

**UNIVERSIDAD NACIONAL
DE
SANTIAGO DEL ESTERO**

**FACULTAD DE CIENCIAS EXACTAS Y
TECNOLOGÍAS**

PLANIFICACIÓN ANUAL 2024

ASIGNATURA: PROGRAMACIÓN LÓGICA Y FUNCIONAL

**LICENCIATURA EN SISTEMAS DE
INFORMACIÓN**

Plan de Estudio 2011 – Innovación Curricular 2022

Equipo cátedra:

Profesor Adjunto: Méndez Analía

JTP: Zarco Raquel

JTP: Fernández Reuter Beatriz



PLANIFICACIÓN DE LA ASIGNATURA

1. IDENTIFICACIÓN:

1.1.Nombre de Asignatura: Programación Lógica y Funcional.

1.2.Carrera: Licenciatura en Sistemas de Información.

1.3.Plan de Estudios: 2011 – Innovación Curricular 2022.

1.4.Año académico: 2024.

1.5.Carácter: Obligatoria.

1.6.Ubicación de la Asignatura en el Plan de Estudios

1.6.1.Módulo – Año: 6° Módulo – 3° Año.

1.6.2. Trayecto al que pertenece la Asignatura/Obligación Curricular

TRAYECTO	CARGA HORARIA PRESENCIAL
Ciencias Básicas y Específicas	-----
Algoritmos y Lenguajes	75 horas
Arquitectura, Sistemas Operativos y Redes	-----
Ingeniería de Software, Bases de Datos y Sistemas de Información	-----
Aspectos Sociales y Profesionales	-----
Otros contenidos	-----
CARGA HORARIA TOTAL DE LA ACTIVIDAD CURRICULAR	75 horas

Tabla 1: Carga horaria por trayecto

1.6.3. Correlativas

1.6.3.1. Anteriores:

Regulares: Estructuras de Datos y Programación II, Análisis II, Lógica II.

Aprobada: Estructuras de Datos y Programación I.

1.6.3.2. Posteriores:

Regular: Inteligencia Artificial.

Aprobada: Programación Avanzada.

1.7.Carga horaria:

1.7.1. Carga horaria semanal total: 5 horas.

1.7.2. Carga horaria semanal destinada a la formación práctica: 3 horas.



1.7.3. Carga horaria total dedicada a las distintas actividades de formación práctica: 45 horas.

1.8. Ámbitos donde se desarrollan las actividades de formación práctica a las que se hace referencia en el punto anterior: Las actividades de formación experimental se desarrollan en el ámbito de los Laboratorios de Informática dependiente del Departamento Académico de Informática, utilizando software específico para las actividades programadas en los talleres.

1.9. Indique la cantidad de comisiones en las que se dicta la asignatura: 1 comisión.

2. PRESENTACIÓN

2.1. Ubicación de la Asignatura como tramo de conocimiento de una disciplina.

Esta asignatura corresponde al Trayecto de Algoritmos y Lenguajes. Está orientada fundamentalmente al estudio, análisis y diseño de soluciones algorítmicas para resolver problemas computacionales a través de la programación lógica y funcional.

2.2. Conocimientos y habilidades previas que permiten encarar el aprendizaje de la Asignatura.

- Resolución de problemas con computadoras, aplicando lenguajes de programación y el Paradigma Orientado a Objeto, adquiridos en las asignaturas Estructuras de Datos y Programación I y Estructuras de Datos y Programación II.
- Estructuras de datos lineales, recursividad, tipos de datos recursivos y algoritmos recursivos, estudiados en Estructuras de Datos y Programación I.
- Estructuras de datos no lineales y estrategias de diseño de algoritmos, estudiados en Estructuras de Datos y Programación II.
- Sistemas sintácticos y semánticos de la lógica proposicional, esquemas de axiomas y axiomas y cálculo de predicados de primer orden, estudiados en Lógica II.
- Funciones vectoriales de variable real, funciones reales de variable vectorial, funciones vectoriales de un vector y función potencial, estudiados en Análisis II.

2.3. Aspectos del Perfil Profesional del Egresado a los que contribuye la asignatura.

Posee:

- Los conocimientos básicos (lógico-matemáticos y computacionales) para una adecuada fundamentación teórica de su quehacer profesional específico.
- Profundos conocimientos sobre Algoritmos y Lenguajes de Programación.

Está capacitado para:

- Identificar, formular y resolver problemas de Informática.
- Gestar, diseñar y desarrollar proyectos de Informática.
- Gestionar, planificar, ejecutar y controlar proyectos de Informática.
- Utilizar técnicas y herramientas de aplicación en la Informática.
- Generar desarrollos tecnológicos y/o innovaciones tecnológicas.



- Comunicarse de manera efectiva en el ámbito profesional.
- Actuar con ética y responsabilidad social en el ámbito profesional.

2.4. Integración horizontal y vertical con otras asignaturas.

Los alumnos en las asignaturas Estructuras de Datos y Programación I y Estructuras de Datos y Programación II, adquieren conocimientos de estructuras de datos lineales y no lineales, recursividad, algoritmos recursivos y el desarrollo de habilidades de programación de computadoras aplicando el paradigma de programación orientada a objetos.

En Programación Lógica y Funcional y en Programación Avanzada se progresa hacia el aprendizaje de otros paradigmas de programación de manera de lograr que el alumno formalice y extienda sus conocimientos con respecto a la programación.

En Teoría de la Computación los alumnos estudian respecto a conceptos tales como evaluación de computabilidad, complejidad computacional y análisis de Algoritmos.

En la Figura 1 se expresa la integración horizontal, con asignaturas del 3° Año de la carrera, y vertical, con asignaturas correlativas previas y posteriores, de la asignatura Programación Lógica y Funcional.

