Programa de Curso

Paradigmas de Programación

Ingeniería de Sistemas de Información con grado en Bachillerato y salida lateral de Diplomado en Programación de Aplicaciones Informáticas

NOMBRE DEL CURSO:	Paradigmas de Programación
TIPO DE CURSO:	Regular
CÓDIGO DE CURSO:	EIF400
NIVEL Y GRADO ACADÉMICO:	2do ciclo del 3er nivel Bachillerato
PERIODO LECTIVO:	II Ciclo 2025
MODALIDAD:	Presencial
NATURALEZA:	Teórico / Práctico.
CRÉDITOS:	4
HORAS TOTALES SEMANALES:	11
HORAS DEL CURSO:	4 (3 teoría, 1 práctica)
HORAS DOCENTE:	4
HORARIO DE ATENCIÓN ESTUDIANTE:	Dr. Carlos Loría-Sáenz: L, J 3pm-5pm (**con cita previa** 24 horas de antelación)
REQUISITOS:	EIF206 Programación 3
CORREQUISITOS:	**NA**
PERSONA DOCENTE:	Carlos Loría-Sáenz grupos 04 (51389), 06 (51386) y 05 (51387).

"En esta Universidad nos comprometemos prevenir, investigar y sancionar el hostigamiento sexual entendido como toda conducta o comportamiento físico, verbal, no verbal escrito, expreso o implícito, de connotación sexual no deseado o molesto para quien o quienes lo reciben, reiterado o aislado. Si usted está siendo víctima de hostigamiento diríjase a la Fiscalía de Hostigamiento Sexual de la UNA o llame al teléfono 2277-3961".

I. Descripción

Al elaborar un modelo para resolver un problema mediante programación, existen diferentes enfoques sobre cómo se debe realizar la abstracción de los diferentes elementos de dicho problema. Así, de acuerdo con la situación que se desea modelar, cada uno de estos distintos enfoques o paradigmas de programación tiene ventajas y desventajas, que facilitan o entorpecen la construcción de un programa. En el curso se estudian de manera comparativa diferentes paradigmas de programación existentes, logrando de esta manera conocer los criterios más importantes para la selección de un

lenguaje determinado. El curso busca también complementar el conocimiento de los estudiantes de ingeniería informática en los paradigmas y técnicas de modelado que no se han estudiado en los cursos regulares de programación

II. Objetivos

OBJETIVO GENERAL

Ofrecer a los estudiantes un panorama general sobre los principales paradigmas de programación, los conceptos teóricos que los fundamentan y las técnicas fundamentales utilizadas en cada uno.

OBJETIVOS ESPECÍFICOS

Al terminar el curso, se pretende que el estudiante haya logrado adquirir el conocimiento y las habilidades necesarias para:

1. Esbozar la historia de los lenguajes de programación y comprender la manera en que estos han evolucionado para adaptarse a las necesidades de desarrollo de software.
2. Reconocer e identificar los principales paradigmas de programación existentes (según el modelo teórico que los sustenta) y sus características principales.
3. Identificar las características de cada lenguaje de programación, su implementación y ambiente de ejecución.
4. Identificar y explicar conceptos de teoría de la computación para comprender los problemas y limitaciones que existen en la traducción de lenguajes y las principales técnicas utilizadas en compiladores e intérpretes.
5. Identificar las particularidades de un lenguaje de programación que pueden afectar de una u otra manera la construcción de un programa.
6. Conocer los principios teóricos que sirven de fundamento para los modelos de programación declarativa (funcional y lógica).
7. Emplear técnicas funcionales en la construcción de aplicaciones generales.
8. Modelar y resolver problemas computacionales utilizando los diferentes paradigmas estudiados.

