



## PROGRAMA ANALITICO Y DE EXAMEN

### 1. IDENTIFICACION

- 1.1. FACULTAD DE CIENCIAS EXACTAS Y NATURALES Y AGRIMENSURA
- 1.2. DEPARTAMENTO: **Informática**
- 1.3. AREA: **Programación**
- 1.4. ASIGNATURA: **Programación Orientada a Objetos**
- 1.5. CARRERAS: **Licenciatura en Sistemas de Información**  
Año en que se dicta: **Segundo** - Cuatrimestre: **Segundo**
- 1.6. PROFESOR RESPONSABLE:  
Apellido y Nombres: **Greiner, Cristina Liliam**  
Máximo Título Alcanzado: **Magister en Informática y Computación**
- 1.7. MODALIDAD: **Cuatrimestral**
- 1.8. CARGA HORARIA TOTAL: **128 hs**
- 1.9. CARGA HORARIA SEMANAL TEORICA/PRACTICA: **8 hs.**

### 2. DESCRIPCION

#### Fundamentación

El software se ha convertido en un elemento crucial en la economía mundial. En consecuencia, se realizan grandes esfuerzos por lograr procesos de desarrollo de software eficientes, y que permitan obtener productos de calidad. La calidad en los productos software es una exigencia creciente, dado que cada vez es más amplio su uso en procesos que son críticos para las organizaciones. Ante estas exigencias, surge la ingeniería del software, que según Ivar Jacobson, es "...un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad" y "... estudia la aplicación de técnicas formales de desarrollo a la construcción de software, buscando obtener productos de calidad, con la mejor relación costo/eficiencia".

En este sentido, la Programación Orientada a Objetos (POO) es una metodología de desarrollo de software que permite crear unidades funcionales extensibles, de forma que el desarrollador las pueda reutilizar. Es un paradigma que se fundamenta en concebir a un sistema como un conjunto de entidades que representan al mundo real (los objetos), que tienen distribuida y encapsulada la información y funcionalidad necesaria, y que cooperan entre sí para el logro de un objetivo común. Esta metodología de desarrollo se enfoca principalmente a la reutilización del código y a facilitar su mantenimiento. Esto se consigue mediante mecanismos de abstracción y otros principios, como jerarquía, encapsulación y polimorfismo, características propias de la POO, y que redundan en un menor tiempo en el proceso, tanto de desarrollo como del posterior mantenimiento y extensibilidad.

La cátedra tiene como propósito lograr en el alumno un aprendizaje significativo, entendiéndose por tal "...aquel que el alumno incorpora a su estructura de conocimiento de modo significativo, es decir, que las nuevas adquisiciones se relacionan con lo que él ya sabe, siguiendo una lógica, con sentido, y no arbitrariamente"<sup>1</sup>.

Para este fin, los contenidos están estructurados de modo de incorporar progresivamente los conceptos, facilitando su comprensión mediante la implementación práctica de los mismos.

---

1 Ausubel-Novak-Hanesian (1983) *Psicología Educativa: Un punto de vista cognoscitivo* .2º Ed. TRILLAS México.



El curso comienza con un análisis de las características que hacen a la calidad del software. Este análisis sirve para fundamentar el surgimiento de la POO, como una solución a las carencias propias de la programación imperativa y de otras técnicas de diseño más clásicas.

El núcleo del curso se ocupa del estudio de los mecanismos que definen el paradigma de la POO, como son el desarrollo de aplicaciones en términos de clases de objetos, la organización de las clases en jerarquías de herencia, el polimorfismo y la vinculación dinámica. Tomando Java como lenguaje de referencia, para cada mecanismo concreto se exploran distintas alternativas de ejemplos que propician su comprensión.

Adicionalmente, se integran aspectos distintivos de los lenguajes de POO, tal como excepciones y concurrencia, dada la importancia que estas tecnologías tienen en la actualidad. Así también, se da relevancia al rol que tiene la documentación en todo proceso de desarrollo de software.

## Metodología

Para el logro de los objetivos de este curso, se impartirán los contenidos detallados en el programa analítico mediante el dictado de clases teóricas, prácticas, y la práctica experimental intensiva en el Laboratorio de Informática. Las primeras brindarán los conceptos necesarios que serán integrados con las actividades prácticas. Éstas tienen como propósito fundamental que los alumnos comprendan cómo se implementan en forma práctica los principios que sustentan este paradigma. Las clases de laboratorio favorecen este proceso, ya que permiten observar los resultados en forma directa.

El objetivo de esta asignatura es que los alumnos comprendan los conceptos fundamentales del paradigma de POO, percibiendo las ventajas que proporciona en la resolución de problemas e internalizándolas, dado que introduce técnicas que facilitan el mantenimiento y la reutilización.

Los criterios de evaluación son los siguientes: Capacidad para resolver los problemas planteados utilizando las técnicas de la programación orientada a objetos, correcta interpretación del diagrama de clases en UML, utilización de los términos técnicos específicos, capacidad de síntesis y de relación de los distintos conceptos.

Como complemento a las actividades presenciales, la asignatura dispone de un aula virtual en la plataforma de UNNE, cuyo propósito es servir de apoyo a las actividades de la asignatura, concentrando allí toda la información necesaria para alumnos y docentes: apuntes teóricos, guías de trabajos prácticos y de laboratorio, planificación de las actividades, la composición de los grupos de prácticos y docentes responsables y la información administrativa de los alumnos, notas obtenidas en cada examen parcial y situación académica final. Además cuenta con una sección de sitios sugeridos y la bibliografía recomendada. A través de la misma se realiza la atención de consultas, y el diseño del aula también contempla la inclusión de foros para el debate de temas puntuales, de los cuales interesa conocer la interpretación de los alumnos.