3. OBJETIVOS

- Que el alumno desarrolle las siguientes competencias específicas:
 - C1. Analizar problemas y formular soluciones algorítmicas utilizando el Paradigma Declarativo.
 - C2. Desarrollar programas utilizando el Paradigma Declarativo.
 - C3. Ejecutar y verificar proyectos de programación aplicando los principios de programación declarativa.
 - C4. Utilizar herramientas modernas de desarrollo de software que faciliten el desarrollo de la programación aplicando el Paradigma Declarativo.
 - C5. Capacidad para desarrollar proyectos aplicando el Paradigma de Programación Declarativa, innovando en el diseño y el desarrollo.
- Que el alumno desarrolle las siguientes competencias transversales:
 - C6. Colaborar en el desarrollo de actividades grupales de integración entre teoría y práctica considerando el Paradigma de Programación Declarativo.
 - C7. Comunicar de manera clara y precisa conceptos, procedimientos y soluciones relacionados con el Paradigma de Programación Declarativo.
 - C8. Identificar y evaluar los impactos éticos asociados con su práctica de programación declarativa, aplicando un pensamiento crítico y reflexivo.
 - C9. Desarrollar un aprendizaje compartido y continuo en el ámbito de la programación lógica y funcional tomando iniciativa para adquirir nuevos conocimientos y habilidades de manera autónoma.

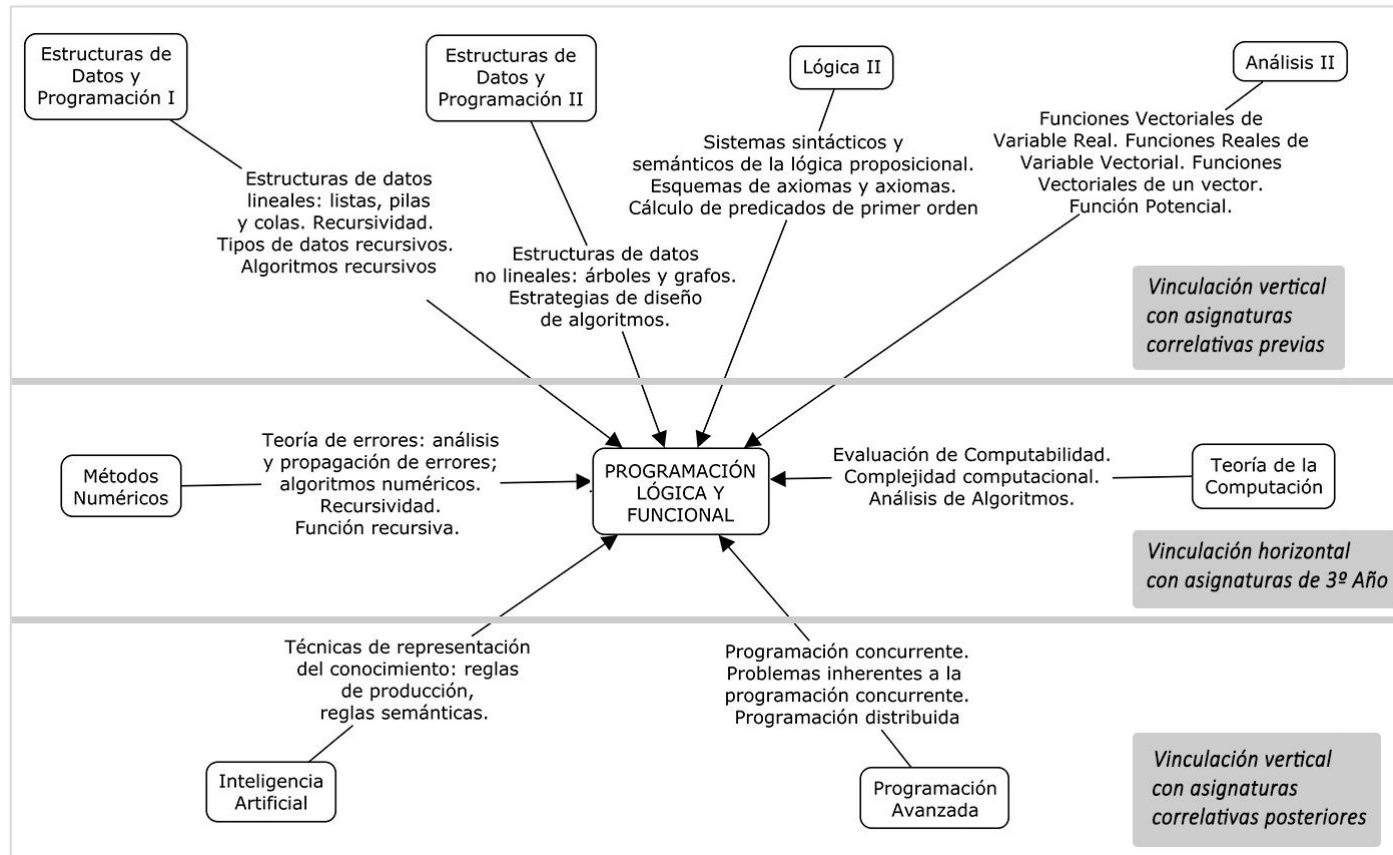


Figura 1. Integración horizontal y vertical de Programación Lógica y Funcional.



4. SELECCIÓN Y ORGANIZACIÓN DE CONTENIDOS

4.1. Contenidos mínimos establecidos en el Plan de Estudios para la Asignatura.

Paradigma Funcional. Principales características: transparencia referencial, evaluación diferida, recursividad, listas y funciones de orden superior. Cálculo lambda. Resolución de problemas y algoritmos sobre listas y árboles. Lenguajes funcionales: características, tipos de datos y aplicaciones. Paradigma lógico. Principales características: lógica proposicional, declaraciones, inversibilidad, unificación y backtracking. Resolución de problemas y algoritmos sobre grafos. Lenguajes lógicos: características, tipos de datos y aplicaciones.

4.2. Programa Sintético sobre la base de los contenidos mínimos

Unidad I: Introducción al Paradigma de Programación Declarativa

Paradigmas de programación. Concepto. Clasificación. Paradigmas declarativos: paradigma funcional y paradigma lógico.

