

Programa de Cátedra

Para el envío electrónico, nombrar el archivo programa_planificación_asignatura

	Asignatura:	Departamento:
	Bloque: Disciplinas Tecnológicas	Área: Programación II
	Régimen: Segundo Trimestre	Horas semanales: 6
	Tipo: Troncal	Horas semestrales/anuales: 96
	Carrera: Tecnicatura Universitaria en Programación	Nivel (Año):
	Ciclo lectivo: 2023	<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:**
- **Profesor Asociado:**
- **Profesor/es Adjunto/s:**

Nombre del Profesor	Tipo de Dedicación	Cantidad de Dedicaciones
Martín Julio Vargas	Adjunto	1
Cristina Domizio	Adjunto	1

- **Auxiliares de Docencia:**

a) Fundamentación de la materia dentro del plan de estudios

La cátedra de Programación II se dicta dentro del segundo semestre de la carrera, tiene como objeto desarrollar habilidades complejas con el lenguaje de programación de JAVA, basando su enseñanza dentro del paradigma de la orientación a objetos, el cual es independiente de la tecnología aplicada.

De este modo el alumno aprende a realizar un proceso de abstracción a través de la teoría de objetos y materializa dichos contenidos en el lenguaje de programación antes mencionado.

El hecho de utilizar JAVA, permite al alumno manipular unos de los lenguajes de programación más utilizados en el mercado mundial, esto se debe entre otras cosas a su gran plataforma tecnológica y a la gran masa de soporte mundial.

Conocer y dominar las últimas tecnologías empleadas en el mercado laboral sobre el Front End y Back end comúnmente utilizadas en la actualidad para la construcción de software, como así también aquellas que podrían ser utilizadas en el futuro, sentando las bases para la investigación posterior por parte del alumno devenido en profesional de la materia.

b) Objetivos de la materia

- **Objetivos Generales:**

La cátedra de enfoca en los siguientes objetivos generales:

- Enseñar un lenguaje de programación aplicando la teoría universal de objetos.
- Realizar la ingeniería directa, que consiste en llevar a la codificación de un lenguaje formal en el modelo analizado.
- Manipular la arquitectura aplicada para la mejor solución.
- Conocer la tecnología, paradigma, metodología 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.
- Aplicar prácticas del diseño de sistemas, que llevan a los programadores a realizar la división de los mismos en cuatro capas.
- Manipular herramientas de productividad que acompañan el proceso de desarrollo de software.

- **Objetivos Específicos:**

- Comprender notación UML.
- Producir software a través de la interpretación de diagramas de clases.
- Administrar entornos de desarrollo como NetBeans y Eclipse.
- Interpretar, integrar, demostrar y aplicar los conceptos subyacentes en el paradigma OO.
- Desarrollar aplicaciones gráficas simples (SWING).
- Conocer la aplicación práctica del paradigma funcional y de objetos, aplicado a librerías y frameworks que se utilizan en el desarrollo de software en Front End y Backend respectivamente.
- Administrar Contenedores para el manejo funcional adecuado en la manipulación de valores diversos.
- Comprender las ventajas del manejo de errores en tiempo de ejecución.
- 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.
- Aplicar técnicas de desarrollo según modelos de diseño (MVC y 4 capas con DAO).
- Seleccionar el mejor esquema de persistencia para brindar la solución más efectiva.
- Aplicar concepto de marcos de trabajo (frameworks).

c) Contenidos Mínimos ()

Nociones de Diagramas de UML. Principios de la Orientación a Objetos. Encapsulación. Herencia. Clase Abstracta. Polimorfismo. Tipos de Asociaciones. Unidireccional. Bidireccional. Composición. Agregación. Ingeniería directa aplicada al lenguaje Java. Programación Concurrente. Estructura de datos y Framework de Colecciones y arreglos de objetos. Interfaz gráfica Swing. Nociones de bases de datos relacionales. Tipos de driver JDBC. Aplicaciones en 4 capas. Presentación – Negocio – Controlador – Acceso a datos. Trabajo integrador.