Por otra parte, se propicia la realización de actividades para estimular el aprendizaje independiente.

Al inicio del cursado se pondrá a disposición de los alumnos la planificación completa de todas las actividades previstas en la asignatura, clases teóricas y prácticas, fechas de exámenes parciales y finales, y fechas de evaluación de los trabajos de laboratorio, respetando el calendario académico establecido. Esta información les permitirá organizar su tiempo de estudio y realizar su propia planificación, en función de las otras obligaciones académicas.

Como parte de las estrategias de enseñanza y aprendizaje, al finalizar el cuatrimestre se efectuará una encuesta a los alumnos, instrumento de recolección de información que permite obtener datos sobre el dictado, de manera tal que los resultados permitan proponer mecanismos o estrategias superadoras.

## Articulación con otras asignaturas

Este espacio curricular se asienta en los conceptos iniciados en las asignaturas **Algoritmos y Estructuras de Datos I y II**, precedentes en el área Programación, en la que se imparten las estructuras de datos simples y compuestos, y los algoritmos que operan sobre ellos, sentando las



bases del paradigma de programación procedural y afianzando el pensamiento lógico para la resolución de problemas.

En la asignatura **Paradigmas y Lenguajes** se realiza una introducción a los distintos paradigmas de programación, incluido el paradigma de objetos, objeto de estudio en el presente espacio curricular. Los lenguajes estudiados en las asignaturas del área, precedentes a ésta, así como los lenguajes de programación utilizados como herramienta, abren el camino para el aprendizaje del lenguaje Java, herramienta utilizada para la implementación de los conceptos del paradigma POO.

En el **Taller de Programación II**, se partirá de la aplicación desarrollada como ejercicio integrador al final del dictado de esta asignatura, con el propósito de adicionar características avanzadas, tanto del lenguaje como de las herramientas utilizadas en el desarrollo de aplicaciones OO.

Las asignaturas que junto a e ésta tratan temas relativos al desarrollo de software orientado a objetos (**Ingeniería del Software I y II, Bases de Datos I**), comparten enunciados de ejercicios, con el propósito de dar continuidad a los conceptos aprendidos por los alumnos.

## **2.1. OBJETIVO(S):**

### **2.1.1 Objetivos Generales**

El objetivo general de la enseñanza de la programación es capacitar a los alumnos para construir metódicamente programas legibles, bien documentados, correctos, eficientes y fáciles de mantener y reutilizar. Dentro de este marco, la POO introduce una serie de técnicas y mecanismos que favorecen fundamentalmente los dos últimos criterios de calidad: la facilidad para mantener y modificar los programas, y la posibilidad de desarrollar programas o componentes fácilmente reutilizables.

### **2.1.2 Objetivos Particulares**

En este contexto, se pretende lograr que el alumno:

- a) Comprenda los conceptos básicos de la POO, abarcando las características fundamentales de esta metodología: definición de objeto, clase, herencia, polimorfismo, relaciones y envío de mensajes, mediante la resolución de problemas diseñados a tal efecto
- b) Valore e internalice las ventajas que proporciona el paradigma POO en la resolución de problemas, mediante la aplicación de técnicas que facilitan el mantenimiento y la reutilización de objetos (encapsulamiento, herencia, polimorfismo, etc)
- c) Sea capaz de resolver problemas planteados en cualquier área del conocimiento a través del empleo sistemático y disciplinado de métodos, modelos y herramientas
- d) Implemente los conceptos de POO impartidos utilizando uno de los lenguajes más representativos de este paradigma, y de mayor difusión, por su uso masivo en Internet, como es el lenguaje JAVA
- e) Interprete las especificaciones de diseño de la solución de un problema planteado, representado en el lenguaje estándar de modelado de objetos UML (Lenguaje de Modelado Unificado)
- f) Conozca y aplique técnicas de documentación, advirtiendo la importancia de la misma en la obtención de software de calidad
- g) Maneje con precisión el lenguaje técnico asociado al paradigma de objetos.

Además de los conceptos propios del paradigma POO, se propone estimular la adquisición de competencias generales y específicas en el proceso de formación profesional (contenidos procedimentales y actitudinales), tales como:

- a) Capacidad de resolución de problemas, reforzando las habilidades adquiridas en asignaturas previas y propiciando las "buenas prácticas" valoradas en la programación.



- b) Disposición al aprendizaje activo y autónomo.
- c) Aptitud para el uso eficiente de la documentación propia de las herramientas de desarrollo y los estándares publicados por organismos especializados.
- d) Habilidad para la interpretación de documentación escrita en inglés.
- e) Predisposición al trabajo en equipo mediante el cumplimiento de los roles específicos que le competen a los programadores en un proyecto de desarrollo.
- f) Desempeño responsable a través de la exigencia del cumplimiento en tiempo y forma de los trabajos solicitados.
- g) Acatamiento de protocolos definidos para la elaboración y entrega en tiempo y forma de los trabajos, fomentando el trabajo profesional (formas estándares de identificar los elementos, de agrupar la información, etc.). Esta actividad introduce buenas prácticas sobre el intercambio de información, sujeto a reglas y protocolos.
- h) Actitud ética de respeto y valoración del trabajo ajeno.