Unidad II: Paradigma Funcional

Paradigma funcional. Principales características: transparencia referencial, evaluación diferida, recursividad, listas y funciones de orden superior. Concepto de programa en el paradigma funcional. Cálculo Lambda.

Unidad III: Programación Funcional

Lenguajes funcionales: características, tipos de datos y aplicaciones. Lenguaje de programación Haskell. Resolución de problemas y algoritmos sobre listas y árboles.

Unidades IV: Paradigma Lógico

Paradigma lógico. Principales características: Lógica proposicional, declaraciones, inversibilidad, unificación y backtracking. Predicados. Relaciones, hechos y reglas. Definición de programa en paradigma lógico. Diferencia entre una función y una relación.

Unidades V: Programación Lógica

Lenguajes lógicos: características, tipos de datos y aplicaciones. Lenguaje de programación Prolog. Resolución de problemas y algoritmos sobre grafos.

4.3. Articulación Temática de la Asignatura.

En la siguiente figura se presentan los principales conceptos a tratar en la asignatura y la relación entre los mismos.

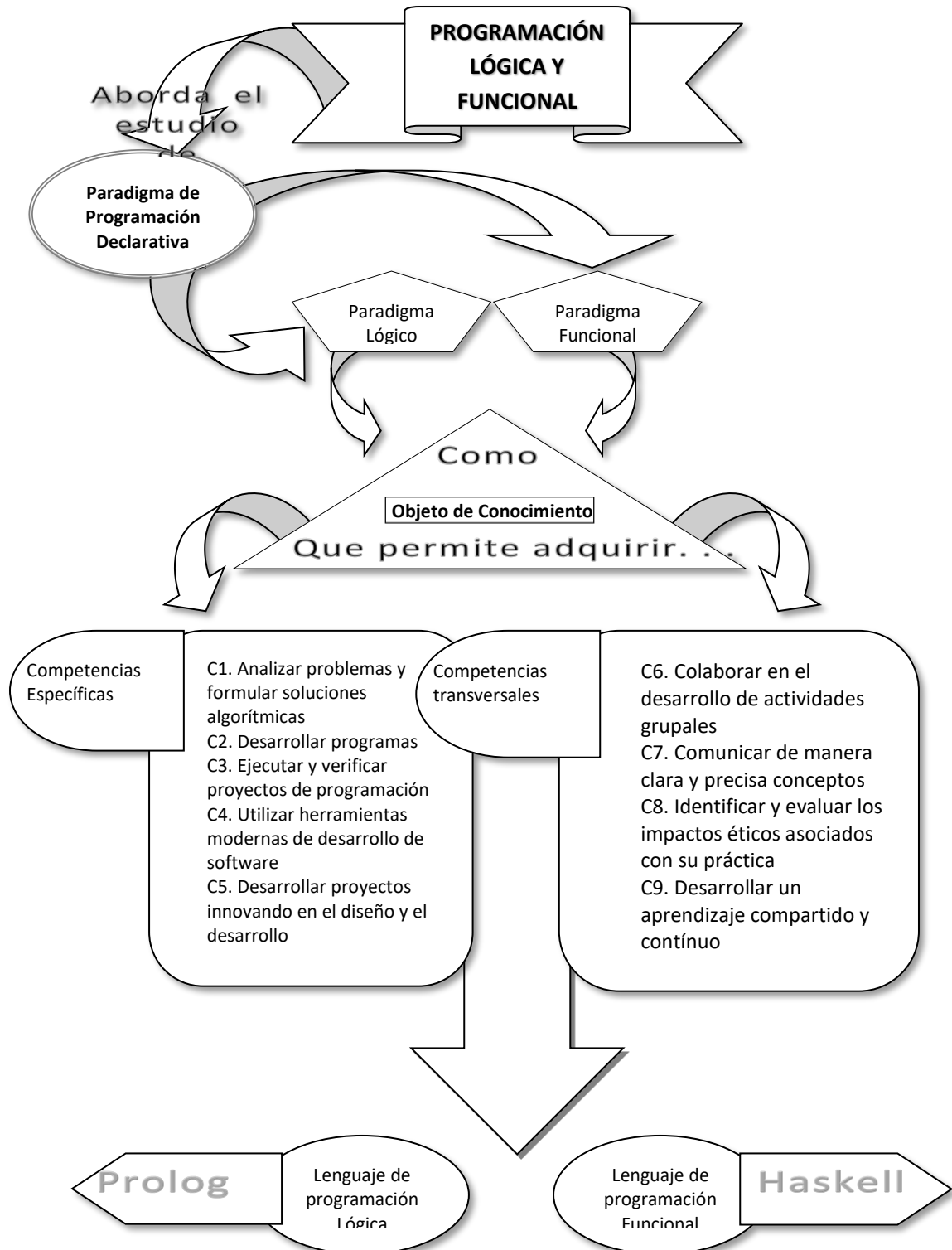


Figura 2. Articulación temática de Programación Lógica y Funcional.



4.4. Programa Analítico

Unidad I: Introducción al Paradigma de Programación Declarativa

Paradigmas de programación. Concepto. Clasificación. Paradigmas declarativos: paradigma funcional y paradigma lógico.

Unidad II: Paradigma Funcional

Paradigma funcional. Principales características: transparencia referencial, evaluación diferida, recursividad, listas y funciones de orden superior. Concepto de programa en el paradigma funcional. Concepto de función. Definición de funciones. Estructuras de datos. Tuplas. Cálculo Lambda. Polimorfismo y tipos genéricos.

Unidad III: Programación Funcional

Lenguajes funcionales: características, tipos de datos y aplicaciones. Ventajas de los lenguajes funcionales. Lenguaje de programación Haskell. Notación de listas por comprensión. Operadores infijos y prefijos. Reglas de precedencia. Funciones para el manejo de listas. Funciones para el manejo de tuplas. Resolución de problemas y algoritmos sobre listas y árboles.