d) Programa Analítico

UNIDAD TEMÁTICA	CONTENIDOS	Referencia bibliografía (Nro.)
<p>1</p> <p><u>Paradigma de Orientación a Objetos.</u></p>	<p><u>Objetivo:</u></p> <ul style="list-style-type: none"> - Interpretar, integrar, demostrar y aplicar los conceptos subyacentes en el paradigma OO. - Comprender notación UML. - Producir software a través de la interpretación de diagramas de clases. - Administrar entornos de desarrollo como NetBeans y Eclipse. <p><u>Paradigma de Orientación a Objetos:</u></p> <ol style="list-style-type: none"> 1- Introducción a la POO y aplicaciones del paradigma al lenguaje JAVA. 2- Abstracción – Encapsulamiento – Ocultamiento. 3- El modelo orientado a objetos y la Ingeniería directa Objetos y UML. 4- Clases – Atributos – Constructores – Métodos - Mensajes Herencia – Polimorfismo – Interfaz. 5- Asociaciones: Unidireccionales – Bidireccionales. Agregación – Composición 	<p>Apuntes de cátedra</p>
<p>2</p> <p><u>Estructuras de memoria estáticas, dinámicas y manejo de excepciones.</u></p>	<p><u>Objetivo:</u></p> <ul style="list-style-type: none"> - Conocer la aplicación práctica del paradigma funcional y de objetos, aplicado a librerías y frameworks que se utilizan en el desarrollo de software en el Backend. - Administración de Contenedores para el manejo funcional adecuado en la manipulación de valores diversos. - Comprender las ventajas del manejo de errores en tiempo de ejecución. <p><u>Estructuras de memoria estáticas y dinámicas</u></p> <ol style="list-style-type: none"> 1- Arreglos primitivos – Arreglos de Objetos. 2- Estructuras Dinámicas el Framework de Colecciones: Colecction , List , Set, Map. <p><u>Excepciones</u></p> <ol style="list-style-type: none"> 3- Lanzamiento de Excepciones. 4- Try, Catch y Finally. 	<p>Apuntes de cátedra</p>
	<p><u>Objetivo:</u></p> <ul style="list-style-type: none"> - Aplicar la mejor arquitectura de sistemas que se adecue a la solución solicitada. 	<p>Apuntes de Cátedra</p>

UNIDAD TEMÁTICA	CONTENIDOS	Referencia bibliografía (Nro.)
<p align="center">3</p> <p><u>Nociones fundamentales sobre interfaces Swing. Manejo de patrón de diseño.</u></p>	<ul style="list-style-type: none"> - Manipular una interfaz gráfica de usuario. - Identificar los diferentes componentes y sus funcionalidades. - Determinar el correcto uso de patrones para el desarrollo de software en el Front End y Backend. - Aplicar técnicas de desarrollo según modelos de diseño Modelo-Vista-Controlador (MVC). <p><u>Nociones fundamentales sobre interfaces Swing:</u></p> <ol style="list-style-type: none"> 1- Interfaces de usuario controladas por eventos y manejo de eventos en Swing - Ejemplo sencillo: botón en un cuadro Contenedores: - JFrame, JDialog, JPanel y JScrollPane. 2- Componentes: JComponent , JLabel y JButton , JToggleButton y JRadioButton , Deslizadores y controles numéricos , JTextField, JTextArea , JTable . 3- Diseños : FlowLayout, BorderLayout, GridLayout y BoxLayout 4- Menús, acciones y barras de herramientas: JMenu y JMenuItem ,JToolBar y Action. <p><u>Diseño de Patrón de desarrollo de software</u></p> <ol style="list-style-type: none"> 5- Patrones de diseño. MVC. 6- Capas, funciones y distribución del software. 	<p>Apuntes de cátedra</p>
<p align="center">4</p> <p><u>Acceso a datos a través a JDBC.</u></p>	<p><u>Objetivo:</u></p> <ul style="list-style-type: none"> - Seleccionar el mejor esquema de persistencia para brindar la solución más efectiva. <p><u>Acceso a datos a través a JDBC</u></p> <ol style="list-style-type: none"> 1. Introducción a las bases de datos relacionales – MySQL 2. Noción de entidades. Relaciones. Multiplicidad máxima y mínima. 3. Definición de JDBC – Seleccionar una base de datos – Establecer la conexión – Seleccionar una tabla - Recuperar Valores desde una Hoja de Resultados - Actualizar Tablas - Utilizar Sentencias Preparadas - Crear Aplicaciones JDBC Completas. 4. Manejo del Drivers. 	<p>Apuntes de cátedra</p>
<p align="center">5</p>	<p><u>Objetivo:</u></p> <ul style="list-style-type: none"> - Aplicar la mejor arquitectura de sistemas que se adecue a la solución solicitada. 	<p>Apuntes de cátedra</p>