## 2.2. TIPO/S DE ACTIVIDAD/ES: *(marque con una cruz)*

Clases:

- Teóricas: **X**
- Prácticas: **X**
- De Laboratorio: **X**
- Trabajo de Campo
- Investigación

**2.2.1. Técnicas o Estrategias didácticas:** *(Exposición del docente/del alumno, trabajos grupales, estudio independiente, resolución de situaciones problemáticas, resolución de ejercicios de aplicación, presentación de monografías, de informes, etc., presentación y desarrollo de un proyecto, etc.).*

Para el cumplimiento de los objetivos del curso se prevé el dictado de clases teóricas y prácticas, y la práctica experimental intensiva en el Laboratorio de Informática.

Un papel importante en la estrategia docente cumple también el apoyo complementario del Aula Virtual.

A continuación se detalla cada una de estas actividades, así también como el conjunto de estrategias didácticas que complementan esta propuesta pedagógica:

### a) Clases Teóricas

Las clases teóricas tienen como objetivo primordial la presentación de los conceptos, su análisis y la descripción de sus principales aplicaciones.

Cada clase presenta la siguiente estructura:

- La exposición se inicia con una introducción donde se plantean los objetivos del tema a desarrollar, así también como su ubicación en el contexto de la asignatura y relación con temas expuestos previamente, o a ser desarrollados con posterioridad.
- Se intenta captar la atención de los alumnos, realizando preguntas sobre temas afines, tratando de que cobren significado los nuevos conceptos a introducir, de modo que el alumno pueda desarrollar las relaciones necesarias.
- Luego se procede al desarrollo propiamente dicho, donde se exponen los contenidos en forma organizada y precisa, propiciando su comprensión mediante el empleo de ejemplos, relacionados con la actividad profesional, que faciliten el entendimiento de los conceptos.



Esta etapa se acompaña de ejemplos de código fuente y/o aplicaciones prácticas, de modo de favorecer el proceso de aprendizaje.

- La exposición concluye con una revisión de lo expuesto, resaltando los contenidos centrales.
- Finalmente, a través de la interacción con el alumno, se verifica el grado de comprensión de los conceptos vertidos y se busca despertar su interés por los temas tratados. Para cumplir con estos propósitos se fomenta la discusión y la participación del alumno a través de la interrogación y el análisis de ejemplos de programación característicos.

Para el dictado de las clases se empleará un proyector multimedia, recurso que permite la presentación de programas y algoritmos, gráficos, animaciones, y otros elementos multimediales, que contribuyen significativamente en el aprendizaje.

#### **b) Clases Prácticas**

El desarrollo de la clase práctica es de fundamental importancia para la apropiación y el afianzamiento de los conceptos, dado que permite al alumno poner en práctica los conocimientos adquiridos para alcanzar la solución a los problemas planteados.

Para el dictado de estas clases, el total de alumnos es dividido en dos grupos, cada uno a cargo de un JTP, que cuenta con el apoyo de un auxiliar.

Durante la clase los alumnos deben implementar la solución a problemas presentados en guías de trabajos prácticos, empleando las técnicas de la POO, con la herramienta indicada (lenguaje Java).

En la implementación de estos ejemplos se insistirá en la puesta en práctica de estructuras de datos y procedimientos adecuados al problema que se resuelve.

Cabe señalar que como un minúsculo aporte al **objetivo de formación del alumno en el idioma inglés**, cada serie de trabajos prácticos incluye un enunciado en dicho idioma. Los alumnos son informados del propósito de dicha práctica, aprovechando la oportunidad para resaltar la importancia del dominio del mencionado idioma.

Cada clase se desarrolla de la siguiente manera:

- Se hace una breve introducción teórica, a fin de refrescar los conceptos adquiridos en la teoría, necesarios para la resolución de los ejercicios propuestos.
- Se señalan los objetivos perseguidos con cada serie de trabajos prácticos, lo cual evita la dispersión del alumno en cuestiones que no son centrales.
- Se desarrolla en la pizarra o en máquina, utilizando un cañón para proyectar, un ejercicio representativo, a modo de ejemplo.
- Se invita a los alumnos a resolver los ejercicios previstos en la guía, estimulando la resolución de los problemas por sí mismos, resaltando la importancia de que el alumno sea un objeto activo en el proceso de aprendizaje.
- Se recorre el aula de modo de atender las dudas y consultas de aquellos alumnos que generalmente no recurren a los profesores para evacuar sus dudas.
- Una vez que los alumnos resuelven los problemas planteados, se solicitará a uno de ellos que desarrolle una posible solución en el pizarrón. La solución se discute y se analizan los distintos enfoques que se presentan, enriqueciendo los distintos puntos de vista.
- Previo a cada examen parcial, se realizará un repaso de los temas a evaluar, a fin de reforzar los conceptos fundamentales y resaltar los puntos en que suelen presentarse dificultades, y en los que se comenten errores con mayor frecuencia.
- Con posterioridad al examen parcial, se resolverán todos los temas del examen en la pizarra, señalando los errores detectados con mayor frecuencia.



### c) Laboratorio

Para esta actividad el total de alumnos es dividido en la mayor cantidad de grupos posible, para una mejor atención, en función de la disponibilidad de recursos (generalmente entre 6 y 8 grupos).