Unidades IV: Paradigma Lógico

Paradigma lógico. Principales características: Lógica proposicional, declaraciones, inversibilidad, unificación y backtracking. Predicados. Razonamiento y silogismos. Relaciones, hechos y reglas. Definición de programa en paradigma lógico. Diferencia entre una función y una relación. Concepto de variable o incógnita. Estructuras de datos: listas. Pattern matching.

Unidades V: Programación Lógica

Lenguajes lógicos: características, tipos de datos y aplicaciones. Ventajas de los lenguajes lógicos. Lenguaje de programación Prolog. Estructura de un programa en Prolog. Predicadores de orden superior. Functores. Polimorfismo. Resolución de problemas y algoritmos sobre grafos.

4.5. Cronograma para el desarrollo de las Unidades Temáticas.

UNIDAD	CARGA HORARIA	CRONOGRAMA DE DICTADO
Unidad I: Introducción al Paradigma de Programación Declarativa	4	Semanas 1 y 2
Unidad II: Paradigma Funcional	6	Semanas 3 a 8
Unidad III: Programación Funcional	7	Semanas 3 a 8, y 15
Unidad IV: Paradigma Lógico	6	Semanas 9 a 14
Unidad V: Programación Lógica	7	Semanas 9 a 15
TOTAL	30	

Tabla 2: Cronograma para el desarrollo teórico de las unidades temáticas



5. FORMACIÓN PRÁCTICA

5.1. Descripción de las actividades de formación práctica

Las actividades prácticas de la asignatura se desarrollan en Talleres de Programación organizados temáticamente:

- Taller de Programación Funcional.
- Taller de Programación Lógica.

A continuación se describen cada uno de los Talleres.

TALLER DE PROGRAMACIÓN FUNCIONAL

Contenidos

Introducción al entorno de programación del lenguaje Haskell. Expresiones aritméticas y prioridad de operadores. Tuplas. Tipos de datos. Sintaxis: Patrones, Guardas, Cláusula Where, Bindings, Expresiones Case. Módulos. Tipos definidos por el usuario. Árboles. Entrada y Salida. Introducción a la programación funcional en lenguaje Python. Tuplas. Listas. Listas por comprensión. Aplicación parcial. Funciones lambda. Recursión. Funciones de orden superior. Composición de funciones.

Metodología

Este taller será de carácter teórico práctico, con actividades prácticas guiadas, resolución de ejercicios de programación en aula, y el desarrollo de proyectos domiciliarios. Las clases se desarrollarán en el Laboratorio de Informática.

Evaluación

Para aprobar el Taller de Programación Funcional el alumno deberá completar las actividades prácticas propuestas, individuales y grupales, y luego presentar y defender satisfactoriamente un proyecto de programación. La escala de valoración a emplear será cualitativa dicotómica (aprobado – desaprobado).

Instrumentos para la actividad

Para desarrollar esta actividad de formación experimental es necesario contar con los siguientes recursos:

- Espacio de la asignatura en la plataforma CUV (Centro Universitario Virtual).
- Laboratorio de computadoras con la instalación de:
 - Lenguaje de programación Haskell.
 - Interfaz WinGHCI de Haskell para Windows.
 - Lenguaje de programación Python.



TALLER DE PROGRAMACIÓN LÓGICA

Contenidos

Introducción al lenguaje de programación Prolog. Elementos del lenguaje: variables, términos, operadores, hechos, reglas. Cláusulas de Horn. Tipos de Datos simples. Objetos de datos estructurados. Concepto y utilización de las listas. Recursión. El corte en Prolog. Predicados de control. Unificación. Entrada/salida.

Metodología

El desarrollo del taller será de carácter teórico práctico, con actividades prácticas guiadas, resolución de ejercicios de programación en aula, y el desarrollo de proyectos domiciliarios. Las clases se desarrollarán en el Laboratorio de Informática.

Evaluación

Para aprobar el Taller de Programación Lógica el alumno deberá completar las actividades prácticas propuestas, individuales y grupales, y luego presentar y defender satisfactoriamente un proyecto de programación. La escala de valoración a emplear será cualitativa dicotómica (aprobado – desaprobado).

Instrumentos para la actividad

Para desarrollar esta actividad de formación experimental es necesario contar con los siguientes recursos:

- Espacio de la asignatura en la plataforma CUV (Centro Universitario Virtual).
- Laboratorio de computadoras con la instalación de:
 - Lenguaje de programación SWI-Prolog.

5.2. Formación en Ejes Transversales

En la Tabla 3 se relacionan cada uno de los Ejes Transversales de Formación con las actividades y los resultados de aprendizaje esperados.

Tabla 3: Formación en Ejes Transversales

EJE	ACTIVIDADES	RESULTADOS DE APRENDIZAJE	GPT
Identificación, formulación y resolución de problemas de informática	A1. Expresar la resolución de problemas mediante el uso de especificaciones formales de algoritmos utilizando lenguajes como Haskell, Python y Prolog, definiendo los tipos de datos, las funciones y la reglas lógicas, según corresponda.	RA.1. Desarrolla especificaciones detalladas y precisas de algoritmos en lenguajes declarativos, asegurando que las soluciones sean comprensibles y replicables.	Alto



