

Programa de Asignatura

00 Código:	F	C	Asign
	T	417	09

01. Facultad: Tecnología Informática / Carrera: Analista Programador

02. Asignatura: T17 – 09 - PROGRAMACIÓN ORIENTADA A OBJETOS.

03. Año lectivo: 2022

04. Año de Cursada: 2°

05. Cuatrimestre: 1°

06. Carga horaria semanal: 6

07. Créditos:

08. Equipo Docente

Profesor Titular: Ing. Cardacci Darío Guillermo
Adjuntos:
Auxiliar:
Tapia Gastón

09. Asignaturas correlativas previas y posteriores:

Correlativas previas: Programación I

Correlativas posteriores: Lenguajes de Última Generación
Programación II

10. Fundamentación:

a- Aporte específico de la asignatura a la formación académico profesional

Los sistemas utilizados en la actualidad demandan para su desarrollo la utilización de paradigmas que se adapten a una realidad cambiante, así como acceder a grandes volúmenes de datos. Para ello, los paradigmas utilizados en el desarrollo de los nuevos sistemas de información, deben poseer las características necesarias que permitan representar virtualmente en un ámbito computacional los problemas reales que requieren una solución.

En particular el modelo orientado a objetos y la programación orientada a objetos, permiten abstraer escenarios reales que deben representarse en entornos computacionales. Esto permite formular software de calidad con las características que requieren los desarrollos orientados a objetos.

La adquisición de conocimientos referidos a la arquitectura de servicios que ofrecen los entornos de desarrollo permite que los programadores reutilicen la experiencia y los conocimientos previamente adquiridos y ofrecidos a través de los componentes que lo conforman. La reutilización permite que el software desarrollado se más eficiente.

Finalmente cabe mencionar que comprender cómo funcionan los puertos de comunicaciones extiende las posibilidades del software a que pueda controlar y accionar sobre dispositivos de hardware conectados a la computadora por cables o de forma inalámbrica, así como intercambiar datos entre distintas aplicaciones.

Esta asignatura le aporta al alumno la posibilidad de incorporar conocimientos sobre las técnicas más avanzadas de programación.

b- El /los marco/s conceptual/es que sustenta/n el enfoque de la asignatura

La asignatura aborda el modelo orientado a objetos como metodología que optimiza la forma de producir software. También explora las tecnologías asociadas que permiten lograr software de alta calidad

c- La articulación de la asignatura con asignaturas previas y posteriores

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.

Le provee a Metodología de Desarrollo de Sistemas II una visión práctica y teórica sobre orientación a objetos lo cual facilita la tarea de aprendizaje de las formas de modelado estándar desarrolladas en la asignatura con la que articula.

A Lenguajes de Última Generación le provee las herramientas de programación para encarar los aspectos relacionados con el acceso a datos desde un programa informático y la aplicación de técnicas de programación refinadas.

Articulación Horizontal:

La articulación con la asignatura Metodología de Desarrollo de Sistemas I permite que se puedan trabajar los aspectos referidos a la documentación y los métodos para desarrollar software.

11. Competencias / sub-competencias y resultados de aprendizaje a las que tributa la asignatura:

Competencias del perfil	Subcompetencia	Nivel de dominio de la subcompetencia	Resultados de aprendizaje
<i>2- Colaborar en la análisis, planificación, programación, implementación y mantenimiento de desarrollo de software de forma efectiva.</i>	2.1 - Capacidad para seleccionar el paradigma de programación más adecuado para enfocar los procesos de construcción de un software.	2	T4-17-09-2-1-2-RA1: [Comprende] + [las características de los programas orientados a objetos] + [para formular software de calidad] + [utilizando métodos y estrategias estandarizadas]
<i>2 - Colaborar en la análisis, planificación, programación, implementación y mantenimiento de desarrollo de software de forma efectiva.</i>	2.3 - Capacidad para reutilizar y adaptar rutinas y programas existentes, integrándolas con código propio.	2	T4-17-09-2-3-2-RA2: [Comprende] + [las tecnologías de optimización e intercambio de datos entre aplicaciones]+ [para mejorar los desarrollos de software] + [aplicando el paradigma OO]
<i>5 - Integrar tareas técnicas de servicios informáticos en instituciones públicas, en empresas de producción y servicios públicos y privados, para realizar aportes que se incorporen de manera sustantiva desde la programación.</i>	5.1 - Capacidad para actuar técnicamente con eficiencia en diferentes tipos de organizaciones.	1	T4-17-09-5-1-1-RA3: [Implementa]+ [interfaces y delegados]+ [para obtener eficientemente software] + [considerando las posibilidades de los lenguajes de programación OO]