En los horarios previstos para esta labor, se presentan dos instancias:

- **Práctica en máquina:** los alumnos transcriben el código de la solución escrito en las clases prácticas, con el propósito de continuar con las etapas del desarrollo de software, que inician con la interpretación del diagrama de clases en UML, y que deben compilar, corregir los errores y probar con un conjunto de datos. Para cumplir esta tarea utilizan un entorno de desarrollo (IDE) de uso académico, que satisface los objetivos de esta práctica. Los docentes atienden las consultas, intentando que el propio alumno aprenda a interpretar los mensajes de error.
- **Evaluación:** en las fechas establecidas y publicadas en la planificación al inicio del cuatrimestre, cada alumno en forma individual ejecuta en la computadora los ejercicios resueltos, que no deben presentar errores, y debe responder a las preguntas del docente a cargo, a fin de determinar la comprensión de los conceptos aplicados. El código debe estar documentado con la herramienta provista a tal efecto (Javadoc).

Se prevé implementar otra modalidad de trabajo, consistente en presentar a los alumnos (que trabajarán en grupos de 2 ó 3 integrantes, dependiendo de los recursos del laboratorio y la cantidad de alumnos que cursan) códigos desarrollados por los docentes, que contendrán errores (referidos a los conceptos específicos del paradigma, o de sintaxis, de lógica, etc). La consigna es que los alumnos detecten los errores, y/o agreguen alguna funcionalidad. Con esta práctica se pretende acercar al alumno al ambiente real del desempeño profesional, en el que con frecuencia se deben corregir programas, incluso desarrollados por otras personas. Esto permite a la vez valorar la importancia de seguir ciertas metodologías al momento de la construcción del software, y de la documentación del mismo, lo cual aporta a la legibilidad y extensibilidad del mismo. La elaboración del material necesario para esta práctica será asignada a adscriptos a la cátedra, como parte de su Plan de Trabajo de la adscripción.

### d) Aula Virtual

La introducción y el uso pedagógico de los medios y de las tecnologías de la información y la comunicación (TIC) es uno de los desafíos que enfrentan las instituciones educativas en la dinámica de transformación y modernización de sus prácticas pedagógicas<sup>2</sup>. Existe un consenso generalizado sobre la necesidad o conveniencia de incorporar y utilizar las TIC en la enseñanza.

En esta asignatura se ofrece un Aula Virtual implementada en la plataforma de la UNNE, como complemento al dictado presencial. Brinda la posibilidad de mantener un canal de comunicación permanente con los alumnos, a través del chat y de correo electrónico, así también como de incentivarlos a manifestar sus dudas y a participar en los foros habilitados a tal efecto. Esta facilidad acorta tiempos y distancias, ya que el alumno no debe esperar a la siguiente clase para realizar alguna consulta, lo cual le permite seguir avanzando en su proceso de aprendizaje, a su propio ritmo.

Por otra parte, permite poner a disposición de los alumnos todo el material didáctico requerido para el dictado, consistente en apuntes teóricos, guías de trabajos prácticos y documentos específicos referidos al lenguaje Java. Además cuenta con una sección de sitios sugeridos y bibliografía recomendada, para ampliar conocimientos.

Otra sección permite publicar el programa analítico de la asignatura, la planificación de las actividades y sus fechas establecidas y los criterios de evaluación.

---

<sup>2</sup> [http://tecnologiaedu.us.es/simposio\\_iberoamericano/ponencias/pdf/CO.5.29.pdf](http://tecnologiaedu.us.es/simposio_iberoamericano/ponencias/pdf/CO.5.29.pdf)



**e) Para el aprendizaje autónomo**

- Se pone a disposición del alumno una serie de cuadernillos con la sintaxis del lenguaje Java, y de los conceptos de POO implementados en dicho lenguaje. Por ejemplo, estructuras básicas, mensajes y métodos, colecciones, manejo de archivos, entre otros. Estos cuadernillos son de lectura obligatoria, y su objetivo es que el alumno **aprenda a aprender** un lenguaje de programación por sus propios medios, dado que el informático en su vida profesional con frecuencia se encuentra ante la necesidad de aprender nuevos lenguajes. Se considera que el alumno cuenta con las habilidades necesarias para el aprendizaje autónomo, dado que en asignaturas previas ya ha aprendido otros lenguajes de programación. Por lo tanto, sólo debe aprender la nueva sintaxis.
- En la plataforma virtual se recomiendan sitios de interés, relacionados con cada uno de los temas del programa, a los que el alumno puede acceder para ampliar sus conocimientos.
- En el aula virtual, además de los foros moderados por un tutor, se habilita un foro sólo para alumnos (sin moderador, pero con supervisión), exclusivo sobre cuestiones avanzadas del lenguaje Java, dado que no constituye el objetivo de la asignatura, pero despierta el interés de los alumnos. Este espacio propicia la vinculación horizontal entre pares.

**f) Evaluación**

Singular importancia cobra en el proceso de enseñanza-aprendizaje la evaluación de la marcha del mismo.

Según Palou<sup>3</sup>, la evaluación tiene dos funciones: la evaluación de los procesos de aprendizajes y la evaluación de resultados, con distintas finalidades: la de realimentar el proceso de enseñanza, en el primer caso, y la de acreditación, en el otro.