UNIDAD TEMÁTICA	CONTENIDOS	Referencia bibliografía (Nro.)
<u>Introducción al modelado en 4 capas.</u>	<ul style="list-style-type: none"> - Determinar el correcto uso de patrones para el desarrollo de software en el Front End y Backend. - Aplicar técnicas de desarrollo según modelos de diseño (MVC y 4 capas con DAO). - Aplicar concepto de marcos de trabajo (frameworks). <p><u>Introducción al modelado en 4 capas</u></p> <p>1- La capa de la vista, negocio, control y modelo.</p> <p>2- Ventajas- cohesión – acoplamiento. – Independencia de las clases.</p> <p>3- Desarrollo de una aplicación integral modelando en 4 capas.</p>	

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: Programación de diferentes diagramas de clase.	<p>Conocer los sitios oficiales de descarga y los procesos de instalación del software de base para el desarrollo.</p> <p>Realizar el traspaso de diagrama a código.</p> <p>Reconocer y aplicar las distintas relaciones UML a código, en el framework correspondiente.</p>	1 – 2 – 3 – 4 – 5
1	Práctico 2: Responder test teóricos prácticos sobre el paradigma y la codificación.	Realizar un test que contribuya a integrar conceptos y formas de codificación.	1 – 2 – 3 – 4 – 5
1	Práctico 3: Práctica de Interfaces y Clases Abstractas	Comprender las similitudes y diferencias entre clases abstractas e interfaces.	3 – 4 – 5

		Manejar las potencialidades de las interfaces y aplicarla en la codificación.	
2	Práctico 4: Arreglos y contenedores de JAVA.	Lograr una administración de contenedores para el manejo funcional adecuado en la manipulación de valores diversos.	1 – 2
3	Práctico 5: Codificación de proyecto de Interfaz Gráfica (SWING)	Manipular una interfaz gráfica de usuario. Identificar los diferentes componentes y sus funcionalidades. Determinar el correcto uso de patrones para el desarrollo de software en el Front End	1 – 2 – 3 – 4
3	Práctico 6: Codificación de proyecto del patrón MVC.	Determinar el correcto uso de patrones para el desarrollo de software en el Front End y Backend. Aplicar técnicas de desarrollo según modelos de diseño Modelo-Vista-Controlador (MVC).	5 – 6
4	Práctico 7: Relaciones entre entidades (DER).	Comprender las diferentes relaciones entre entidades	1 – 2
4	Práctico 8: Ejercicio de Base de datos Relacional (SQL en MySQL).	Seleccionar el mejor esquema de persistencia para brindar la solución más efectiva.	3 – 4
5	Práctico 9: Proyecto integrador en 4 capas	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.	1 – 2 – 3