12. Unidades de desarrollo de los contenidos:

UNIDAD 1: OBJETOS Y CLASES.

El modelo orientado a objetos.

Jerarquías “Es - Un” y “Todo - Parte”.

Concepto de Objeto. Características básicas de un objeto: estado, comportamiento e identidad.

Ciclo de vida de un objeto.

Modelos. Modelo estático. Modelo dinámico. Modelo lógico. Modelo físico.

Concepto de análisis diseño y programación orientada a objetos.

Conceptos de encapsulado, abstracción, modularidad y jerarquía. Concurrencia y persistencia.

Concepto de clase. Definición e implementación de una clase. Campos y Constantes

Propiedades. Concepto de Getter() y Setter(). Propiedades de solo lectura. Propiedades de solo escritura. Propiedades de lectura-escritura. Propiedades con indizadores. Propiedades autoimplementadas. Propiedades de acceso diferenciado.

Métodos. Métodos sin parámetros. Métodos con parámetros por valor. Métodos con parámetros por referencia. Valores de retorno de referencia. Sobrecarga de métodos. Constructores. Constructores predeterminados. Constructores con argumentos. Finalizadores. Clases anidadas.

Tiempo: 24 Hs.

UNIDAD 2: OBJETOS Y CLASES - RELACIONES

Eventos. Suscripción a eventos. Suscripción a eventos utilizando el IDE. Suscripción a eventos mediante programación. Suscripción a eventos mediante métodos anónimos. Publicación de eventos. Desencadenar eventos. Modificadores de acceso. Clases abstractas, selladas y estáticas. Miembros estáticos en clases estáticas. Sobreescritura de métodos. Métodos virtuales. Relaciones básicas entre clases. “Generalización-Especialización”, “Parte de”. Relaciones derivadas entre clases. Herencia. Herencia simple. Herencia múltiple. Polimorfismo. Agregación. Simple y con contención física. Asociación y relación de Uso. Teoría de Tipos. Tipos anónimos. Elementos que determinan la calidad de una clase: acoplamiento, cohesión, suficiencia, compleción y primitivas. Relaciones entre objetos: enlace y agregación. Acceso a la clase base desde la clase derivada. Acceso a la instancia actual de la clase.

Tiempo: 24 Hs.

UNIDAD 3: FRAMEWORKS Y MANEJO DE EXCEPCIONES.

Concepto de frameworks. Elementos de un framework. Tipos de frameworks. Arquitectura de .NET. Interoperatividad entre .NET y COM. Código administrado y no administrado. Common Language Runtime CLR. Lenguaje intermedio IL. El compilador Just-in-Time (JIT). Concepto de assembly. Administración de la memoria en .NET. El Garbage Collector. Manejo de excepciones. Control de excepciones. El objeto Exception. La instrucción Try – Catch – Finally. La instrucción Throw. Depuración de aplicaciones. Herramientas de depuración. Análisis del comportamiento de las aplicaciones.

Tiempo: 12 horas

UNIDAD 4: INTERFACES Y DELEGADOS

Interfaces. Desarrollo e Implementación de una interfaz. La interfaz IComparable. La interfaz IComparer. La interfaz ICloneable. Las interfaces IEnumerable e IEnumerator.