La evaluación alternativa o educativa, como la denominan diversos autores, focaliza la atención sobre el alumno, según sus propios aprendizajes, proporciona al docente la posibilidad de hacer un seguimiento del proceso de enseñanza-aprendizaje, creando una historia evaluativa respecto del individuo o del grupo, provee información que facilita la acción curricular, y permite a los estudiantes participar en su propia evaluación. "... lo que interesa no es lo mucho que el profesor enseña sino lo que el alumno realmente aprende"<sup>4</sup>. Desde esta mirada, la evaluación está estrechamente ligada a la naturaleza del conocimiento, y no debe ser desligada de él. Debe estar integrada al proceso para asegurar el aprendizaje continuo y oportuno.

Para efectuar una evaluación con las características mencionadas, se debe pensar en instrumentos de evaluación *abiertos* (que evalúen procesos complejos y no memoria mecánica), *globalizadores* (integración de aprendizajes), *confiables* (que permita reunir la evidencia de lo que se desea evaluar, que sean claros respecto a qué se desea evaluar y que los alumnos comprendan qué se les está pidiendo) y *coherentes* (con los criterios explicitados y con el proceso de enseñanza-aprendizaje)<sup>5</sup>.

En consonancia con lo expuesto, en la cátedra se elaboran y utilizan diferentes instrumentos, con distintos objetivos:

- l) Para evaluación de proceso de aprendizaje:** la evaluación debe ser entendida como actividad crítica de aprendizaje, en el sentido que por ella adquieren conocimiento todos los implicados: el profesor aprende para mejorar la práctica docente y para colaborar en el aprendizaje del alumno, consciente de las dificultades que debe superar. El alumno aprende de la propia evaluación y de la corrección, de la información que le ofrece el profesor, que debe ser siempre crítica y argumentada, pero nunca descalificadora ni

---

<sup>3</sup> Palou de Maté, C. *La enseñanza y la evaluación*. (2003)

<sup>4</sup> Mateo, J.A. *La evaluación educativa, su práctica y otras metáforas*

<sup>5</sup> Sanjurjo, L.; Vera, M. *Aprendizaje significativo y enseñanza en los niveles medios y superior*. (1998)



penalizadora. Si la evaluación es un ejercicio continuo, no hay razón para el fracaso, pues siempre se llegará a tiempo para actuar e intervenir en el momento oportuno, cuando el alumno necesita orientación y ayuda para evitar que cualquier error detectado se convierta en definitivo<sup>6</sup>.

Para la evaluación de los procesos se utilizan diferentes instrumentos, en distintos espacios:

- A. **en el aula: lista de cotejo**, concretado en planillas de registro, producidas a lo largo del dictado de la asignatura, consistente en asignar un + ó -, cuando el alumno (seleccionado al azar en la planilla de asistencia) pasa al pizarrón a desarrollar alguna tarea que se les haya indicado. Por otra parte, se alienta a los alumnos a pasar al pizarrón aún cuando no estén seguros de la resolución de sus trabajos, señalándoles que las dificultades que tenga pueden ser comunes a otros compañeros, por lo que puede convertirse en una valiosa instancia de aprendizaje. Es mucho más importante observar que el alumno aborda la solución con una estrategia adecuada, aunque no llegue a un resultado correcto, que el haberlo conseguido de una forma mecánica. Al decir de Palou, "Indagando a partir de los errores se pueden observar las estrategias puestas en juego para resolver un problema, los supuestos sobre los que se basó el alumno para llegar a ese resultado y las relaciones que realizó"<sup>7</sup>.

Esta práctica tiene como objetivo promover y evaluar actitudes, tales como responsabilidad, participación en clase, etc. Los alumnos son informados del objetivo, y de que la valoración tiene que ver con el concepto del alumno, y es útil a la hora de definir una nota en la instancia de evaluación acreditativa.

- B. **en el laboratorio de informática: coloquio** para la defensa de los trabajos prácticos, en máquina. Según Álvarez Méndez, para asegurar el aprendizaje reflexivo de contenidos concretos, quienes aprenden necesitan explicar, argumentar, preguntar, defender sus propias ideas y creencias. Los conocimientos así adquiridos les permiten actuar de modos diferentes en contextos nuevos, no conocidos.

El alumno debe ejecutar los programas desarrollados en respuesta a los problemas planteados, y además se le realizan preguntas sobre el modo en que resolvió, y debe justificar el uso de las distintas técnicas (por ej.: *accessors*, *seudovariab*les, atributo de clase, etc). En la resolución del problema además se verifica que se haya respetado lo especificado en el diagrama de clase en UML (Lenguaje de Modelado Unificado), lo cual demuestra la correcta interpretación de dicho concepto. Esta instancia de evaluación tiene como objetivo detectar tempranamente las dificultades en la comprensión de los conceptos teóricos propios del paradigma, lo cual permite reorientar la enseñanza. De este modo, el alumno se prepara mejor para la instancia de acreditación, ya que al ser individual esta práctica, surgen los problemas de aprendizaje puntuales, y el alumno tiene la posibilidad de expresar sus dificultades. Permite detectar tempranamente las dificultades en la comprensión de los conceptos básicos, sin lo cual no es posible la construcción progresiva del resto de los conceptos.

- C. **en el laboratorio de informática: autoinforme del alumno**, explicitando qué conceptos teóricos usó y dónde los aplicó en el trabajo. El alumno debe presentar un informe en cada coloquio (al finalizar cada serie de trabajos prácticos).