III. Contenidos y Aprendizajes Integrales

Los contenidos del curso están organizados en tres grandes partes. Dada la inherente interrelación entre los contenidos, el orden en que los temas del curso serán finalmente cubiertos, no corresponde, necesariamente, con el detalle secuencial siguiente, sino que estos podrán ser estudiados y evaluados de la manera en que cada profesor considere conveniente. En particular, dichos temas pueden ser estudiados de manera gradual, como conjunto y no obligatoriamente de manera secuencial. Dependiendo del desarrollo del curso, es posible que se reorganicen algunas lecciones del cronograma para poder cubrir adecuadamente cada uno de los temas propuestos. La parte 1 del curso describe el marco general de referencia, las partes 2 y 3 abordan ese marco en casos de estudio concretos de paradigmas y lenguajes representativos. Los temas de la parte 1 podrán ser retomados, ampliados o contrastados a la luz de los casos de estudio de lenguajes. Eso implica que ciertos temas de la primera parte se extienden transversalmente durante todo el curso y pueden ser reevaluados más de una vez.

Parte 1 – Introducción y características generales de los lenguajes de programación, traducción y paradigmas

1. Introducción a los lenguajes de programación (Objetivos 1 ,2, 4)
    a. Contexto histórico de los lenguajes de programación.
    b. Problema de la Abstracción computacional: datos y control
    c. Programación imperativa versus declarativa (expresiones versus estatutos)
    d. Contraste entre Iteración versus recursión
    e. Lenguajes dinámicos versus estáticos
    f. Portabilidad, Modularidad, Reutilización
    g. Manejo memoria
    h. Lenguajes de propósito específico versus uso general

i. Lenguajes y frameworks

j. IA generativa y LLMs

2. Introducción a traducción, compilación y evaluación (Objetivo 3, 4)

   a. Diseño de lenguajes de programación.

   b. Paso parámetros, scoping y resolución del binding

      i. Reconocimiento de patrones (pattern matching)

      ii. Clausuras

      iii. Polimorfismo

   c. Análisis sintáctico (Lexing, Parsing, tabla de símbolos, árboles de sintaxis abstracta (AST))

   d. Gramáticas libres de contexto, Autómatas de estado Finito (FSA) y expresiones regulares.

   e. Análisis semántico estático (disciplinas de tipificación)

   f. Evaluación y estrategias (aplicativa, diferida)

   g. Nociones de generación de código (máquinas virtuales)

3. Lenguajes de programación y arquitectura (Objetivos 2, 4, 5, 6)

   a. Tipos de datos y mecanismos de abstracción.

   b. Métodos de encapsulamiento y modularidad

   c. Herencia y polimorfismo, sobrecarga

   d. Estructuras de control secuenciales y asincrónicas (manejo por eventos)

   e. Manejo de memoria

4. Introducción a los paradigmas de programación.

   a. Modelos conceptuales y computacionales teóricos

b. Programación imperativa
    i. Modelos computacionales basados en estados.
    ii. Las estructuras básicas y la programación estructurada por control.
    iii. Programación estructurada
    iv. Programación asincrónica y dirigida por eventos
c. Programación declarativa
    i. Recursión
    ii. Programación funcional
    iii. Programación lógica
    iv. Enfoques híbridos o multiparadigma
d. Otros paradigmas de programación

Parte 2 – Programación funcional (Objetivos 2, 3, 5, 6)

Para esta parte del curso, se utilizarán diferentes lenguajes de programación funcional para estudiar los conceptos teóricos y algunas técnicas prácticas de programación.

1. Principios de programación funcional (Objetivos 2, 3)
    a. Justificación del modelo funcional.
    b. Características y principios fundamentales
2. Introducción al cálculo lambda (Objetivos 2, 5, 7)
    a. Historia y descripción.
    b. Sintaxis: Variable, abstracción Lambda y aplicación
    c. Reducción (evaluación por reemplazo y reescritura: normalización, estrategias, confluencia)

i. Reducción-α

ii. Reducción-β

iii. Reducción-η

d. Representación de la Aritmética y Lógica.

