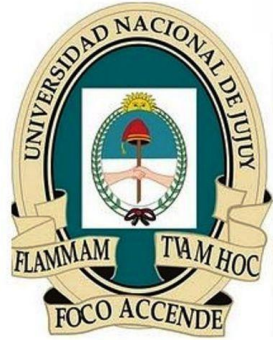


FACULTAD DE INGENIERÍA
UNIVERSIDAD NACIONAL DE JUJUY



PROPUESTA PEDAGÓGICA
PLANIFICACIÓN DE CÁTEDRA

PERÍODO LECTIVO	2023	
CARRERA Y PLAN DE ESTUDIO	Ingeniería Industrial	2022
BLOQUE DE CONOCIMIENTOS	Electivas	
NOMBRE DE LA ASIGNATURA	Programación Avanzada (Electiva)	
RESPONSABLE DE CÁTEDRA	Esp. Ing. Laura Adriana Garay	
EQUIPO DE CÁTEDRA	Esp. Ing. Delia Cristina Cruz	
MODALIDAD DE CURSADO	Cuatrimestral	
CARÁCTER	Electiva	
CARGA HORARIA SEMANAL	6	
CARGA HORARIA TOTAL	90	
AÑO EN EL PLAN DE ESTUDIO	5to	

• DESCRIPCIÓN DE LA ASIGNATURA

La asignatura de Programación Avanzada en el currículo de Ingeniería Industrial desempeña un papel fundamental al proporcionar a los estudiantes las habilidades y conocimientos necesarios para utilizar la programación como una herramienta poderosa en la resolución de problemas relacionados con la optimización, el análisis de datos y la automatización de procesos en el ámbito industrial.

1.1. Relación de la asignatura con el Perfil de Egreso

En un mundo cada vez más orientado a la tecnología, los ingenieros industriales deben estar equipados con habilidades que les permitan modelar, simular y controlar sistemas complejos de manera eficiente. La programación brinda a los estudiantes la capacidad de traducir problemas del mundo real en algoritmos y programas de software que pueden analizar datos, tomar decisiones basadas en datos, y mejorar la eficiencia y productividad en entornos industriales.

A través de esta asignatura, los estudiantes se familiarizarán con los conceptos fundamentales de la programación, incluyendo la lógica de programación, estructuras de control, funciones y estructuras de datos. Además, aprenderán a aplicar estos conceptos en la solución de problemas específicos de la ingeniería industrial, como la gestión de inventarios, la planificación de la producción, el diseño de sistemas de calidad y la optimización de procesos logísticos.

1.2. Relación de la Asignatura con los Alcances del Título.

Al finalizar la asignatura, los estudiantes habrán adquirido la capacidad de analizar problemas complejos, descomponiéndolos en componentes manejables, y desarrollar soluciones efectivas a través de la escritura de programas. Estarán preparados para enfrentar desafíos industriales del mundo real utilizando la programación como una herramienta para modelar y mejorar sistemas, contribuyendo así al avance y la eficiencia de las operaciones en una variedad de entornos industriales.

1.3. Competencias

Genéricas

CG1. Capacidad para relacionar conceptos.

CG2. Capacidad para manejar y articular de manera eficaz distintos lenguajes comunicación (formal, simbólico y gráfico).

CG3. Capacidad para desempeñarse adecuadamente en equipos de trabajo

Específicas

CE1. Capacidad para realizar la búsqueda creativa de soluciones algorítmicas y seleccionar la alternativa más adecuada.

CE2. Capacidad para el diseño y ejecución de programas conforme a los principios o fundamentos específicos de la Programación Imperativa y la Programación Orientada a Objetos.

CE3. Capacidad para emplear, según la situación problemática los conceptos fundamentales de Programación.

1.4. Resultados de Aprendizaje (RA)

RA1 Diseñar, implementar y depurar programas en C++, utilizando conceptos fundamentales como variables, tipos de datos, estructuras de control, funciones y clases, para resolver problemas de programación de nivel intermedio, demostrando un entendimiento sólido de los principios de la programación orientada a objetos y la capacidad de aplicarlos en la creación de software funcional y eficiente.

RA2 Comprender los conceptos básicos de las bases de datos, incluyendo la estructura de las bases de datos relacionales, la creación y modificación de tablas, la realización de consultas simples utilizando SQL, y la comprensión de los principios fundamentales de la normalización de bases de datos.

RA3 Diseñar y crear bases de datos simples, ingresar y recuperar datos utilizando consultas sencillas, y comprender la importancia de mantener la integridad de los datos a través de buenas prácticas de diseño de bases de datos.

