios.**

La confección de los exámenes parciales y recuperatorios contemplarán diversas técnicas para poder observar desde distintos ángulos el objeto de evaluación. Se intercalarán preguntas a desarrollar, del tipo múltiples casos, guiadas y lógicas. Se evaluará la capacidad de razonamiento del alumno y el marco teórico que lo sustenta.

#### **Trabajos Prácticos.**

Se desarrollarán los previstos en el punto 15.

#### **Carpeta de ejercicios.**

La carpeta de ejercicios llevará una nota de calificación entre 1 (uno) y 10 (diez) puntos la cual se puede promediar con la nota de concepto.

#### **Trabajo de Indagación.**

Los trabajos deberán ser expuestos por los alumnos. Los trabajos y la exposición serán grupales y la evaluación del mismo estará conformada por un promedio de notas que involucren la producción grupal y el desempeño individual. Los aspectos a tener en cuenta en dicha evaluación son: calidad de la indagación, preparación y clasificación del material, producción del material, calidad de la exposición oral, aprovechamiento de los medios y administración del tiempo.

#### **Nota conceptual.**

Una nota conceptual basada en la participación en clase del alumno, el interés demostrado en la asignatura y el empeño puesto de manifiesto en las actividades de estudio.

#### **Criterios para la promoción.**

Para aprobar la asignatura y acceder al examen final o coloquial el alumno deberá tener sus dos exámenes parciales y el trabajo práctico aprobado, más la entrega de la carpeta con la resolución de los ejercicios de programación.

El alumno deberá tener como mínimo cuatro notas para poder promediar sin contar los recuperatorios de parciales.

Para aprobar la asignatura se debe tener un promedio de 4 o más puntos. Esto habilita al alumno a rendir un examen Final. En caso que el promedio sea de 7 o más puntos el alumno accede a dar un examen coloquial.