Esta práctica tiene dos propósitos:

- Favorecer en el alumno el desarrollo de habilidades para la comunicación oral y escrita
- Que el alumno tome conciencia de los aprendizajes realizados, lo cual se espera que favorezca la comprensión de los conceptos teóricos llevados a la práctica. Esta

---

<sup>6</sup> Álvarez Méndez, J.M. Evaluar para conocer, examinar para excluir. (2001).

<sup>7</sup> Palou de Maté, C. *La enseñanza y la evaluación*. (2003) pp. 103.





idea surge de lo expresado por dos especialistas en el tema (Alvarez Méndez y Mateo): "Quien aprende tiene mucho que decir de lo que aprende y de la forma en que lo hace. De este modo se puede descubrir la calidad de lo aprendido, las dificultades que encuentra y la naturaleza de las mismas, la profundidad y consistencia de lo aprendido y la capacidad generadora para nuevos aprendizajes". "Evaluar el nivel de logro de algún aprendizaje es un proceso en el que la evaluación y el propio aprendizaje están simbióticamente unidos, se producen en forma paralela, y se retroalimentan continua y respectivamente".

- D. **en el aula y en el hogar: trabajo integrador en equipo**, consistente en el desarrollo de una aplicación de mediana complejidad, que exige el ejercicio de distintos roles en el proceso de desarrollo, simulando una situación del ámbito laboral. En cada grupo se forman subgrupos, que desarrollan diferentes módulos, que luego deben ser integrados de modo que no solo deben funcionar las partes en forma independiente, sino el conjunto una vez reunido. Este trabajo es integrador, en el sentido que requiere la aplicación de todos los conceptos aprendidos. Se realiza al final del cuatrimestre.

II) **Para evaluación de resultados de aprendizaje (examen parcial o final)**: en esta instancia, en general los alumnos están más preocupados por aprobar el examen que por observarse y reflexionar sobre su aprendizaje. "El alumno aprende a dar examen a lo largo de su carrera universitaria ... Consiste en descubrir la manera de sortear con menos dificultad el desafío de ocultar ante su maestro aquellas cosas que no sabe..."<sup>8</sup>.

- A. **examen parcial**: para esta instancia se elaboran enunciados de problemas que integran los conceptos enseñados en los trabajos prácticos. La descripción del problema va acompañada con la especificación del modelo en un diagrama de clases en UML. El alumno debe resolverlo codificando la solución en un programa en el lenguaje java, que es el utilizado en los trabajos prácticos.

Se explicitan los criterios de evaluación, los cuales se corresponden con los objetivos de aprendizaje publicados.

Este instrumento tiene como objetivo calificar y acreditar los aprendizajes (permite regularizar la materia).

Esta instancia de evaluación es aprovechada también como espacio de aprendizaje, realizando una devolución individual fundamentando los logros, obstáculos y necesidades de superación o revisión, buscando la reflexión del alumno para que tome conciencia de su situación en el proceso de aprendizaje. Cuando el momento de la corrección se convierte en un acto de aprendizaje, la evaluación se convierte en una actividad de conocimiento<sup>9</sup>.

- B. **examen final**: exposición oral autónoma del alumno y respuesta a preguntas, también oral, lo cual implica organizar los conocimientos que se poseen y expresarlos utilizando los términos técnicos adecuados (uno de los criterios de evaluación). Permite comprobar directamente la calidad y característica de la respuesta, la fluidez verbal, capacidad para organizar la información, etc.

En las instancias de evaluación son de importancia decisiva los **criterios** adoptados. Por tal motivo, además de figurar en el programa de la asignatura, son expresados permanentemente en forma verbal, desde la clase inaugural, y a lo largo del dictado. Son publicados también en el Aula Virtual gestionada por la cátedra.

Los **criterios de evaluación** son:

- Capacidad para resolver los problemas planteados utilizando las técnicas de la programación orientada a objetos

---

<sup>8</sup> Bohoslavsky en Sanjurjo, L.; Vera, M. *Aprendizaje significativo y enseñanza en los niveles medios y superior*.

<sup>9</sup> Álvarez Méndez, J.M. *Evaluar para conocer, examinar para excluir*. (2001).



- Correcta interpretación del diagrama de clases en UML
- El programa que representa la solución debe ser ejecutado sin errores
- Capacidad de integración de los distintos conceptos impartidos
- Utilización del lenguaje técnico apropiado
- Claridad en la exposición, capacidad de síntesis y de relación de los distintos conceptos.

Es necesario además definir modalidades de **seguimiento del proceso de enseñanza-aprendizaje**. Se propone:

- **Ficha individual de seguimiento:** con la identificación del alumno (nombre y foto), grupo de práctico y laboratorio. En ella se asienta la fecha de evaluación y aprobación de cada trabajo práctico en el laboratorio, y las observaciones que el docente considere pertinentes, para un mejor seguimiento de los logros y dificultades.
- **Planilla de cálculo** con el registro de las notas de parciales y laboratorio, de modo de tener un conocimiento integral del desempeño de cada alumno. Esta planilla permite además obtener en forma automática al final del dictado la condición del alumno.

## 2.3. REGIMEN DE PROMOCION:

### Condiciones para Regularizar la Materia.

75% de asistencia a las clases prácticas

Aprobación de los trabajos prácticos de Laboratorio.

Aprobación de los dos exámenes parciales con nota mayor o igual a 6.

Cada parcial tiene su correspondiente recuperatorio y está previsto un examen extraordinario para recuperar uno de los dos parciales.