Delegados. Delegados con métodos con nombre. Delegados con métodos anónimos.

Tiempo: 12 horas

UNIDAD 5: GENÉRICOS, LINQ Y EXPRESIONES LAMBDA

Introducción a los genéricos. Ventaja de usar genéricos. Clases genéricas. Interfaces y métodos genéricos.

LINQ to Object. Introducción a las consultas con LINQ. Escritura de consultas con LINQ. Retorno y almacenamiento de consultas LINQ. Grupos anidados y subconsultas con LINQ.

Introducción a las expresiones lambda. Funciones anónimas. Uso de expresiones lambda en consultas.

Tiempo: 12 horas

UNIDAD 6: COMUNICACIÓN ENTRE APLICACIONES Y MANEJO DE DISPOSITIVOS

Esquema cliente - servidor. Configuración de aplicaciones remotas en una red. Formas de compartir información entre aplicaciones. Pasaje de información batch vs. On line.

Conceptos básicos de protocolos. TCP. UDP. Concepto sobre IP, TCP/IP. Concepto de software cliente – servidor. Ventajas y desventajas. Distribución de procesos y Almacenamientos. Sistema de mensajería de un sistema cliente – servidor.

Concepto de Subproceso. Utilización de Sockets. Sockets de clientes sincrónicos y asincrónicos. Socket de servidores sincrónicos y asincrónicos.

Tiempo: 12 hs.

13. Metodología:

a. Encuadre metodológico de la asignatura

La asignatura plantea un abordaje metodológico sustentado en diferentes estrategias. Temporalmente las actividades se encuentran distribuidas en 16 semanas. En el primer encuentro sincrónico se presenta la asignatura, los resultados de aprendizaje esperados y la distribución de las distintas actividades que se desarrollarán. Para los siguientes encuentros sincrónicos se le solicita al alumno que previamente aborde el material que será tratado desde las guías de orientación desarrolladas para la asignatura.

En los encuentros se orienta al alumno en los aspectos conceptuales que ha abordado. Se propone que los alumnos trabajen grupalmente en la resolución de problemas que deban aplicar los temas desarrollados en los orientadores y los encuentros. Deberán presentar la solución a través de los foros y actividades planteadas en la plataforma on line. En los

encuentros, como actividad de cierre se reflexiona grupalmente sobre las conclusiones a las que se ha arribado y como esto aporta a los resultados de aprendizaje finales que se espera lograr. Finalmente se plantea la actividad a desarrollar de manera asincrónica y los objetivos a alcanzar para el siguiente encuentro sincrónico. Se sugiere realizar:

- La lectura del orientador.
- La resolución de lo correspondiente a la guía de revisión conceptual y la guía de trabajo práctico.
- La participación en el foro correspondiente a la unidad de la asignatura que se está abordando.
- Cuando la situación lo amerite la resolución de un desafío, una indagación bibliográfica o la resolución de un trabajo integrador que se desarrollará en algunas oportunidades de manera grupal y en otras de forma individual.

b. Contextos de desarrollo de las actividades formativas

Las actividades formativas se llevarán a cabo en dos ámbitos diferente.

Los ámbitos utilizados serán: El aula virtual en el LMS institucional y los encuentros sincrónicos.

De manera complementaria y de forma asincrónica el alumno complementará el abordaje de las guías de orientación, la guía de revisión conceptual y la guía de trabajos prácticos.

Se propone como complemento a las estrategias de aprendizaje, el empleo de tecnologías digitales que favorezcan el desarrollo de actividades (individuales o grupales) participativas y colaborativas que promuevan la autonomía del estudiante en el proceso de enseñanza sobre la base de siguientes actividades:

- Profundización de contenidos: Tutoriales, videos, lecturas complementarias.
- Demostraciones: Resultado de procesos deductivos
- Resolución de problemas: Indagación de respuestas sobre problemas simulados a fin de crear una solución.
- Foros: Discusión colaborativa sobre temas de interés de la asignatura.

