

Programa de Cátedra

 <p>UTN FACULTAD REGIONAL MENDOZA</p>	Asignatura:	Departamento:
	Bloque: Disciplinas Tecnológicas	Área: Laboratorio de Computación I
	Régimen: Primer Semestre	Horas semanales: 6
	Tipo: Troncal	Horas semestrales/anuales: 90
	Carrera: Tecnicatura Universitaria en Programación	Nivel (Año):
	Ciclo lectivo: 2020	<input checked="" type="checkbox"/> 1° <input type="checkbox"/> 2° <input type="checkbox"/> 3° <input type="checkbox"/> 4° <input type="checkbox"/> 5° <input type="checkbox"/> 6°

Integrantes de la Cátedra:

- **Profesor Titular:**

Cristina Domizio

- **Profesor Asociado:**

Gerardo Magni

- **Profesor/es Adjunto/s:**

Nombre del Profesor	Tipo de Dedicación	Cantidad de Dedicaciones
Ninguno		

--	--	--

- **Auxiliares de Docencia:**

Nombre del Profesor	Categoría	Tipo de Dedicación	Cantidad de Dedicaciones
Ninguno			

a) Fundamentación de la materia dentro del plan de estudios

La estructura curricular del espacio bajo análisis dispone de 6 horas-cátedra semanales. Se propone un aula-taller que básicamente persigue el objetivo de incrementar en el alumno su natural habilidad para utilizar el pensamiento elaborado lógicamente a fin de resolver problemas a través de la descomposición de los mismos, en sub-problemas más simples (diseño descendente), estableciendo la relación entre el todo y las partes (enfoque sistémico), considerando sólo los aspectos relevantes (abstracción), mediante el uso de herramientas fundamentales representativas de los procesos deductivos asociados a la programación, integrando la sintaxis elemental de un lenguaje de programación en Laboratorio. A su vez se intenta modelar un ambiente creativo de niveles crecientes de realismo respecto del mundo laboral pertinente. Esto nos lleva a establecer la metodología de carpeta abierta y uso de documentación en soportes y en línea.

En definitiva el espacio curricular es uno de los más propicios para lograr “El desarrollo del pensamiento reflexivo, del juicio crítico, de la creatividad y de la capacitación para el auto-aprendizaje”, y a la vez, propiciar el derecho del alumno a recibir “formación científica, profesional, humanística y técnica en el más alto nivel, contribuir a la preservación de la cultura nacional, promover la generación y desarrollo del conocimiento en todas sus formas, y desarrollar las actitudes y valores que requiere la formación de personas responsables, con conciencia ética y solidaria, reflexivas, críticas, capaces de mejorar la calidad de vida, consolidar el respeto al medio ambiente, a las instituciones de la República y a la vigencia del orden democrático”.

Es obvio, luego, el rol de auxiliar y complementaria de esta asignatura, posibilitando un enfoque más práctico y orientado al uso de herramientas presentes en un entorno integrado de desarrollo (IDE por sus siglas en inglés).

Laboratorio I permite desarrollar estrategias y codificación de problemas computacionales utilizando lenguajes de programación.

El desarrollo e implementación de distintos algoritmos permite lograr la primera aproximación del programador a situaciones reales de experiencia que, en un futuro, le permitirán desarrollar algoritmos más complejos y orientados a la lógica de negocios que necesite resolver.

b) Objetivos de la materia

- **Objetivos Generales:**

La cátedra se enfoca en los siguientes objetivos generales:

- Resolver problemas complejos de programación y la arquitectura aplicada para la mejor solución.
- Conocer tecnologías, paradigmas, metodologías y herramientas de última generación que se aplican en el desarrollo de software.
- Motivar las buenas prácticas de desarrollo en función de los estándares de calidad.
- Relacionar los saberes adquiridos en análisis y diseño de sistemas, como así en programación funcional y de objetos, adaptándolos a las necesidades solicitadas para un perfil de desarrollador Full Stack.
- Conocer herramientas de productividad que acompañan el proceso de desarrollo de software.

- **Objetivos Específicos:**

La cátedra se enfoca en varios objetivos a conseguir por parte del alumno:

- Buena interpretación de las consignas propuestas para resolver mediante la programación.
- Elaborar soluciones creativas a las consignas propuestas.
- Adquirir principios de buen diseño de software.
- Organización y planificación del trabajo, para la resolución de problemas complejos.
- Desarrollo de la comunicación, de la curiosidad investigativa y del trabajo en grupo.