e. Combinadores de recursión (puntos fijos)

f. Computabilidad e indecidibilidad

3. Aplicaciones y limitaciones de la programación funcional (Objetivos 3, 5, 7)

a. Inmutabilidad de datos

b. Tipificación dinámica versus estática

c. Lambdas y Clausuras

d. Combinación con orientación a Objetos

e. Programación recursiva y su eliminación

f. Programación con combinadores y colecciones

g. Reconocimiento de patrones (pattern-matching)

h. Notación Curry y aplicación parcial

i. Programación reactiva y asincrónica

j. Combinación con programación orientada a objetos

Parte 3 – Programación lógica (Objetivos 2, 3, 5, 6)

Para esta parte del curso, se utilizará (preferiblemente) Prolog como lenguaje de programación declarativo arquetípico para estudiar la realización del paradigma lógico

1. Programación lógica (Objetivos 2,3,4,7)

a. Programación lógica vs. programación en Prolog.

b. Principios teóricos de la programación lógica.

c. Lógica de primer orden y cláusulas de Horn

d. Computación por razonamiento lógico-simbólico

i. El algoritmo de unificación

ii. El método de resolución.

iii. Árboles de refutación (prueba)

iv. Negación por falla y cut (faceta imperativa)

v. Metapredicados

e. Sobre Recursión versus Backtracking.

2. Aplicaciones y limitaciones de programación lógica

a. Modelamiento por reglas

b. Búsqueda combinatoria

c. Parsing

IV. Metodología

El curso contará con la exposición magistral por parte del profesor de cada uno de los contenidos descritos. Además, se harán sesiones de resolución de problemas y prácticas de laboratorio para enfrentar al estudiante de manera directa con las principales dificultades y técnicas utilizadas en programación. De esta manera también se logra que el estudiante aplique los conceptos expuestos en situaciones específicas.

Para su implementación en este curso, se dispondrán, como recursos de apoyo, videos cortos previos (asincrónicos) a las clases y grabaciones de las clases en vivo (sincrónicas). Habrá material digital en formato PDF disponibles para todos los distintos temas que incluyen prácticas de preparación para los exámenes.

Se podrá usar la plataforma MS-Teams para la realización de tutorías complementarias o reposiciones de clases si las hubiera. Esto incluye las sesiones de consulta. Dicha herramienta podría ser cambiada por cada docente según sus posibilidades técnicas o intereses en coherencia con los objetivos del curso.

De manera complementaria a las clases magistrales habrá proyectos de programación que requieren más tiempo del que se dispone durante las lecciones, donde los estudiantes resuelven en grupo ejercicios de dificultad media o alta, para conocer, estudiar y resolver problemas representativos.

También se realizarán investigaciones y se podrán solicitar exposiciones o demostraciones sobre diferentes temas, cuyo estudio no puede hacerse dentro de las limitaciones de tiempo de la clase, pero se consideran importantes para conseguir cumplir cabalmente el objetivo general del curso.

Por la naturaleza y variedad de los temas de este curso, existen variados criterios profesionales y vivencias personales sobre cómo hacer llegar los contenidos al estudiante. Esto incluye, en particular, el que hay distintas herramientas y estrategias que pueden ser usadas para alcanzar los mismos objetivos teóricos y conceptuales del curso. Siendo lo más importante los objetivos de la carta por encima de las herramientas, esta carta es flexible sobre la selección de las herramientas y la organización lógica del temario. Cada docente puede buscar, dada su experiencia profesional y docente, organizar los temas y seleccionar las herramientas que le permitan su mejor desempeño y el de sus estudiantes, siempre que se respeten los objetivos generales y específicos, así como los criterios y rubros de evaluación, los que serán los mismos entre distintos docentes y grupos.

Las herramientas de software requeridas durante las actividades de aprendizaje serán comunicadas al estudiante con la debida anticipación para que su oportuna instalación. Las mismas estarán disponibles en el laboratorio del curso en aquellas clases que lo requieran. Se espera que el estudiante disponga de las mismas en sus equipos personales para su trabajo más allá de la clase.