### Condiciones para promocionar la Materia (Sin Examen Final).

Los alumnos **regulares** cuyo promedio de notas de parciales sea mayor o igual a 8 y con el 75% de asistencia a las clases teóricas, aprobarán la materia sin examen final. Para el promedio se considerarán los parciales aprobados en cualquier instancia. No se computan los parciales desaprobados.

### Condiciones Para Aprobar la Materia Con Examen Final.

- Los alumnos **regulares** rinden un examen oral ó escrito sobre los contenidos teóricos del programa vigente.
- Los alumnos **libres** deben aprobar un examen práctico para luego rendir el examen teórico.

## 2.4 PROGRAMA ANALITICO

### 2.4.1. CONTENIDOS POR UNIDAD:

#### Unidad I: Conceptos de Programación Orientada a Objetos

##### Tema 1: Introducción a la POO.

Introducción. La complejidad del software. La crisis del software. Ingeniería del Software. Factores de calidad del software. La orientación a objetos. Características de la programación orientada a objetos: abstracción, encapsulamiento, modularidad, jerarquía. Beneficios de la programación orientada a objetos. Lenguajes de programación orientados a objetos.



### **Tema 2: Concepto de Objeto y Clase.**

Concepto de Objeto. Propiedades de los Objetos. Objeto y encapsulación. Interfaz pública. Interacción entre objetos: mensajes. Concepto de Clase. Propiedades de una clase: atributos y métodos. Instancias de una clase. Encapsulación – Visibilidad. Implementación de una clase en Java. Métodos Constructores. La Programación Orientada a Objetos.

### **Tema 3: La documentación en el desarrollo del software.**

Introducción. Documentación externa. Modelado. Modelo. Lenguaje de modelado. UML. Diagramas. Documentación Interna. El estilo de programación - Reglas de buen estilo. Los Comentarios. Qué documentar. Tipos de Comentarios. Javadoc. Nomenclatura. Identación.

### **Tema 4: Objeto, Clase, UML y Java: un enfoque práctico**

Objeto. Comportamiento. ¿Qué hace? ¿Cómo lo hace? Implementación: método vs. mensaje. El estado interno. Envío de mensajes. Formato de mensajes. Métodos. Especificación de un método. Observadores y mutadores. Objeto y clase. Clase y UML. Envío de mensajes considerando el concepto de clases. Atributos y mensajes de clase. Formas de conocimiento. Seudovariable this.

### **Tema 5: Reutilización: Clases predefinidas**

Colecciones/Contenedores. Colecciones en los lenguajes OO. Tipos de colecciones: homogéneas/heterogéneas, estáticas/dinámicas, ordenadas/no ordenadas. Operaciones con colecciones. Colecciones en Java. Jerarquía de Interfaces y de Clases de la Java Collections Framework. Persistencia de objetos. Separación de la capa de acceso a datos. Persistencia y Bases de Datos Relacionales (BDR). Bases de datos Orientadas a Objetos (BDOO). Lenguajes orientados a Objeto y la persistencia. Clases para entrada/salida en Java. Clases para tratar fechas en Java.

### **Tema 6: Objetos y Eventos**

Programación secuencial, interactiva y orientada por eventos. Eventos y programación. Programación guiada por Eventos y POO. POO e interfaz gráfica de usuario (GUI). Interfaz gráfica en Java. AWT/Swing.

### **Tema 7: Mecanismos de abstracción. Herencia.**

Introducción. Mecanismos de abstracción. Reutilización y mecanismos de abstracción. Herencia. Herencia simple. Notación UML. Redefinición. Constructores y destructores considerando herencia. Herencia múltiple. Interfaces. Herencia y reutilización. Mecanismo de respuesta a mensajes en una jerarquía de clases. Herencia y ocultación de información

### **Tema 8: Polimorfismo.**

Introducción. Aplicación del Polimorfismo. Ventajas del polimorfismo. Redefinición. Sobrecarga y redefinición. Ligadura dinámica. Métodos virtuales. Clases y métodos abstractos. Notación UML. Transformación de tipos en tiempo de ejecución. Transformación implícita y explícita. Información de tipos en tiempo de ejecución.

## **Unidad II: Conceptos de los lenguajes de POO**

### **Tema 9: Tratamiento de problemas en tiempo de ejecución: Excepciones**

Comprobación de estados. Enfoques conservadores. Enfoques optimistas. Errores y excepciones. Formas de tratar excepciones. Lanzamiento de excepciones. Captura de excepciones: manejadores. Atributos y métodos en excepciones creadas por el programador. Jerarquía de excepciones en Java.

### **Tema 10: Concurrencia y programación orientada a objetos**

Conceptos generales. Concurrencia. Procesos e hilos. Concurrencia y lenguajes de programación. Hilos en Java. Ciclo de vida de un hilo. Threads Demonios. Sincronización. Tareas programadas. Prioridades. Comunicación entre hilos.