- Aplicar la mejor arquitectura de sistemas que se adecue a la solución solicitada.
- Determinar el correcto uso de patrones para el desarrollo de software en el Front End y Backend.
- Seleccionar el mejor esquema de persistencia para brindar la solución más efectiva.

c) Contenidos Mínimos ()

a. Algoritmos: Programa

- Estilos de programación (Estructurada – Orientada a objetos).
- Estructura de un programa Java.
- Tipos de datos en Java.
- Secuencia: Declaración de variables.

- Métodos. Instrucciones en Java.
- Decisiones: Simples. Alternativas. Múltiples. Instrucciones en Java.
- Iteraciones: Bucles. Anidamientos.
- Array: Asignación. Almacenamiento . Recorrido.
- Ordenamiento. Búsquedas.
- Conceptos Avanzados: Recursividad.
- Introducción a la programación orientada a objetos. Noción de clases.
- Introducción a listas, pilas, colas.
- Conceptos generales: Entorno de desarrollo IDE. Depuración. Stepping. Break-point. Librerías. Packages.
- Apoyo práctico a la cátedra Programación I.

d) Programa Analítico

UNIDAD TEMÁTICA	CONTENIDOS	Referencia bibliografía (Nro.)
1 Algoritmos	<p><u>Objetivo :</u> Conocer las herramientas utilitarias que se usan en un Software Factory y las distintas plataformas que se usan en el desarrollo de software.</p> <p><u>Contenido :</u></p> <ol style="list-style-type: none"> 1. Problema. Contexto. Datos asociados. 2. Comprensión de problemas y metodología general de resolución. 3. Nociones de acción, estado y secuencia. 4. Algoritmo. Concepto. Elementos. Características. Tipos de algoritmos. 5. Programa. Concepto y características. Instrucción. Elementos. Constantes y variables simples. Tipos de datos. 6. Conocimiento del entorno de desarrollo. 7. Expresiones numéricas y alfanuméricas. Expresiones condicionales: relacionales y lógicas. Tablas de verdad. 	Apuntes de cátedra Y 3

UNIDAD TEMÁTICA	CONTENIDOS	Referencia bibliografía (Nro.)
2 Estructuras secuenciales	<p><u>Objetivo :</u> Establecer los conocimientos necesarios de las distintas estructuras de programación secuenciales y de operaciones, requeridas en un programa.</p> <p><u>Contenido :</u></p> <ol style="list-style-type: none"> 1. Conceptos. 2. Representación gráfica. 3. Declaración de variables. Asignación, lectura y escritura. 4. Pre-condiciones (estado inicial requerido) y pos-condiciones (estado final provocado). 	2 - 3
3 Estructuras de decisión.	<p><u>Objetivo :</u> Establecer y asegurar la estructura de la toma de decisiones dentro de un programa, y asegurar el funcionamiento correcto para cada caso particular, analizando el dato conocido y el desconocido. Armado correcto de afirmaciones lógicas.</p> <p><u>Contenido :</u></p> <ol style="list-style-type: none"> 1. Concepto. 2. Representación gráfica con diagramación de flujo y estructurada (Chapín). 3. Composición condicional (decisión simple) y alternativa (decisión doble). Decisiones 	2 – 3 - 4

UNIDAD TEMÁTICA	CONTENIDOS	Referencia bibliografía (Nro.)
	múltiples (alternativas anidadas y composición selectiva). 4. Pre y pos-condiciones de cada una.	
4 Estructuras cíclicas	<p>TECNOLOGIAS DE PROGRAMACION DEL LADO DEL SERVIDOR</p> <p><u>Objetivo :</u> Establecer el conocimiento del funcionamiento de la estrucura repetitiva, estableciendo la condición adecuada según la necesidad para el cumplimiento del objetivo final.</p> <p><u>Contenido :</u></p> <ol style="list-style-type: none"> 1. Conceptos. Representación gráfica. 2. Bucles lógicos: Control de inicio de Bucle (Mientras-Fin Mientras, Para- Fin Para) y Control de fin de Bucle (Repetir – Hasta que). 3. Pre y pos-condiciones de cada una de ellas. Definición de invariante de un bucle. 	<p>6,7</p> <p>2 -3 -4 -5</p>
5 Recursión	<p><u>Objetivo :</u> Conocer y construir la estructura general de un programa, y lograr agrupar acciones repetitivas en bloques, para que sean llamados en forma recursiva, evitando el código redundante y logrando la robustez y la reducción de código innecesario.</p> <p><u>Contenido :</u></p> <ol style="list-style-type: none"> 1. Subprogramas: concepto, argumentos, parámetros. 2. Pasaje de argumentos por valor y por referencia. 3. Noción de llamada a procedimientos y funciones. 	<p>1- 2- 3- 4- 5</p>