V. Evaluación

Reglas generales

1. Los exámenes en el curso buscan medir y evaluar la compresión de cada estudiante del material estudiado durante el curso y del trabajo realizado en los proyectos. Los exámenes deben realizarse y entregarse **individualmente**. Cuando algún trabajo, práctica o proyecto se realice en grupos, los criterios para la conformación de dichos grupos de trabajo en cada proyecto serán a discreción de cada uno de los docentes.
2. La fechas, horarios y formas de evaluación y calificación y manejo de ausencias serán siempre conforme a la normativa general de la universidad. Toda ausencia o llegada tardía requiere **una justificación formal apropiada** en especial en el caso de evaluaciones o revisiones. Sólo se evalúa la materia vista hasta 8 días antes de la prueba.
3. Los alcances específicos de los exámenes, trabajos y proyectos serán los que cada docente determine y comunique según su enfoque y su cobertura particular de los contenidos del curso. Dicho enfoque incluye la forma concreta de aplicación de los exámenes y la revisión de proyectos e investigaciones que cada docente determine.
4. Los exámenes cortos se realizarán según el cronograma, durante todo el curso. **El docente puede realizar evaluaciones cortas sin necesidad de avisar previamente su aplicación.**
5. Es obligatoria la entrega de **todos** los proyectos del curso para poder aprobar el curso.
6. La asistencia al curso es **puntual** y **obligatoria**. Cada estudiante deberá asistir al menos al 80% de las sesiones durante el ciclo lectivo. En caso contrario, reprobará el curso. Se está ausente si se es llamado al pasar lista al inicio de la clase o no se encuentra al ser llamado durante la lección. Se consideran tardías el no estar al pasar lista. Tres tardías cuenta por una ausencia.
7. Habrá trabajos grupales, los grupos se conformarán a libertad por los estudiantes cumpliendo completamente las normas y fechas que les serán comunicadas la primera semana de clase en anexo a esta carta. Una vez formados no podrán cambiarse. Si un estudiante no se registra como miembro de algún grupo en el lapso asignado

perderá el derecho a presentar los trabajos grupales y correspondientes puntos. El grupo nombrará una persona coordinadora la que responde por el grupo ante el profesor. El coordinador es responsable de entregar los trabajos asignados. Todo cambio o situación especial en un grupo debe ser comunicado oportunamente al profesor para su debida atención. Se parte de que cada grupo se **autoorganiza** y trabaja de manera **equitativa** y **justa**. El coordinador deberá informar al profesor oportunamente de situaciones que requieran su atención sobre la participación individual o grupal. En un trabajo grupal, cada miembro puede ser llamado a defender el trabajo, lo cual podrá ser tomado en cuenta en la nota grupal del trabajo en cuestión.

8. Los trabajos grupales con valor evaluativo deben ser realizados **exclusivamente por cada grupo** sin compartir información propia de ninguna especie con los otros grupos sean estos del mismo horario o no. Coincidencias injustificables en detalles puntuales de trabajos podrán ser razón para sanciones según normativas de plagio y copia.
9. Por la naturaleza del curso, es inevitable que **los contenidos desarrollados sean acumulativos** para los exámenes y trabajos prácticos. Es decir, aunque en una actividad que conlleva una evaluación (sea individual o grupal) se deba profundizar en uno o más temas específicos, esto no implica que no se puedan incluir temas anteriormente evaluados.
10. Al ser un curso cuya evaluación contempla aspectos prácticos, como laboratorios y proyectos programados, **no es posible un examen extraordinario**.
11. La suma de los porcentajes obtenidos por el estudiante en los rubros de evaluación indicados determina su nota de aprovechamiento (NA). **El curso se aprueba con una NA igual o superior al 70%.**

Descripción de los rubros de evaluación	Porcentaje
Primer examen parcial	**20%**
Segundo examen parcial	**20%**
Quices, Tareas, Prácticas, Apreciación Se realizarán al menos 5 (cinco) exámenes cortos (*quices*) para evaluar el progreso del grupo. También se asignarán tareas cortas y prácticas que pueden ser completadas en clase o asignadas para ser trabajadas fuera de las horas de clase.	**10%**