		<p>Aplicar técnicas de desarrollo según modelos de diseño (MVC y 4 capas con DAO).</p> <p>Aplicar concepto de marcos de trabajo (frameworks).</p>	
--	--	---	--

g) Distribución de horas

Formación teórica	20
Formación experimental	21
Resolución de problemas de ingeniería	25
Proyecto y diseño	30

h) Correlativas

Para cursar		Para rendir
Tener regulares	Tener Aprobadas	Tener aprobadas
<ul style="list-style-type: none"> Programación I Laboratorios de Computación I 	<ul style="list-style-type: none"> ----- ----- 	<ul style="list-style-type: none"> Programación I Laboratorios de Computación I

i) Bibliografía

Nro.	Autor/es	Título	Editorial	Año de edición	Principal	Complementaria
1	ECKEL BRUCE	PIENSA EN JAVA	PEARSON ALHAMBRA	2007	x	
2	Joseph Schmuller	APRENDIENDO UML EN 24 HRS	PRENTICE HALL	2006	x	
3	DEITEL HARVEY y DEITELPAUL	JAVA COMO PROGRAMAR	PEARSON EDUCACION	2010	x	
4	James Rumbaugh, Grady Booch, Ivar Jacobson	EL LENGUAJE UNIFICADO DE MODELADO: MANUAL DE REFERENCIA	PEARSON EDUCACION	2007		x

10- SITIO WEB OFICIAL

- **ORACLE** : <http://www.oracle.com/>
- **Netbeans**: <http://netbeans.org/>
- **JqueryMobile**: <http://jquerymobile.com/>
- **Mysql**: <http://www.mysql.com/>
- **Ampps**: <http://www.ampps.com/>
- **JAVA**: <https://www.java.com/>

Nombre del director	Nombre del encargado de la Cátedra
Ing. Osvaldo Giordaninni	Martín Julio Vargas Cristina Domizio
Firma del Director	Firma de los encargados de la Cátedra
Fecha de entrega del programa	28 de febrero de 2023

Planificación de Cátedra

a) Metodología de Enseñanza

La metodología de enseñanza de esta materia es teórico-práctica, el abordaje de los temas por parte de los profesores invita a los estudiantes a la realización continua de ejercicios prácticos

Los docentes guían y motivan a los estudiantes en la formación de equipos de trabajo, tratando de lograr la homogeneidad de los integrantes respecto de carga de materias, disponibilidad de tiempo, conocimientos y experiencia laboral, a fin de disponer en el curso de equipos con similares condiciones.

Cada estudiante debe tener la carpeta de trabajos prácticos completa, para su trabajo y a disposición de los docentes, en todas las clases. **Para tal fin se utilizará una modalidad de dictado presencial y un máximo de un 20% virtual (sincrónico o asincrónico). El docente del espacio curricular, avisará previamente con un tiempo prudente cual clase u actividad se realizará de manera virtual.**

Metodología	Explicación de la modalidad
Clases teóricas	<ul style="list-style-type: none">. Explicación interactiva del docente.. Discusiones entre equipos de trabajo.. Autoaprendizaje e interpretación de textos.
Trabajos Prácticos	<ul style="list-style-type: none">. Discusiones entre equipos de trabajo.. Investigación y presentación de tareas y proyectos codificados.
Prácticas en laboratorio	<ul style="list-style-type: none">. Análisis y resolución de casos en clase.. Presentación de experiencias y conocimientos.
Otras	

b) Cronograma de actividades

Semana N°	Unidad	Contenidos	Objetivos	Actividades	Evaluaciones	Modalidad de Cursado	Recursos	
							Bibliográficos	Didácticos
1	1	Introducción a la POO y aplicaciones del paradigma al lenguaje JAVA	Instalar el entorno de desarrollo.	Desarrollo teórico del paradigma.		Presencial	Apuntes de cátedra	PowerPoint
								Laboratorio
	1	Abstracción – Encapsulamiento – Ocultamiento	Comprender el paradigma y aplicar dichos conceptos en la codificación en la pc.	Desarrollo teórico y práctico de los conceptos. TP:1				Aula Virtual Videos tutoriales de la cátedra
	1	El modelo orientado a objetos y la Ingeniería directa						Internet
	1	Objetos y UML - Clases – Atributos – Constructores – Métodos - Mensajes						
2	-	Actividad de Repaso sobre contenidos adquiridos en la	Reafirmar y validar los conceptos de la	Desarrollo teórico y práctico del concepto de		Presencial	Apuntes de cátedra	PowerPoint
								Laboratorio
								Aula Virtual