Tabla 3: Formación en Ejes Transversales

EJE	ACTIVIDADES	RESULTADOS DE APRENDIZAJE	GPT
Concepción, diseño y desarrollo de proyectos de informática	A2. Codificar algoritmos utilizando las características funcionales de Haskell y Python, como la inmutabilidad, recursión y funciones de orden superior, y las reglas y predicados en Prolog para resolver problemas lógicos, incluyendo búsqueda y resolución de restricciones.	RA.2. Implementa correctamente soluciones algorítmicas utilizando programación funcional en los lenguajes Haskell y Python y utilizando programación lógica en el lenguaje Prolog.	Medio
Gestión, planificación, ejecución y control de proyectos de informática	A3. Aplicar las técnicas de programación funcional y lógica en un entorno de proyecto real.	RA. 3. Ejecuta y verifica proyectos de programación utilizando Haskell, Python y Prolog, aplicando principios de programación funcional y programación lógica de manera efectiva.	Medio
Utilización de técnicas y herramientas de aplicación en la informática	A4. Usar una GUI (interface gráfica de usuario), así como entornos de desarrollo integrado (IDEs), para la codificación, depuración y prueba de programas aplicando el Paradigma Declarativo.	RA.4.1. Utiliza una GUI (interface gráfica de usuario), tal como WinGhci, para desarrollar programas en lenguaje Haskell y el IDE Pycharm para desarrollar programas en lenguaje python, aplicando programación funcional.	Alto
		RA.4.2. Utiliza el entorno de programación SWI-Prolog para desarrollar programas en lenguaje Prolog, aplicando programación lógica.	
Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.	A5. Desarrollar proyectos de programación para resolver problemas utilizando el Paradigma de Programación Declarativa innovando en el diseño y desarrollo.	RA. 5. Demuestra capacidad para innovar en el diseño y desarrollo de aplicaciones de software, aplicando principios de programación funcional y lógica.	Alto
Fundamentos para el desempeño en equipos de trabajo	A6. Colaborar con su equipo en el desarrollo de tareas de integración entre teoría y práctica considerando el Paradigma de Programación Declarativo.	RA.6. Colabora participativamente en el desarrollo de actividades de revisión teórica y actividades de resolución de problemas aportando al equipo de trabajo sus habilidades y conocimientos sobre los Paradigmas de Programación Declarativos.	Alto
Fundamentos para la comunicación efectiva	A7. Elaborar y exponer presentaciones de análisis y revisión sobre soluciones de programación lógica y funcional.	RA.7. Realiza presentaciones escritas y orales en las que expone y defiende soluciones de programación lógica y funcional, demostrando dominio del contenido y capacidad de respuesta a preguntas técnicas.	Medio



Tabla 3: Formación en Ejes Transversales

EJE	ACTIVIDADES	RESULTADOS DE APRENDIZAJE	GPT
Fundamentos para la acción ética y responsable.	A8. Evaluar las decisiones tomadas en el diseño e implementación de programas aplicando el Paradigma de Programación Declarativo, analizando su alineación con principios éticos.	RA.8. Establece conclusiones sobre su práctica de programación, considerando los impactos éticos de las soluciones desarrolladas aplicando el Paradigma de Programación Declarativo.	Medio
Fundamentos para el aprendizaje continuo	A9. Analizar, debatir y discutir sobre temas relacionados con la programación declarativa colaborando en la resolución de problemas y fomentando un aprendizaje compartido y continuo.	RA.9. Desarrolla habilidades de aprendizaje compartido y continuo en el campo de la programación lógica y funcional, identificando y adquiriendo nuevos conocimientos de manera autónoma y de manera colaborativa.	Medio

(*) GPT: Grado de Profundidad en el Tratamiento.

5.3. Cronograma de formación práctica

ACTIVIDAD		CARGA HORARIA	CRONOGRAMA DE DESARROLLO
Taller de Programación Funcional	Actividad Práctica 1: Introducción al lenguaje Haskell	6	Semanas 1 a 2
	Actividad Práctica 2: Listas por comprensión. Funciones.	6	Semanas 3 a 4
	Actividad Práctica 3: Recursión.	3	Semana 5
	Actividad Práctica 4: Funciones de orden superior.	6	Semanas 6 a 7
Taller de Programación Lógica	Actividad Práctica 1: Base de conocimientos.	6	Semana 8 a 9
	Actividad Práctica 2: Recursividad.	6	Semanas 10 a 11
	Actividad Práctica 3: Listas.	6	Semanas 12 a 13
	Actividad Práctica 4: Predicados de orden superior.	6	Semanas 14 a 15
TOTAL		45	

Tabla 4: Cronograma para el desarrollo de las Actividades Prácticas



6. BIBLIOGRAFÍA.

Tabla 5: Bibliografía

TÍTULO	AUTOR/ES	EDITORIAL	EJEMPLARES DISPONIBLES	AÑO DE EDICIÓN
Empezando a usar: Una guía paso a paso	Alonso, J. C. & Ocampo, M. P.	Editorial Universidad Icesi.	eLibro	2024
	ENLACE DE ACCESO	https://elibro.net/es/lc/unsebiblio/titulos/225843		
Introducción a la Programación funcional con Haskell	Bird Richard	Prentice Hall	1 en Centro de Documentación	2000
Bases de conocimiento con Prolog	Garzón Torres, N. M. & Torres Soler, L. C	Editorial Universidad Autónoma de Colombia	eLibro	2018
	ENLACE DE ACCESO	https://elibro.net/es/lc/unsebiblio/titulos/160376		
Lógica para la computación	Luis de Ledesma	AlfaOmega	1 en Centro de Documentación	2010
Programación Lógica: Teoría y practica	Pascual Julián - Iranzo, Maria Alpuente	Prentice Hall	1 en Centro de Documentación	2007
Razonando con Haskell	Ruiz Blas, Gallardo Jose, Guerrero Pablo, Gutierrez Francisco	Editorial PARANINFO	1 en Centro de Documentación	2007
Introducción a los lenguajes prolog y lisp: teoría y práctica	Vázquez Torres, F	Instituto Politécnico Nacional	eLibro	2010
	ENLACE DE ACCESO	https://elibro.net/es/lc/unsebiblio/titulos/72666		
Programming in Prolog Using the ISO Standard	William F. Clocksin, Christopher S. Mellish	Springer	1 en Centro de Documentación	2003

7. ESTRATEGIAS METODOLÓGICAS.

7.1. Aspectos pedagógicos y didácticos

La metodología de enseñanza que se aplicará en el aula será llevada a cabo en un espacio de diálogo y construcción, en el que se trabaje interactuando permanentemente, y de esta manera tanto los alumnos como el docente se consideran fuente de información.