Trabajos de investigación Se hará trabajo de investigación sobre alguno o varios temas relacionados con los objetivos del curso. El profesor dará la guía de contenidos a cubrir en dicha tarea y sus alcances generales, valorando el nivel de complejidad, el aporte a los temas y objetivos del curso de forma que el desarrollo de la tarea sea el adecuado. Se recomienda que esta tarea demande el uso de las bases de datos de la universidad. Un aspecto deseable del curso es que los estudiantes continúen desarrollando su capacidad de trabajo en grupo y mejoren sus habilidades para comunicar el resultado de sus investigaciones y desarrollos al resto de sus compañeras y compañeros.	**10%**
Proyectos Los proyectos servirán para evaluar aspectos prácticos concretos de los temas estudiados en el curso. Los proyectos asignados serán preferiblemente programados, pero pueden tratar también de algún tipo de desarrollo teórico. Pueden ser varios proyectos independientes o un único proyecto integrado, según el criterio de cada profesor. **Serán grupales** en grupos que se formarán por los estudiantes y según las indicaciones del profesor.	**40%**

VI. Normas Específicas

1. Se pasará lista por asistencia cada clase en los primeros 5 minutos de cada sesión. Puede considerarse como tardía si es reportada antes de los 10 primeros minutos. Ausencias o tardías pueden justificarse según normativa y en lapso requerido. Una tardía injustificada vale por un tercio de una ausencia. Un excelente promedio (95%+) de asistencia puede ser usado como un puntaje positivo de apreciación.

2. No se aceptan justificaciones de ausencias asociadas con temas laborales, de tránsito o personales asociados con terceros que no sean relevantes a la universidad. Toda justificación debe ser formal y válida según normativa. En general, no se aceptan justificaciones de ausencias al curso provocadas por

tránsito, choques con otros cursos o exámenes o actividades de estos que coincidan con el horario de matrícula. La ausencia en una revisión de proyecto debe ser igualmente justificada como en un examen.

3. Se trabajará en grupos de 4 (o menos solo por situación inevitable). Los grupos se forman una sola vez y así permanecerán todo el semestre. Los proyectos e investigaciones son grupales. Los grupos son solo entre estudiantes de un mismo NRC. Durante la revisión de un proyecto cualquier miembro deberá poder responder sobre su contribución individual al trabajo y será evaluado en función de su respuesta. Cada grupo tendrá un coordinador que podrá responder sobre el avance del grupo y comunicar a tiempo situaciones sobre el trabajo y su avance.

4. Durante cualquier clase, puede haber evaluaciones cortas individuales y grupales. Se pueden hacer preguntas con valor evaluativo en cada clase con relación a la materia vista en clase anterior. Esas evaluaciones entran el rubro de quices o apreciación, según sea el caso.

5. La participación en clase es tomada en cuenta como parte de la apreciación.

6. Podrá haber espacio para propuestas de trabajos extra que le permitan al estudiante demostrar un deseo de mejoramiento adicional y aprendizaje extendido. Deberán ser aceptados por el profesor primero. El promedio de los extras no podrá exceder un 5% de la nota final absoluta (antes del redondeo a base 10). Cada extra podrá tener valor distinto según el tema o complejidad.

7. En los exámenes no se usa la computadora, pues se evalúan conceptos. Se hacen en cuaderno tradicional, respondiendo a propio puño y letra. Pueden incluir escribir código a mano y/o explicar o completar código.

8. Para los proyectos y las investigaciones se entregará, con la debida anticipación a su revisión, una especificación (SPEC) con los requerimientos y productos mínimos esperados y criterios de revisión y evaluación, que comprenderán calidad del alcance logrado, desarrollo original, ejemplos,

respeto a la propiedad intelectual y estructura formal del entregable. La revisión siempre será a criterio y experiencia profesional del profesor.

9. El curso asume desde muy temprano conocimientos de algoritmia, estructuras discretas y de datos básicos (listas, pilas, colas árboles, grafos), recursión, orientación a objetos y patrones, y programación en general de acuerdo con el contenido y los requisitos. Estos incluyen temas básicos de concurrencia y asincronía.