Se promoverá la utilización del aula invertida con el objetivo de potenciar los trabajos de integración que permitan desarrollar los resultados de aprendizaje establecidos en el programa de la asignatura para ir acercándolos paulatinamente a las problemáticas profesionales características de la carrera.

Actividad formativa	Ámbito	Resultado de aprendizaje	Contenidos comprometidos	Producción de los estudiantes	Observaciones
Profundización de contenidos	LMS	Comprende las características de los programas orientados a objetos para formular software de calidad utilizando métodos y	Unidad 1 Unidad 2	Solución de la Guía de Revisión conceptual	

		estrategias estandarizadas			
Demostraciones	LMS	Comprende las tecnologías de optimización e intercambio de datos entre aplicaciones para mejorar los desarrollos de software aplicando el paradigma OO	Unidad 3 Unidad 6	Solución de la Guía de Trabajos Prácticos	
Resolución de problemas	LMS	Implementa interfaces y delegados para obtener eficientemente software considerando las posibilidades de los lenguajes de programación OO	Unidad 4 Unidad 5	Solución de la Guía de Trabajos Prácticos	
Foros	LMS	Comprende las características de los programas orientados a objetos para formular software de calidad utilizando métodos y estrategias estandarizadas	Unidad 1 Unidad 2 Unidad 3 Unidad 4 Unidad 5 Unidad 6	Elaboración de contenidos asociados a la propuesta del foro correspondiente a cada unidad	

c. Recursos didácticos

Se utilizarán los siguientes recursos didácticos:

- Guía de orientación por unidad
- Guía de revisión conceptual
- Guía de trabajos prácticos
- Ejercicios Integradores
- Material multimedial
- Código de programación
- Foros
- Docs de Microsoft
- Learn de Microsoft

14. Procedimiento de evaluación y criterios de promoción:

a. Modalidad y Criterios de Evaluación

La evaluación se sustenta en la participación de los encuentros, la producción domiciliaria (TP's y trabajos de indagación) y los parciales. Para lograr una evaluación continua se trabajará con rúbricas que permitirán observar a lo largo del cuatrimestre el grado de apropiación de los temas tratados en la asignatura.

A continuación, se detallan los criterios aplicados para cada instrumento:

Parciales.

Los exámenes parciales serán como mínimo dos, cada uno con articulación teórica 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. Si el alumno no se presenta a un examen parcial se calificará con una nota insuficiente.

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- el cual 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.

a. Elaboración de una carpeta de ejercicios prácticos. Se desarrollará de forma individual y el docente podrá seleccionar ejercicios para que se realicen en forma grupal. La carpeta de trabajos prácticos a resolver es la **“Guía de Trabajos Prácticos de la Asignatura”** y la **“Guía de Revisión Conceptual”**. La primera 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 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 segunda guía son preguntas de autoevaluación sobre los temas abordados en cada unidad. El docente indicará la fecha y forma de entrega dentro del cuatrimestre.

b. Trabajo de indagación. Los trabajos de 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 estos será fijada por el docente dentro del plazo en que se desarrolla el cuatrimestre.

Ejercicios integradores.

Los ejercicios integradores 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.

Se desarrollará un trabajo práctico de indagación bibliográfica obligatorio referido a temas de la asignatura y deberá ser expuesto por los alumnos. El trabajo y la exposición será grupal y la evaluación de este 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 investigación, preparación y clasificación del material, producción del material, calidad de la exposición oral, dominio del lenguaje específico, 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.

b. Evidencias

Las evidencias que darán cuenta del logro de los objetivos planteados en el curso son:

- a. Resolución de la guía de trabajos prácticos.
- b. Resolución de la guía de abordaje conceptual
- c. Trabajo de indagación.
- d. Rúbricas.
- e. Foros
- f. Resolución de los ejercicios integradores.
- g. Parciales

c. Requisitos de aprobación de la asignatura

Para aprobar la asignatura y acceder al examen final o coloquio 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 los estudiantes deberán cumplir los siguientes requerimientos:

1. Asistencia: Los estudiantes deben contar con una asistencia mínima del 70%.
2. Aprobar mediante alguna de estas modalidades:
 - a. Evaluación final “integradora coloquial”: accederán a este régimen de evaluación aquellos alumnos cuyo promedio de cursada se encuentre comprendido entre 6 y 10 puntos. Los alumnos podrán presentarse a la mesa examinadora en grupos de no más de tres personas. La evaluación se realizará en forma individual, debiendo demostrar el dominio de la asignatura como unidad y la capacidad de asociarla con otras materias del plan de estudio ya cursadas.

- b. Examen final: acceden a este régimen de evaluación aquellos alumnos que han obtenido durante su cursada un promedio comprendido entre 4 y 5,99 puntos. El alumno se presentará en forma individual ante un tribunal examinador, el que interrogará sobre el programa de la asignatura, debiendo demostrar su capacidad de integrarla y relacionarla con otras asignaturas del plan de estudio.

Examen recuperatorio: En las carreras de grado, aquellos alumnos cuyo promedio sea inferior o igual a 3,99 puntos (obtenidos en las dos o tres evaluaciones parciales) y tengan una asistencia igual o mayor al 50%, deberán rendir un Examen Recuperatorio de Materia en la misma fecha que se indica para el Examen Final.

Habiendo aprobado el Examen Recuperatorio de la Materia, con nota mínima de 4 (cuatro) puntos, estarán en condiciones de acceder al Examen Final en el siguiente llamado, con la previa inscripción realizada 48 horas hábiles antes del mismo.

Para rendir Examen Recuperatorio de la Materia no es necesario presentar Permiso de Examen, pero deberá registrarse con no menos de 48 horas hábiles de anticipación. Los alumnos de la modalidad a distancia deberán registrarse con no menos 72 horas hábiles de anticipación.

El derecho a rendir Examen Recuperatorio de la Materia tendrá validez por dos años o dos presentaciones a Exámenes Recuperatorios de la Materia, lo que se produzca primero.

Objetivos de la evaluación.

Validar que el alumno alcanzó los resultados de aprendizaje con el nivel de dominio planteado para la asignatura y el grado de internalización de los contenidos.

Verificar los errores y cambios conceptuales en las estructuras cognitivas de los alumnos.

Garantizar la transferencia del conocimiento.

Reconocer la integración de aprendizajes significativos.

Contenidos de la evaluación.

Se tendrán en cuenta los contenidos conceptuales, procedimentales y actitudinales que denoten la apropiación de los resultados de aprendizaje esperados.

15. Bibliografía

Obligatoria:

Deitel Harvey M. Y Paul J. Deitel. 2007. **Cómo programar en C#**. Segunda edición. Pearson. Prentice Hall. México.

Cardacci Darío y Booch, Grady. 2013. **Orientación a Objetos. Teoría y Práctica**. Pearson Argentina. Buenos Aires, Argentina.

Lenguaje C#. Docs de Microsoft. Tutoriales. <https://docs.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/tutorials/>.

Lenguaje C#. Docs de Microsoft. Documentos C#. <https://docs.microsoft.com/es-es/dotnet/csharp/>

Ampliatoria:

-
- Zacker, Craig Rourke, John. 2001. **PC Hardware: manual de referencia**. McGraw-Hill Interamericana. Madrid.
- Martin, James; Odell, James J. 1997. **Métodos orientados a objetos: conceptos fundamentales**. Prentice Hall Hispanoamericana. México, DF.
- Rumbaugh, James; Blaha, Michael; Premerlani, William y otros. 1991. **Modelado y diseño orientado a objetos**. Prentice Hall Regent. Madrid.