En las clases teóricas se aplicará el método expositivo / lección magistral donde la exposición explicativa se realizará con el diseño de las clases en PowerPoint o con el uso del pizarrón, usando como recursos didácticos: esquemas, tablas, gráficos, etc. Cada contenido desarrollado es mediado para su mejor comprensión y con el fin de propiciar el diálogo y discusión; además se realizan ejercicios de aplicación con el fin de aclarar conceptos, técnicas y métodos a utilizar.

Durante el desarrollo del Taller de Lenguaje de Programación Funcional y el desarrollo del Taller de lenguaje de programación Prolog se profundiza en una práctica de formación experimental que permita encontrar soluciones a los problemas planteados en las clases teóricas.

En la práctica se utiliza software libre; se procura lograr el desarrollo de habilidades y capacidades prácticas, y preparar a los alumnos para enfrentar los distintos cambios del mundo tecnológico.

Las actividades de estudio tanto prácticas como de tipo teórico serán apoyadas y desarrolladas mediante la plataforma Moodle del Centro Universitario Virtual (CUV). Mediante esta plataforma se utilizarán herramientas tales como foros, wikis y cuestionarios evaluativos y de diagnóstico, así como la presentación digital de las actividades aplicativas teóricas y los proyectos propuestos en las prácticas de formación experimental.

En el desarrollo de las actividades áulicas se destaca el uso de las instalaciones del laboratorio de computadoras, tanto para llevar a cabo ejercicios prácticos respectivos a los temas en estudio, así como para el uso de herramientas de apoyo didáctico.

Las actividades áulicas se caracterizarán principalmente por una integración entre los conceptos teóricos impartidos y su aplicación en ejercicios prácticos.

7.2. Mecanismos para la integración de docentes

Se plantea principalmente la integración con las asignaturas Estructuras de Datos y Programación I y Estructuras de Datos y Programación II correspondientes al segundo año de la carrera de Licenciatura en Sistemas de Información.

En ambas asignaturas los alumnos adquieren habilidades en el desarrollo de la programación de computadoras aplicando el paradigma de Programación Orientada a Objetos en la resolución de problemas y algoritmos.

La coordinación entre las asignaturas se realiza mediante encuentros anuales en los que se revisan los logros alcanzados en la comprensión y aplicación de los paradigmas de programación estudiados y sus actividades prácticas.

7.3. Recursos Didácticos

Se utilizarán diversos recursos didácticos a fin de lograr los objetivos propuestos. Se prevé la utilización de recursos didácticos: pizarra, libros, apuntes, Internet, laboratorio de informática, diapositivas, retroproyector o videoprojector y softwares, tales como:

- Lenguaje Haskell.



- Interfaz WinGHCI de Haskell para Windows.
- Lenguaje Prolog.
- Lenguaje de programación SWI-Prolog.
- Sistema operativo Windows.
- Lenguaje de programación Python.

Se contará con el espacio de la asignatura en la plataforma Moodle del Centro Universitario Virtual (CUV), que permitirá al alumno acceder a los recursos digitales, apoyo de prácticas, etc. También brindará a los alumnos un canal de comunicación permanente donde podrán acceder a información actualizada de la asignatura: fechas de evaluaciones, resultados de parciales, condición final de la cursada, etc.

8. EVALUACIÓN

8.1. Evaluación Diagnóstica

Teniendo en cuenta que la evaluación diagnóstica no sólo es una estimación, sino que tiene como propósito contribuir al aprendizaje, se llevará a cabo una única evaluación diagnóstica, al comienzo de las clases, cuya finalidad será determinar el nivel de conocimientos y habilidades previas que permitan encarar el aprendizaje de la asignatura.

Se evalúan contenidos adquiridos en las asignaturas Lógica II, Estructuras de Datos y Programación I y Estructuras de Datos y Programación II.

La evaluación diagnóstica será especialmente diseñada, individual, escrita y objetiva. Se hará una prueba de opción múltiple para que el alumno marque la opción correcta. El nivel de calificación será cualitativa politómica (nivel medio, bajo o alto).

8.2. Evaluación Formativa

La evaluación formativa es de carácter continuo y está dirigida a evaluar los procesos de enseñanza y de aprendizaje durante todo el desarrollo de la asignatura. Para ello se tendrá en cuenta el desempeño que demuestren los alumnos en la realización de las actividades planteadas, a fin de encarar, si fuera necesario, acciones correctivas.

Se aplicarán herramientas disponibles a través de la plataforma del CUV (Centro de Universitario Virtual). En particular se utilizarán herramientas foro, wiki, encuesta y cuestionario.

Esta modalidad de evaluación permitirá identificar la evolución en el aprendizaje de los alumnos y el grado de impacto de la propuesta educativa que lleva a cabo la cátedra.



8.3. Evaluación Parcial

8.3.1. Programa de Evaluaciones Parciales

EVALUACIÓN	TEMA	MODALIDAD	SEM	FECHA	DEVOLUCIÓN DE RESULTADO
Parcial Teórico 1	Paradigma Funcional	Individual. Cuestionario multiple choice mediante plataforma del CUV.	7	26/09	26/09
Recuperatorio 1			8	03/10	03/10
Parcial Teórico 2	Paradigma Lógico		13	07/11	07/11
Recuperatorio 2			14	14/11	14/11

Tabla 6: Descripción y cronograma de Evaluaciones Parciales

8.3.2. Criterios de Evaluación

Respecto de las evaluaciones Parcial Teórico 1, Parcial Teórico 2 y sus respectivos recuperatorios, así como el Trabajo Integrador Final, el criterio de evaluación es el siguiente:

- En relación al Resultado de Aprendizaje RA.7.:

CE.7. El estudiante demuestra conocimiento de la programación lógica y funcional, explicando detalles complejos con confianza y precisión y respondiendo correctamente las preguntas técnicas planteadas.