1.5 Resultados de Aprendizaje / Competencias / Unidades Analíticas

RA	COMPETENCIAS		UNIDADES DEL PROGRAMA ANALÍTICO
	Genéricas	Específicas	
RA01	1,2,3	1,2,3	1, 2, 3, 6
RA02	1,2,3	1	4, 5
RA03	1,2,3	1	4, 5

• CONTENIDOS

2.1. Contenidos Mínimos

Lenguaje C++ avanzado, sentencias, funciones y estructuras. Vectores. Entrada y Salida. Técnicas de organización de bases de datos. Álgebra relacional. Seguridad y controles. Herramientas CASE para diseño. Lenguajes de consulta. Documentación específica. Programación orientada a objetos.

2.2. Programa Analítico

UNIDAD 1: Introducción C++: Estructura de un programa en C++. Elementos de programas C++. Tipos de datos de C++. Expresiones. Entrada y Salida. Estructuras de control

UNIDAD 2: Funciones en C++. Funciones de devolución de valor. Funciones de biblioteca. Funciones void (vacías).

UNIDAD 3: Tipos estructurados en C++. Arreglos y Registros. Declaración. Acceso a elementos. Operaciones básicas. Combinación arreglos y registros.

UNIDAD 4: Base de Datos. Introducción. El modelo Entidad-Relación. Modelo Relacional. Diseño de Base de Datos. Introducción a SQL. Lenguaje de Definición de Datos (DDL). Lenguaje de Manipulación de Datos (DML). Formulación de Consultas con SQL

UNIDAD 5: Programación Orientada a Objetos. Clases y Objetos. Instanciación de objetos. El objeto this. El constructor. Extensión y herencia. Visibilidad y encapsulación.

• ACTIVIDADES PRÁCTICAS

1. **Indicar la carga horaria semanal** dedicada al dictado de la actividad curricular y, en particular, a las actividades de formación práctica. Si la actividad curricular no incluye actividades de formación práctica todas las semanas, estimar el promedio. Las cargas horarias se deben indicar en horas reloj (no usar horas cátedra).

PLAN DE ESTUDIOS	CARGA HORARIA SEMANAL	
2022	Carga horaria semanal dedicada a la formación teórica	3
	Carga horaria semanal dedicada a la formación práctica.	3
	Carga Horaria semanal total	6

UNIDAD TEMÁTICA	ACTIVIDAD	CONOCIMIENTOS DE:
1, 2 y 3	<i>Resolución de Problemas mediante entorno de programación de C++</i>	Introducción C++
		Funciones en C++
		Datos estructurados en C++
	<i>Entrega grupal de ejercicios asignados por la cátedra</i>	Equipos de trabajo
Entorno de Programación C++		
Expresión oral y escrita		
4	<i>Resolución de Problemas en SQL</i>	Introducción a Base de Datos
		BD: Diseño y Desarrollo de Aplicaciones
	<i>Entrega grupal de los ejercicios asignados por la cátedra</i>	Equipos de trabajo
		Entorno de Programación C++

UNIDAD TEMÁTICA	ACTIVIDAD	CONOCIMIENTOS DE:
		Expresión oral y escrita
5	<i>Resolución de Problemas en POO mediante entorno de programación Java</i>	Programación Orientada a Objetos
1 a 5	<i>Autoevaluación y evaluación entre pares provista por la cátedra</i>	Rúbricas provistas por la cátedra
		Uso de aula virtual

2. Especifique los ámbitos donde se desarrollan las actividades de formación práctica a las que se hace referencia en el punto anterior (Ejemplo: Laboratorio de Química Analítica, LUTI)

Las actividades de formación práctica se llevarán a cabo bajo modalidad híbrida con clases virtuales mediante encuentro sincrónicos a través de la plataforma ZOOM y disposición del material en aula virtual en plataforma Moodle. En cuanto a la modalidad presencial, las mismas se realizarán oportunamente en sala LUTI.

• METODOLOGÍA DE ENSEÑANZA

UT	TP	TEMA	MODALIDAD	RECURSOS DIDÁCTICOS
U1	1	Programación Básica en C++	Híbrida (Presencial/Virtual)	Clases de enseñanza interactiva y participativas con aprendizaje activo y colaborativo. Planteo de situaciones problemáticas para el aprendizaje Cooperativo en Grupos Pequeños Presentaciones Orales Presentaciones escritas
U2	2	Programación Modular en C++	Híbrida (Presencial/Virtual)	
U3	3	Datos estructurados en C++	Híbrida (Presencial/Virtual)	
U4	4	Introducción a BD	Híbrida (Presencial/Virtual)	
U5	5	POO	Híbrida (Presencial/Virtual)	