	1	introducción a la programación.	programación funcional.	Test con Mocks.				Apuntes de cátedra Internet
	1	Herencia. – Clase Abstracta – Polimorfismo.	Comprender el paradigma y aplicar dichos conceptos en la codificación en la pc.	Desarrollo teórico y práctico de una aplicación integradora TP:2				
	1	Interfaz.		TP:3				
3	1	Asociaciones: Unidireccionales – Bidireccionales. Agregación – Composición	Implementar varios proyectos de codificación en netbeans	Llevar a la práctica las lecturas de diagramas de clases en notación UML a código. TP:1		Presencial	Apuntes de cátedra	PowerPoint Laboratorio Aula Virtual Apuntes de cátedra Internet
4	1	Asociaciones: Unidireccionales – Bidireccionales. Agregación – Composición	Implementar varios proyectos de codificación en netbeans	Llevar a la práctica las lecturas de diagramas de clases en notación		Presencial	Apuntes de cátedra	PowerPoint Laboratorio Aula Virtual

	1	Primer Parcial sobre POO, conceptos teóricos.	Reafirmar y validar los conceptos de la programación funcional	UML a código. TP:1 Desarrollo teórico y práctico del concepto de Test con Mocks.	Primer Parcial		Apuntes de cátedra	Apuntes de cátedra Internet
5	2	Arreglos primitivos – Arreglos de Objetos.	Manipular arreglos en java.	Desarrollo práctico en pc.		Presencial	Apuntes de cátedra	PowerPoint Laboratorio Aula Virtual Apuntes de cátedra Internet
	2	Estructuras Dinámicas el Framework de Colecciones:	Conocer y manipular el framework de desarrollo de los contenedores en JAVA.	Desarrollo teórico y práctico. TP:4				
6	2	Arreglos primitivos – Arreglos de Objetos.	Manipular arreglos en java.	Desarrollo práctico en pc.		Virtual	Apuntes de cátedra	PowerPoint Laboratorio Aula Virtual Apuntes de cátedra Internet
	2	Estructuras Dinámicas el Framework de Colecciones:	Conocer y manipular el framework de desarrollo de los	Desarrollo teórico y práctico. TP:4				

			contenedores en JAVA.					
7		Segundo parcial y entrega de la carpeta de trabajos prácticos. Elaboración de ejercicios			Segundo parcial	Presencial		Aula Virtual Laboratorio Apuntes de cátedra Internet
8	3	Interfaces de usuario controladas por eventos y manejo de eventos en Swing - Ejemplo sencillo: botón en un cuadro	Conocer los diferentes componentes de la interface gráfica SWING	Desarrollo teórico y práctico de un ejercicio. TP:5		Presencial	Apuntes de cátedra	PowerPoint Laboratorio Aula Virtual Apuntes de cátedra Internet
	3	Diseños: FlowLayout, BorderLayout, GridLayout y BoxLayout	Conocer los diferentes componentes de la interface gráfica SWING	Desarrollo teórico y práctico de un ejercicio. TP:5				
	3	Menús, acciones y barras de herramientas : JMenu y JMenuItem, JToolBar y Action	Conocer los diferentes componentes de la interface gráfica SWING	Desarrollo teórico y práctico de un ejercicio. TP:5				

	3	Contenedores: - JFrame, JDialog, JPanel y JScrollPane Patrón MVC.	Comprender y aplicar el patrón de diseño.	Desarrollo teórico y práctico de un ejercicio. TP:6				
--	---	---	--	---	--	--	--	--