10. El uso de IA como herramienta en el aprendizaje y (auto)estudio es aceptado y promovido. Sin embargo, en el caso de productos con valor evaluativo debe ser autorizado explícitamente por el docente en ese caso se debe declarar por el estudiante y constar como referencia explícita la forma en que se usó la IA. Lo contrario podrá considerarse como copia/plagio según normativa.

VII. Cronograma

El cronograma mostrado abajo es **sugerido** busca enunciar los contenidos y objetivos de la carta. Esta es muy amplia y, por ende, según el avance mostrado por los estudiantes puede requerir ajustes para graduar y ajustar correctamente el aprendizaje. Por el tipo de tema cubierto, es irreal pensar en un cronograma absolutamente rígido y secuencial en cuanto a sesión y temática y la cobertura deseada.

Las fechas de evaluaciones o actividades con valor evaluativo pueden variarse (hacia adelante) si el avance del curso así lo requiere. Siempre serán aplicadas solo en los horarios matriculados.

Un tema indicado como especial**,** es flexible y variable. Se tratará de alcanzarlo hasta donde sea posible según el tiempo lo permita. El nivel de cobertura dependerá del tiempo disponible luego de los temas más centrales y obligatorios y el avance del curso.

Cada sesión incluye clase magistral y práctica de laboratorio acorde los temas aprendidos en la proporción en promedio de al menos 3:1.

Los recursos bibliográficos son principalmente apuntes del docente en PDF que se entregarán, junto con ejercicios de autoestudio. Las pizarras de las clases, si el equipo disponible lo permite, serán digitalizadas y entregadas en formato PDF. Los recursos

podrán ser acompañados de videos asincrónicos complementarios y o tutorías. Las actividades exposición de temáticas a nivel conceptual y su ilustración usando herramientas y ejercicios relevantes.

Sesión	Fecha	Tipo Sesión	Contenidos/Aprendizajes Integrales	Actividades	Recursos Didácticos
1	21-07-25	Presencial	**Presentación Curso**	Entrega Carta Estudiante/anexo	Carta, Anexo y material inicial
2	24-07-25	Presencial	Contexto e Historia lenguajes y paradigmas I	Clase Introducción: Lenguajes, OOP, FP LP. Declarativo vs Operacional. Tendencias. Historia.	Material Introducción A
3	28-07-25	Presencial	Contexto e Historia lenguajes y paradigmas II		Material Introducción B
4	31-07-25	Presencial	Lenguajes: Abstracciones y Diseño I	Clases y Prácticas de Lenguajes: Datos vs Control, recursión, objetos, herencia, polimorfismo, metaprogramación, modularidad, desestructuración,	Material Lenguajes A
5	04-08-25	Presencial	Lenguajes: Abstracciones y Diseño II		Material Lenguajes B
6	07-08-25	Presencial	Lenguajes: Abstracciones y Diseño III		Material Lenguajes C
7	11-08-25	Presencial	Traducción, Evaluación, Compilación I	Clases y Prácticas de Traducción y Evaluación: Sintaxis y Semántica, datos y objetos, Tipificación, scope, binding, clausuras, herencia, Compilación, lexing/parsing y generadores, AST, generación código, máquinas virtuales	Material Traducción y Eval A
8	14-08-25	Presencial	Traducción, Evaluación, Compilación II		Material Traducción y Eval B
9	18-08-25	Presencial	Traducción, Evaluación, Compilación III		Material Traducción y Eval C
10	21-08-25	Presencial	Traducción, Evaluación, Compilación IV		Material Traducción y Eval D
11	25-08-25	Presencial	**Paradigma Lógico (LP)**	Clases y Prácticas de LP y Prolog: modelo lógico, sintaxis, clausulas de Horn, unificación, resolución, Árbol-SLD, recursion y backtracking, metaprogramación, parsing	Material Modelos LP
12	28-08-25	Presencial	LP y Prolog I		Material LP y Prolog A
13	01-09-25	Presencial	LP y Prolog II		Material LP y Prolog B
14	04-09-25	Presencial	LP y Prolog III		Material LP y Prolog C
15	08-09-25	Presencial	LP y Prolog IV (especial)		Material LP y Prolog D
16	11-09-25	Presencial	LP y Prolog V (especial)		Material LP y Prolog E
17	15-09-25	NA	**FERIADO**	NA	NA
18	18-09-25	Presencial	**I Examen Parcial**	Ejecución Examen	NA
19	22-09-25	Presencial	**Paradigma Funcional (FP)**	Clase introductoria a FP	Material Modelos de FP
20	25-09-25	Presencial	Cálculo Lambda I (especial)	Clases y Prácticas de Sintaxis, Reducciones, representación, normalización, decidibilidad	Material Cáclulo Lambda B
21	29-09-25	Presencial	Cálculo Lambda II (especial)		Material Cáclulo Lambda C
22	02-10-25	Presencial	Cálculo Lambda III (especial)		Material Cáclulo Lambda C
23	06-10-25	Presencial	**FP dinámica y JS**	Clases y Prácticas de FP Dinámica en JS: principios, sintaxis, objetos y tipos, prototipos, scope, OOP con clases, funciones y arrows, desetruturación, spread, módulos, generadores, asincronía promesas async/await	Material FP y JS A
24	09-10-25	Presencial	JS I		Material FP y JS B
25	13-10-25	Presencial	JS II		Material FP y JS C
26	16-10-25	Presencial	JS III (especial)		Material FP y JS D
27	20-10-25	Presencial	**FP Estática y Java**	JVM, bytecode, Sistema tipos, Generics, Lambdas, streams, interfaces funcionales, metaprogramación, anotaciones, records, pattern-matching	Material FP y Java A
28	23-10-25	Presencial	Java I		Material FP y Java B
29	27-10-25	Presencial	Java II		Material FP y Java C
30	30-10-25	Presencial	Java III		Material FP y Java D
31	03-11-25	Presencial	Práctica Examen	Práctica de solución de ejercicios	Material de Práctica Examen
32	06-11-25	Presencial	**II Examen Parcial**	Ejecución Examen	NA
33	10-11-25	Presencial	**Entrega y Revisión Proyecto II e Investigación**	NA	NA
34	13-11-25	NA	Publicación Promedios Finales	NA	NA