UT: Unidad Teórica. **TP:** Trabajo Práctico

• EVALUACIÓN Y ACREDITACIÓN

- **Evaluación Teórica:** Se realizarán 2 evaluaciones teóricas a mitad y al final de la cursada. Las evaluaciones se tomarán de manera remota o presencial.
- **Trabajos de Programación:** Se deben realizar las presentaciones de los problemas planteados en tiempo y forma.
- **Trabajo integral:** Se debe realizar la presentación del trabajo integral en tiempo y forma

1) Requisitos para Promocionar la Asignatura

- El alumno promocionará, si
 - Aprueba las evaluaciones teóricas,
 - Aprueba por lo menos el 80 % de las presentaciones de los trabajos de programación
 - Aprueba el trabajo integral con una nota igual o superior a 8
 - Cumple un 80% de asistencia

2) Requisitos para Regularizar la Asignatura

- El alumno regularizará, si
 - Aprueba las evaluaciones teóricas o sus recuperatorios,
 - Aprueba por lo menos el 50 % de las presentaciones de los trabajos de programación
 - Aprueba el trabajo integral con una nota igual o superior a 5.
 - Cumple un 50% de asistencia

3) Evaluación de Actividades Prácticas

ID	ACTIVIDAD	AE	CE	HE
A	<i>Evaluaciones Teóricas</i>	SI		SI
B	<i>Resolución de Problemas</i>			SI
C	<i>Entrega/Presentación grupal de los ejercicios asignados al grupo por la cátedra</i>		SI	SI
D	<i>Trabajo Integral</i>		SI	SI

AE. Autoevaluación. CE. Coevaluación. HE. Heteroevaluación.

4) Cronograma de Evaluaciones

Evaluación	Fecha Prevista
Eval. Teo 1	16/10/2023
Eval. Pract. 1	19/10/2023
Eval. Teo 2	20/11/2023
Eval. Pract. 2	23/11/2023
Trabajo Integral	30/11/2023

5) Requisitos para Examen Final Libre

El examen libre constará de las siguientes instancias

- **INSTANCIA PRÁCTICA:**

- (1) Desarrollo y defensa de un trabajo de programación. El enunciado deberá ser solicitado a la cátedra 1 semana antes de la fecha del examen y el desarrollo deberá ser entregado hasta 24 horas antes de la fecha y horario del examen para su revisión. Aprobado el desarrollo, el alumno realizará la defensa del mismo en la fecha establecida para el examen.
- (2) Resolución de una evaluación práctica el día del examen con problemas relacionados a las 5 unidades de la materia

- **INSTANCIA TEÓRICA:**

- (1) Desarrollo de una evaluación oral (teoría) acerca de los temas de las unidades temáticas de la asignatura.

Para acceder a cada etapa del examen, se debe aprobar la anterior, con el **80%** del desarrollo **COMPLETO y CORRECTO**.

- **BIBLIOGRAFÍA BÁSICA OBLIGATORIA**

Título	Autores	Editorial	Año de edición	Ejemplares disponibles
Fundamentos de programación libro de problemas.	Joyanes Aguilar, Luis et al	Madrid McGrawHill	2003	1
Programación en C++ Algoritmos, estructuras de datos y objetos.	Joyanes Aguilar, Luis	Colomba McGrawHill	2000	1
Algoritmos, datos y programas. Conceptos básicos.	De Giusti, Armando et al	La Plata Editorial Exacta	1998	1
Introducción a la Programación y Estructura de Datos	Braunstein S y Gioia A.	Eudeba	1991 1996	2

ANEXO PROGRAMA ANALÍTICO DE LA ASIGNATURA

PROGRAMACIÓN AVANZADA

UNIDAD 1: Introducción C++: Estructura de un programa en C++. Elementos de programas C++. Tipos de datos de C++. Expresiones. Entrada y Salida. Estructuras de control

UNIDAD 2: Funciones en C++. Funciones de devolución de valor. Funciones de biblioteca. Funciones void (vacías).

UNIDAD 3: Tipos estructurados en C++. Arreglos y Registros. Declaración. Acceso a elementos. Operaciones básicas. Combinación arreglos y registros.

UNIDAD 4: Base de Datos. Introducción. El modelo Entidad-Relación. Modelo Relacional. Diseño de Base de Datos. Introducción a SQL. Lenguaje de Definición de Datos (DDL). Lenguaje de Manipulación de Datos (DML). Formulación de Consultas con SQL

UNIDAD 5: Programación Orientada a Objetos. Clases y Objetos. Instanciación de objetos. El objeto this. El constructor. Extensión y herencia. Visibilidad y encapsulación.