9	4	Introducción a las bases de datos relacionales – MySQL	Conocer un BD relacional, sus relaciones y características principales.	Desarrollo teórico y práctico de los conceptos. TP:7		Presencial	Apuntes de cátedra	Power Point Apuntes de Cátedra Laboratorio Aula Virtual
	4	Definición de JDBC – Seleccionar una base de datos – Establecer la conexión – Seleccionar una tabla – Recuperar Valores desde una Hoja de Resultados	Conocer las distintas funciones JDBC y manipularlas en un proyecto en el framework de programación.	TP:8				
	5	Introducción al modelado en 4 capas. La capa de la vista, negocio, control y modelo. La capa del modelo y controlador	Implementar un concepto de separación de a código para la aplicación del patrón de diseño	Desarrollo práctico de un ejercicio integrador para gestionar la persistencia para base de datos. TP:9				
10	5	Desarrollo de una aplicación	Implementar un concepto	Desarrollo práctico de		Virtual	Apuntes de cátedra	PowerPoint

		integral modelando en 4 capas. La capa de la persistencia.	de separación de a código para la aplicación del patrón de diseño	un ejercicio integrador para gestionar la persistencia para base de datos. TP:9				Laboratorio Aula Virtual Apuntes de cátedra Internet
11	5	<p>Ventajas-cohesión – acoplamiento. – Independencia de las clases.</p> <p>Desarrollo de una aplicación integral modelando en 4 capas. La capa de la persistencia.</p>	<p>Comprender el mecanismo transversal de utilizar el patrón de diseño.</p> <p>Implementar un concepto de separación de a código para la aplicación del patrón de diseño</p>	<p>Desarrollo teórico.</p> <p>Desarrollo práctico de un trabajo final integrador para gestionar la persistencia para base de datos en un patrón MVC. TP:9</p>		Virtual	Apuntes de cátedra	<p>PowerPoint</p> <p>Laboratorio</p> <p>Aula Virtual Apuntes de cátedra</p> <p>Internet</p>
12	5	Desarrollo de una aplicación integral modelando en 4	Implementar un concepto de separación de a código para la	Desarrollo práctico de un ejercicio integrador para		Presencial	Apuntes de cátedra	<p>PowerPoint</p> <p>Laboratorio</p> <p>Aula Virtual</p>

		capas. La capa de la persistencia.	aplicación del patrón de diseño	gestionar la persistencia para base de datos. TP:9				Apuntes de cátedra Internet
13		Tercer parcial que requiere la entrega del desarrollo práctico de la Elaboración de Proyecto integrador para gestionar la persistencia para base de datos en un patrón MVC. Con la carpeta de trabajos prácticos completa.		Desarrollo práctico de un trabajo final integrador para gestionar la persistencia para base de datos en un patrón MVC.	Tercer parcial	Virtual		Laboratorio Aula Virtual Apuntes de cátedra Internet
14		Recuperatorios Entrega de carpet trabajos prácticos.			Recuperatorios	Presencial		Laboratorio Aula Virtual Apuntes de cátedra Internet
15		Parcial Global Integrador y entrega de la carpeta de trabajos			Parcial Global	Presencial		Laboratorio Aula Virtual

		prácticos. Elaboración de Proyecto.						Apuntes de cátedra Internet
16		Recuperatorios de Global			Recuperatorio Global	Presencial		Laboratorio Aula Virtual Apuntes de cátedra Internet
Primer Llamado a Exámenes de Julio								

c) Trabajos de campo, visitas a empresas

Actividad	Objetivo	Lugar	Responsable	Evaluación
-	-	-	-	-

d) Articulación horizontal y vertical con otras materias

En cuanto a la articulación horizontal:

- La cátedra de Programación II debe apoyar a la cátedra de Laboratorio II.

En cuanto a la articulación vertical:

- La cátedra de Programación II funda las bases en cuanto a los primeros pasos de diseño, modelado de objetos y el manejo del Framework para las cátedras de Laboratorio III, Programación III y Diseño y administración de Base de Datos del siguiente semestre.