VIII. Recursos Bibliográficos

Aho A., Sethi Ravi, D. Ullman Jeffrey (2006). *Compilers: Principles, Techniques, and Tools.* Prentice Hall; 2nd edition.

Abelson, H., Sussman, G.J., Sussman, J. (1996). *Structure and Interpretation of Computer Programs*. MIT Press. 2nd edition.

Bramer, M.A. (2010). *Logic Programming with Prolog.* Springer-Verlag.

Bratko, **I.** (2001). *Programming for Artificial Intelligence*. Addison-Welsey. 3erd edition.

Clocksin, W. F. (1997). *Clause And Effect.* Springer-Verlag.

Clocksin, W. F., Mellish, C. S. (2003). *Programming in Prolog: Using the ISO Standard.* Springer-Verlag; 5th edition.

Hankin, C. (2004). *An Introduction to Lambda Calculi for Computer Scientists.* King's College Publications. Londres.

Lee, K.D. (2017). *Foundations of Programming Languages*. Springer-Verlag. 2dn edition.

Loría-Saénz, **C**. (2025) Material Curso Paradigmas de Programación. UNA EIF400 Paradigmas (requiere haber sido registrado(a) para tener acceso).

IX. Elementos Adicionales

Se recomienda a todos los estudiantes hacer uso de las bases de datos del SIDUNA (Sistema e Información Documental de la Universidad Nacional) para acceder a las fuentes recomendadas u otros materiales de ayuda con lo que puedan ayudar a complementar los distintos contenidos del curso y en particular las tareas de investigación.

Este documento será acompañado de una presentación **anexa a la Carta,** explicando donde sea requerido más sobre el enfoque, herramientas, expectativas, **sin contradecir a esta Carta**, anexo que **formará parte integral y vinculante de la misma**.