Los Criterios de Evaluación generales que serán aplicados en el Taller de Programación Funcional y en el Taller de Programación Lógica, son los siguientes:

- En relación al Resultado de Aprendizaje RA.3:

CE.3. El estudiante es capaz de diseñar y ejecutar pruebas adecuadas para verificar la corrección de su implementación, utilizando técnicas de depuración adecuadas.

- En relación al Resultado de Aprendizaje RA.5:

CE.5. El estudiante utiliza conceptos avanzados de programación funcional (como funciones de orden superior, currificación, etc.) y de programación lógica (como resolución de metas, unificación, etc.) de manera eficaz en sus proyectos.

- En relación al Resultado de Aprendizaje RA.7:

CE.7. El estudiante demuestra conocimiento de la programación lógica y funcional, explicando detalles complejos con confianza y precisión y respondiendo correctamente las preguntas técnicas planteadas.



- En relación al Resultado de Aprendizaje RA.9:

CE.9. El estudiante participa activamente en discusiones de grupo, talleres o foros, compartiendo conocimientos y colaborando en la resolución de problemas, demostrando capacidad de aplicar los conocimientos adquiridos de manera autónoma y colaborativa en la resolución de problemas de programación lógica y funcional.

Los Criterios de Evaluación específicos al Taller de Programación Funcional, son los siguientes:

- En relación al Resultado de Aprendizaje RA.1:

CE.1.1. El estudiante debe demostrar competencia en el uso de funciones de orden superior, recursión, y manejo de listas y otros tipos de datos funcionales en Haskell y Python, mostrando entendimiento del paradigma funcional.

- En relación al Resultado de Aprendizaje RA.2:

CE.2.1. El estudiante utiliza correctamente conceptos clave de programación funcional, como inmutabilidad, funciones puras, recursión y manejo de funciones de orden superior, resolviendo el problema planteado de manera correcta.

- En relación al Resultado de Aprendizaje RA.4:

CE.4.1. El estudiante utiliza de manera eficiente la GUI WinGhci para desarrollar programas en lenguaje Haskell y el IDE Pycharm para desarrollar programas en lenguaje Python, aplicando programación funcional.

Los Criterios de Evaluación específicos al Taller de Programación Lógica, son los siguientes:

- En relación al Resultado de Aprendizaje RA.1:

CE.1.2. El estudiante debe utilizar correctamente las técnicas de programación lógica en Prolog, como la unificación, backtracking, y la definición de reglas y hechos, para resolver problemas.

- En relación al Resultado de Aprendizaje RA.2:

CE.2.2. El estudiante emplea de manera adecuada los conceptos fundamentales de Prolog, tales como reglas, hechos, predicados, y el uso de la unificación y el backtracking, resolviendo el problema planteado de manera correcta.

- En relación al Resultado de Aprendizaje RA.4:

CE.4.2. El estudiante utiliza eficientemente el entorno de programación SWI-Prolog para desarrollar programas en lenguaje Prolog, aplicando programación lógica.

Respecto de las Actividades de Aula y Actividades Aplicativas, los criterios de evaluación son siguientes:

- En relación al Resultado de Aprendizaje RA.6.:

CE.6. El estudiante cumple las tareas asignadas dentro del equipo, entregando trabajos de calidad en los plazos establecidos.

- En relación al Resultado de Aprendizaje RA.7.:



CE.7. El estudiante demuestra conocimiento de la programación lógica y funcional, explicando detalles complejos con confianza y precisión y respondiendo correctamente las preguntas técnicas planteadas.

- En relación al Resultado de Aprendizaje RA.8:

CE. 8. El estudiante identifica y analiza cuestiones éticas relevantes en la práctica de la programación lógica y funcional.

- En relación al Resultado de Aprendizaje RA.9:

CE.9. El estudiante participa activamente en discusiones de grupo, talleres o foros, compartiendo conocimientos y colaborando en la resolución de problemas, demostrando capacidad de aplicar los conocimientos adquiridos de manera autónoma y colaborativa en la resolución de problemas de programación lógica y funcional.

Lo descripto se resume en la siguiente Tabla.

Tabla 7: Aplicación de los Criterios de Evaluación

RESULTADOS DE APRENDIZAJE	CRITERIOS DE EVALUACIÓN	EVALUACIÓN EN LA QUE SE APLICA
RA.1. Desarrolla especificaciones detalladas y precisas de algoritmos en lenguajes declarativos, asegurando que las soluciones sean comprensibles y replicables.	CE.1.1. El estudiante demuestra competencia en el uso de funciones de orden superior, recursión, y manejo de listas y otros tipos de datos funcionales en Haskell y Python, mostrando entendimiento del paradigma funcional.	Taller de Programación Funcional.
	CE.1.2. El estudiante debe utilizar correctamente las técnicas de programación lógica en Prolog, como la unificación, backtracking, y la definición de reglas y hechos, para resolver problemas.	Taller de Programación Lógica.
RA.2. Implementa correctamente soluciones algorítmicas utilizando programación funcional en los lenguajes Haskell y Python y utilizando programación lógica en el lenguaje Prolog.	CE.2.1. El estudiante utiliza correctamente conceptos clave de programación funcional, como inmutabilidad, funciones puras y manejo de funciones de orden superior, resolviendo el problema planteado de manera correcta.	Taller de Programación Funcional.
	CE.2.2. El estudiante emplea de manera adecuada los conceptos fundamentales de Prolog, tales como reglas, hechos, predicados, y el uso de la unificación y el backtracking, resolviendo el problema planteado de manera correcta.	Taller de Programación Lógica.