#### **BIBLIOGRAFIA:**

- Apuntes Teóricos Programación Orientada a Objetos. Cristina L. Greiner.
- Orientación A Objetos - Diseño y Programación. Carlos Fontela. Nueva Librería. 2008
- Orientación a Objetos con Java y UML. Carlos Fontela. Nueva Librería. 2004
- Cómo programar en Java. 7ma. Ed. Deitel y Deitel. Pearson Addison-Wesley. 2008.
- Programación Orientada a Objetos. Técnicas avanzadas de programación. Carlos Fontela. Nueva Librería. 2003
- Piensa en Java. 4ta. Ed. Bruce Eckel. Pearson Alhambra. 2007
- Ingeniería de Software Orientada a Objetos con UML, Java e Internet. Alfredo Weitzenfeld. Editorial Thomson. 2005
- JAVA. La Guía total del Programador. Sergio Dos Santos. MP Ediciones. 2005
- JAVA. Manual de usuario y tutorial. Agustín Froufe. Alfaomega – Ra-ma. 2002.
- Programación Orientada a Objetos: Conceptos, Modelado y Codificación en C++. Luis Joyanes Aguilar. Mc Graw-Hill. 1996
- Construcción de software orientado a objetos. Bertrand Meyer. 2da. Ed. Prentice-Hall. 1999
- Cómo programar en C++. 2da. Ed. Deitel y Deitel. Pearson Educación. 1999.

#### **3. - PROGRAMA DE EXAMEN:**

<b>Bolilla</b>	<b>Temas</b>
<b>1</b>	1 – 4 – 7
<b>2</b>	2 – 5 – 8
<b>3</b>	3 – 6 – 9
<b>4</b>	1 – 4 – 8
<b>5</b>	3 – 6 – 9
<b>6</b>	2 – 7 – 10
<b>7</b>	4 - 5 – 7
<b>8</b>	8 – 4 – 9
<b>9</b>	1 – 8 – 7
<b>10</b>	2 – 6 – 10

#### **4. - NOMINA DE TRABAJOS PRACTICOS:**

<b>Nro. TP</b>	<b>Tema/Objetivo</b>
<b>1</b>	Introducción a la sintaxis del lenguaje Java. Familiarización con el entorno de trabajo (BlueJ).
<b>2</b>	Definición de Clases. Constructores. Encapsulamiento. Doble encapsulamiento. Instanciación. Interacción entre objetos.
<b>3</b>	Conocimiento entre objetos. Tipos.
<b>4</b>	Reutilización de clases predefinidas: Archivos.
<b>5</b>	Conocimiento múltiple: Colecciones.
<b>6</b>	Herencia. Polimorfismo.
<b>7</b>	Programación Guiada por Eventos: Interfaz Gráfica de Usuario.
<b>8</b>	Tratamiento de excepciones.
<b>9</b>	Concurrencia.
<b>10</b>	Trabajo Práctico Integrador en Equipo.

En la planificación que se elabora para cada dictado de la asignatura se describe con más detalle cada una de las series. Esto permite una actualización dinámica de las guías de trabajos prácticos.



## 5. - CRONOGRAMA DE ACTIVIDADES:

(Incluir clases teóricas, prácticas, de laboratorio, de campo, de seminario, y toda otra actividad educativa prevista como así también evaluaciones parciales).

- 15 Clases teóricas (una por semana)
- 15 clases prácticas en aula (1 por semana)
- 15 clases prácticas en laboratorio (1 por semana)
- 2 exámenes parciales, 2 exámenes recuperatorios, 1 examen extraordinario

Las actividades se describen con mayor detalle en la planificación que se elabora para cada dictado de la asignatura. Esto permite programar las actividades en función del calendario académico y de los recursos disponibles.

## 6. - EFECTOS SOBRE la formación integral del alumno.

(Indicar los beneficios que obtendría el alumno al finalizar el cursado de la materia.)

Se espera que al completar esta materia, el alumno:

- Comprenda los conceptos que sustentan la Programación Orientada a Objetos.
- Sea capaz de aplicar estos conceptos en el desarrollo de programas mediante el uso de un lenguaje orientado a objetos.
- Aprecie las ventajas que ofrece la POO en el desarrollo del software.
- Adquiera la habilidad de documentar el software y valore la importancia de la misma.
- Maneje con soltura los términos técnicos asociados al paradigma.
- Adquiera competencias genéricas tales como responsabilidad, autonomía de estudio, ética profesional, trabajo en equipo, buenas prácticas de programación.

## 7. - RECURSOS HUMANOS.

### 7.1. NOMINA DE PERSONAL DOCENTE INTERVINIENTE EN EL DICTADO DE LA ASIGNATURA

APELLIDO Y NOMBRES	CARGO	DEPARTAMENTO/ AREA	MAXIMO TITULO ACADEMICO OBTENIDO	TIEMPO DEDICADO (a esta asignatura)
Greiner, Cristina L.	Adjunta	Informática/ Programación	Mgter. en Informática y Computación	15 hs. semanales
Mettini, Aldo J.	JTP	Informática/ Programación	Licenciado en Sistemas	10 hs. semanales
Ringa, Mónica I.	JTP	Informática/ Programación	Licenciado en Sistemas	10 hs. semanales
Medina, Yanina	JTP	Informática/ Programación	Licenciado en Sistemas	10 hs. semanales
Arquero, Mario A.	Aux.Primer	Informática/ Programación	Experto en Estadística y Computación	10 hs. semanales
Barth Müller, Carina L.	Aux.Primer	Informática/ Programación	Licenciado en Sistemas	10 hs. semanales

### 7.2. NOMINA DE PERSONAL DOCENTE ADSCRIPTO INTERVINIENTE EN EL DICTADO DE LA ASIGNATURA

Se detalla en la planificación que se elabora para cada dictado de la asignatura, debido a que el plantel de adscriptos varía año a año.