Temas de la Cátedra	Temas (materia relacionada)	Materia relacionada							
		Laboratorio II.	Laboratorio III	Diseño y Administración de Base de Datos	Programación III	Proyecto Final			
Paradigma de Orientación a Objetos.		x	x		x	x			
Estructuras de memoria estáticas, dinámicas y manejo de excepciones.		x	x		x	x			
Nociones fundamentales sobre interfaces Swing. Manejo de patrón de diseño.		x	x		x				
Acceso a datos a través a JDBC.			x	x	x	x			
Introducción al modelado en 4 capas.			x		x	x			

e) Régimen de cursado y aprobación

1) Aspectos considerados en la evaluación.

La evaluación de la cátedra se establecerá sobre 4 etapas claramente definidas.

La **primera etapa** abarca un examen parcial, enfocado en reafirmar los conocimientos básicos de la programación orientada a objetos, adquiridos durante el cursado de las primeras clases. Para cumplir este requerimiento debe obtener una nota de 4 (60 %) o superior.

La evaluación de la **segunda etapa** abarca la presentación de la carpeta de trabajos prácticos y la aprobación del segundo examen parcial. Nuevamente el alumno debe cumplir este requerimiento con una nota de 4 (60 %) o superior, **ver escala de referencia más abajo**.

La evaluación de la **tercera etapa** abarca la presentación de la carpeta de trabajos prácticos y la aprobación del tercer examen parcial, con un proyecto final integrador definido en las actividades. Para ello, el alumno debe cumplir este requerimiento con una nota de 4 (60 %) o superior.

Para acceder a la **cuarta etapa**, el alumno tiene que haber obtenido en **alguna de las instancias evaluativas anteriores** una calificación superior a 4 (60 %) e inferior a 6 (70 %), sin haber recuperado, alguno de estos exámenes previos. Esta etapa consta de un **examen global** integrador, que de aprobarse con una nota de 6 o superior, dará al alumno la materia aprobada en carácter de **Aprobación Directa (AD)**. De lo contrario el alumno obtendrá la **Aprobación No Directa**.

a) Instancias de Recuperación

Los alumnos tendrán en todas las instancias la posibilidad de recuperar una vez, para poder acreditar con la **Aprobación No Directa**.

b) Acreditación del alumno

Los alumnos obtendrán la **Aprobación Directa (AD)** aprobando con 6 o más (70% o más) 3 parciales, sin haber recuperado ningún examen y habiendo cumplido con el 75 % de asistencia a las clases y entregado todos los trabajos prácticos. También obtendrán la **Aprobación Directa (AD)** los estudiantes que obtuvieron en **alguna de las instancias evaluativas anteriores** una calificación superior a 4 (60 %) e inferior a 6 (70 %), sin haber recuperado, alguno de estos exámenes previos y al rendir un **examen global** integrador, que de aprobarse con una nota de 6 (70 %) o superior, dará al alumno la materia aprobada en carácter de **Aprobación Directa (AD)**.

Cálculo:

Nota Final = $((1^{\circ} \text{ Examen} \geq 6 + 2^{\circ} \text{ Examen} \geq 6 + 3^{\circ} \text{ Examen} \geq 6) / 3)$.

Nota Final = $((1^{\circ} \text{ Examen} \geq 4 + 2^{\circ} \text{ Examen} \geq 4 + 3^{\circ} \text{ Examen} \geq 4) / 3) + \text{Global} \geq 6) / 2$.

Los alumnos obtendrán **Aprobación No Directa (AND)** aprobando con 4 ó 5.9 (60% a 69%) 3 parciales o sus recuperatorios (1 por cada parcial), habiendo cumplido con el 75 % de asistencia a las clases y habiendo

entregado en tiempo y forma los trabajos prácticos. Además, obtendrán esta acreditación todos aquellos que aprobaron las evaluaciones anteriores con 4 (60 %) y pudieron acceder al **examen global integrador**, pero obtuvieron una nota inferior a 6 (70%) en dicho parcial.