UNIDAD TEMÁTICA	CONTENIDOS	Referencia bibliografía (Nro.)
6 Variables dimensionadas	<p><u>Objetivo :</u> Conocer y construir la variable con una o más dimensiones. Utilizar la dimensión para mejorar la performance del programa, Determinar la necesidad de la dimensión para agilizar el funcionamiento y la efectividad de los resultados, basándose en los índices como iniciación de su utilización.</p> <p><u>Contenido :</u></p> <ol style="list-style-type: none"> 1. Concepto. Dimensión e índices. 2. Declaración. Inicialización. Utilización. 3. Operaciones asociadas. 4. Ordenamiento y Búsqueda. 5. Variables multi-dimensionadas. 	1 – 2 – 3 – 4 – 5 – 6 – 7 - 8
7 Colecciones	<p><u>Objetivo :</u> Las colecciones son una especie de arrays de tamaño dinámico. Cuando queremos trabajar con un conjunto de elementos, necesitamos un almacén donde poder guardarlos. En Java, se emplea la interfaz genérica Collection para este propósito. Gracias a esta interfaz, podemos almacenar cualquier tipo de objeto y podemos usar una serie de métodos comunes, como pueden ser: añadir, eliminar, obtener el tamaño de la colección, etc.</p> <p><u>Contenido :</u></p> <ol style="list-style-type: none"> 1. Concepto. 2. Declaración. Inicialización. Utilización. 3. Operaciones asociadas. 4. Collection<E> 5. List<E> (ArrayList<E>, Vector<E>, LinkedList<E>) 	

UNIDAD TEMÁTICA	CONTENIDOS	Referencia bibliografía (Nro.)
	6. Set<E> 7. Map<K,V>	

e) Programa de Examen

Idem programa Analítico.

f) Trabajos Prácticos

Unidad a la que corresponde	Título del trabajo práctico/actividad de laboratorio/taller/etc.	Objetivo	Temas a aplicar/cubrir según programa (Nro.)
1	Práctico 1: Toma de decisiones	Construir afirmaciones lógicas con toda clase de operadores	1.1 al 1.7 1.2
2 - 3	Práctico 2. : Decisiones y Menú	Atribuir una actividad diferenciada según la toma de decisiones	2.1 al 2.4 3.1 al 3.4
4	Práctico 3: Bucles	Identificar acciones repetitivas dentro del código.	4.1 al 4.3

3 - 4	Práctico 4: Diagramación Lógica y estructurada	Realizar instalación y configuración avanzada del archivo POM	3.2, 4.2 y 4.3
5	Práctico 5: Procedimientos y funciones	Identificar acciones repetitivas dentro del código, y construir código independiente, determinando datos conocidos y desconocidos.	5.1 al 5.3
6	Práctico 6: Variables dimensionadas	Determinar la dimensión necesaria para variables, para mejorar el rendimiento del código	6.1 al 6.5
6	Práctico 7: Ordenamiento	Conocer los métodos existentes necesarios para el ordenamiento de variables dimensionadas	6.3 y 6.4
7	Práctico 8: Colecciones	Aplicar colecciones para la manipulación dinámica de datos.	7

g) Distribución de horas

Formación teórica	30
Formación experimental	30
Resolución de problemas de ingeniería	15
Proyecto y diseño	15

h) Correlativas

Para cursar		Para rendir
Tener regulares	Tener Aprobadas	Tener aprobadas
<ul style="list-style-type: none"> Ninguna 	<ul style="list-style-type: none"> Ninguna 	<ul style="list-style-type: none"> Ninguna

i) Bibliografía

Nro.	Autor/es	Título	Editorial	Año de edición	Principal	Complementaria
1	Deitel & Deitel	Java How to Program 7/e	Prentice Hall.	2008		x
2	Fco. Javier Ceballos	JAVA 2: Curso de Programación	Alfaomega-Rama		x	
3	Docente	Apuntes entregados			x	

4	Patrick Naughtn	Manual de JAVA	MC Graw Hill		x	
5	F Lima Diaz	Manual Avanzado de Java	Anaya			x
6	Thomas Wu	Introducción a la Programación Orientada a Objetos con Java	Mc Graw Hill			x
7	Bruce Eckell	Pensando en Java	Prentice Hall.			x
8	Patrick Naughtn	Manual de Java	Mc Graw Hill			x

10- SITIO WEB OFICIAL

- Java - Oracle: <https://www.java.com/es/>
- NetBeans: <https://netbeans.org/>