Tabla 7: Aplicación de los Criterios de Evaluación

RESULTADOS DE APRENDIZAJE	CRITERIOS DE EVALUACIÓN	EVALUACIÓN EN LA QUE SE APLICA
RA. 3. Ejecuta y verifica proyectos de programación utilizando Haskell, Python y Prolog, aplicando principios de programación funcional y programación lógica de manera efectiva.	CE.3. El estudiante es capaz de diseñar y ejecutar pruebas adecuadas para verificar la corrección de su implementación, utilizando técnicas de depuración adecuadas	Taller de Programación Funcional.
		Taller de Programación Lógica.
RA.4. Utiliza una GUI (interface gráfica de usuario), tal como WinGhci, para desarrollar programas en lenguaje Haskell y el IDE Pycharm para desarrollar programas en lenguaje python, aplicando programación funcional, y utiliza el entorno de programación SWI-Prolog para desarrollar programas en lenguaje Prolog, aplicando programación lógica.	CE.4.1. El estudiante utiliza de manera eficiente la GUI WinGhci para desarrollar programas en lenguaje Haskell y el IDE Pycharm para desarrollar programas en lenguaje Python, aplicando programación funcional.	Taller de Programación Funcional.
	CE.4.2. El estudiante utiliza eficientemente el entorno de programación SWI-Prolog para desarrollar programas en lenguaje Prolog, aplicando programación lógica.	Taller de Programación Lógica.
RA. 5. Demuestra capacidad para innovar en el diseño y desarrollo de aplicaciones de software, aplicando principios de programación funcional y lógica.	CE.5. El estudiante utiliza conceptos avanzados de programación funcional (como funciones de orden superior, currificación, etc.) y de programación lógica (como resolución de metas, unificación, etc.) de manera eficaz en sus proyectos.	Taller de Programación Funcional.
		Taller de Programación Lógica.
RA.6. Colabora participativamente en el desarrollo de actividades de revisión teórica y actividades de resolución de problemas aportando al equipo de trabajo sus habilidades y conocimientos sobre los Paradigmas de Programación Declarativos	CE.6. El estudiante cumple las tareas asignadas dentro del equipo, entregando trabajos de calidad en los plazos establecidos.	Actividades de aula Actividades aplicativas



Tabla 7: Aplicación de los Criterios de Evaluación

RESULTADOS DE APRENDIZAJE	CRITERIOS DE EVALUACIÓN	EVALUACIÓN EN LA QUE SE APLICA
RA.7. Realiza presentaciones escritas y orales en las que expone y defiende soluciones de programación lógica y funcional, demostrando dominio del contenido y capacidad de respuesta a preguntas técnicas.	CE.7. El estudiante demuestra conocimiento de la programación lógica y funcional, explicando detalles complejos con confianza y precisión y respondiendo correctamente las preguntas técnicas planteadas.	Taller de Programación Funcional.
		Taller de Programación Lógica.
		Parcial Teórico 1 / Recuperatorio 1 Parcial Teórico 2 / Recuperatorio 2 Trabajo Integrador Final
RA.8. Establece conclusiones sobre su práctica de programación, considerando los impactos éticos de las soluciones desarrolladas aplicando el Paradigma de Programación Declarativo.	CE. 8. El estudiante identifica y analiza cuestiones éticas relevantes en la práctica de la programación lógica y funcional.	Actividades de aula Actividades aplicativas
RA.9. Desarrolla habilidades de aprendizaje compartido y continuo en el campo de la programación lógica y funcional, identificando y adquiriendo nuevos conocimientos de manera autónoma y de manera colaborativa.	CE.9. El estudiante participa activamente en discusiones de grupo, talleres o foros, compartiendo conocimientos y colaborando en la resolución de problemas, demostrando capacidad de aplicar los conocimientos adquiridos de manera autónoma y colaborativa en la resolución de problemas de programación lógica y funcional.	Taller de Programación Funcional.
		Taller de Programación Lógica.

8.3.3. Escala de Valoración

La escala de valoración a emplear para las evaluaciones parciales y los recuperatorios será cuantitativa politómica. Escala: 1 al 10. El puntaje mínimo para aprobar los parciales es de cincuenta (5) puntos, sobre una calificación máxima de 10.

8.4. Evaluación Integradora

La evaluación integradora se llevará a cabo a través de un Trabajo Integrador Final en el que los alumnos deberán generar un informe respecto a las conclusiones a las que arribaron mediante las prácticas realizadas aplicando programación funcional y programación lógica, incluyendo una comparación entre ambas.



8.5. Evaluación Sumativa

8.5.1. Condiciones para lograr la promoción sin Examen Final de la Asignatura. (Rige la Resolución HCD N° 135/00)

Para promocionar la asignatura el alumno deberá:

- Asistir como mínimo al 80% del total de sesiones.
- Aprobar los dos (2) Parciales Teóricos con un puntaje ≥ 7 puntos.
- Presentar y aprobar todas las Actividades Áulicas y Actividades Aplicativas.
- Aprobar en primera instancia el Taller de Programación Funcional y el Taller de Programación Lógica.
- Presentar y exponer el Trabajo Integrador Final.

8.5.2. Condiciones para lograr la Regularidad de la Asignatura.

- Aprobar los dos (2) Parciales Teóricos o sus correspondientes Recuperatorios.
- Aprobar el Taller de Programación Funcional y el Taller de Programación Lógica, en primera instancia o en las actividades de recuperación.

8.6. Examen Final

En el examen final los alumnos serán evaluados sobre los contenidos teóricos previstos en el programa de la asignatura y su aplicación práctica. El examen podrá ser oral o escrito e individual.

8.7. Examen Libre

El examen final libre constará de tres etapas.

Primera Etapa: Presentación, prueba y defensa de un planteamiento práctico. El trabajo deberá ser presentado en soporte digital e impreso, cumplimentando las pautas establecidas en el enunciado, la presentación deberá realizarse al término del plazo de días hábiles indicado a partir de la entrega del enunciado.

Segunda Etapa: Aprobar una evaluación oral/escrita de contenidos teóricos del programa analítico.

Tercera Etapa: Aprobar una evaluación en computadora aplicando los contenidos teóricos y prácticos de la asignatura.

Méndez Analía

.....
Profesor responsable de Asignatura