Cálculo:

Nota Final = $(1^{\circ} \text{ Examen} \geq 4 + 2^{\circ} \text{ Examen} \geq 4 + 3^{\circ} \text{ Examen} \geq 4) / 3$.

Nota Final = $((1^{\circ} \text{ Examen} \geq 4 + 2^{\circ} \text{ Examen} \geq 4 + 3^{\circ} \text{ Examen} \geq 4) / 3) + \text{Global} < 6) / 2$.

Los alumnos obtendrán **No Aprobación (NA)** obteniendo calificaciones de 1, 2 y/o 3 (0 a 59%), en parciales y/o sus recuperatorios, no habiendo cumplido con el 75% de asistencia a las clases y no habiendo entregado en tiempo y forma los trabajos prácticos.

Cálculo:

Nota Final = $(\text{Examen}_1 < 4 \text{ y/o } \text{Examen}_2 < 4 \text{ y/o } \text{Examen}_3 < 4) / 3$.

También deben considerar que aprobado es con 6 (70%) ó más para los exámenes finales.

Aprobación Directa: SI

Notas, valores o % mínimos esperados	Para aprobar en forma directa		Para aprobar en forma no directa	
	Individual	Grupal	Individual	Grupal
Parciales	6		4	
Recuperatorios de parciales	6		4	
Global Integral	6		$\geq 4 \text{ y } < 6$	
Recuperatorios Global Integrador	6			
Otros:				
Asistencia (mínimo 75%).	75%		75%	

Observar escala de referencia:

Puntaje en %	Nota del 1 al 10	Observación
0.0	1	Insuficiente (NA)
9.0	1.4	
10.0	1.5	
19.0	1.9	
20.0	2	
29.0	2.4	
30.0	2.5	
39.0	2.9	
40.0	3	
49.0	3.4	
50.0	3.5	
59.0	3.9	
60.0	4	Aprobado (AND)
69.0	5.9	
70.0	6	Bueno (AD)
79.0	6.9	
80.0	7	
89.0	8.4	Distinguido (AD)
90.0	8.5	
99.0	9.9	
100.0	10	Sobresaliente (AD)

5) Modalidad de examen final

- Evaluación teórica práctica de:
 - . Trabajos prácticos Individuales Nro. 1 al 9
 - . Temas teóricos tratados durante el cursado

f) Actividades del equipo docente

Docente	Categoría	Dedicación	Actividades			
			Docencia	Gestión	Investigación	Extensión
Martín Julio Vargas	Adjunto	1	Responsable de la cátedra (Turno Tarde)			
Cristina Domizio	Adjunto	1	Responsable de la cátedra (Turno Mañana)			

g) Observaciones

Escriba las observaciones que considere pertinentes .

h) Horario de Consulta de Profesores y Auxiliares (a completar antes del 31 de marzo si es una materia del primer semestre o anual, a completar antes del 31 de agosto si es una materia del segundo semestre)

Profesor Martín Vargas

En tiempos de Pandemia la comunicación para evacuar dudas es vía Discord, Telegram y Telefónica, durante todo el año.

- **Primer Semestre**

Nombre del docente:

Lunes: 11:00 a 12:00 hs.

- **Segundo Semestre**

Nombre del docente:

Lunes: 11:00 a 12:00

Profesora Cristina Domizio

- **Primer Semestre**

Nombre del docente:

Martes: 16:45 a 17:45 hs.

- **Segundo Semestre**

Nombre del docente:

Martes: 16:45 a 17:45 hs.

Nombre del director	Nombre del encargado de la Cátedra
Ing. Osvaldo Giodaninni	Lic. Martín Julio Vargas Lic. Cristina Domizio
Firma del Director	Firma de los encargados de la Cátedra
Fecha de entrega de la planificación	28 de febrero de 2023