

**UNIVERSIDAD NACIONAL
DE
SANTIAGO DEL ESTERO**

**FACULTAD DE CIENCIAS EXACTAS Y
TECNOLOGÍAS**

PLANIFICACIÓN ANUAL 2023

ASIGNATURA: PROGRAMACIÓN LÓGICA Y FUNCIONAL

**LICENCIATURA EN SISTEMAS DE
INFORMACIÓN**

Plan de Estudio:

Plan 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.

1.4. **Año académico:** 2023.

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 avanza en 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 3º Año de la carrera, y vertical, con asignaturas correlativas previas y posteriores, de la asignatura Programación Lógica y Funcional.

3. OBJETIVOS

3.1. Objetivos Generales

- Desarrollar creativamente soluciones algorítmicas seleccionando criteriosamente la alternativa adecuada, aplicando distintos paradigmas de programación.
- Emplear estructuras de control y estructuras de datos en la resolución de problemas, aplicando distintos paradigmas de programación.
- Verificar algoritmos desarrollados en los distintos paradigmas de programación.
- Aplicar distintos lenguajes de programación como herramientas computacionales para la resolución de problemas.

3.2. Objetivos Específicos

- Que el alumno desarrolle las siguientes competencias básicas:
 - Seleccionar y procesar la información necesaria para la resolución de problemas y verificar las soluciones encontradas.
- Que el alumno desarrolle las siguientes competencias específicas:
 - Aplicar los principios y técnicas de la Programación Declarativa identificando las principales características del Paradigma Declarativo.
 - Realizar la búsqueda creativa de soluciones algorítmicas y la selección criteriosa de la alternativa adecuada aplicando los mecanismos propios de los lenguajes de Programación Declarativa.

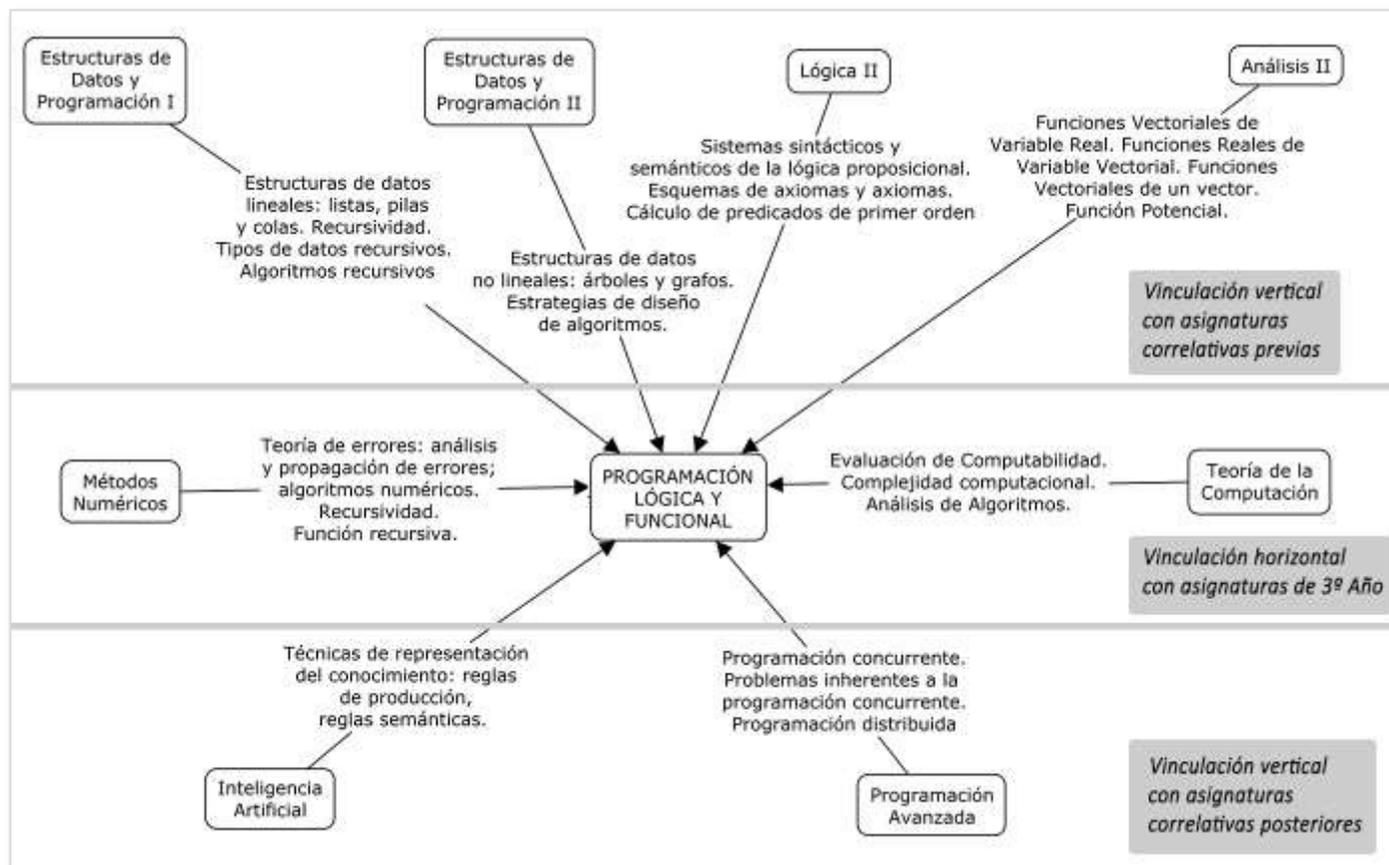


Figura 1. Integración horizontal y vertical de Programación Lógica y Funcional.



- Diseñar, codificar, ejecutar, depurar e interpretar programas utilizando lenguajes de programación propios del Paradigma Declarativo, tales como Haskell y Prolog, entre otros.
- Desarrollar especificaciones de programas simples aplicando la Programación Declarativa.
- Que el alumno desarrolle las siguientes competencias transversales:
 - Expresar de manera clara, comprensible y libre de errores de ortografía y gramática, en forma oral y escrita, los conceptos teóricos relacionándolos adecuadamente con su práctica.
 - Enfocar la comunicación oral y escrita en el intercambio y transmisión de los conocimientos.
 - Integrar participativamente el desarrollo de actividades de revisión teórica y actividades de resolución de problemas aportando sus habilidades y conocimientos al equipo de trabajo.
 - Desarrollar los trabajos asignados de manera colaborativa favoreciendo la resolución de los problemas planteados y el desarrollo de la dinámica grupal.
 - Identificar, plantear y resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad.
 - Desarrollar y apropiarse de nuevas habilidades y conocimientos como resultado del estudio y la práctica aplicando el Paradigma Declarativo.

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.

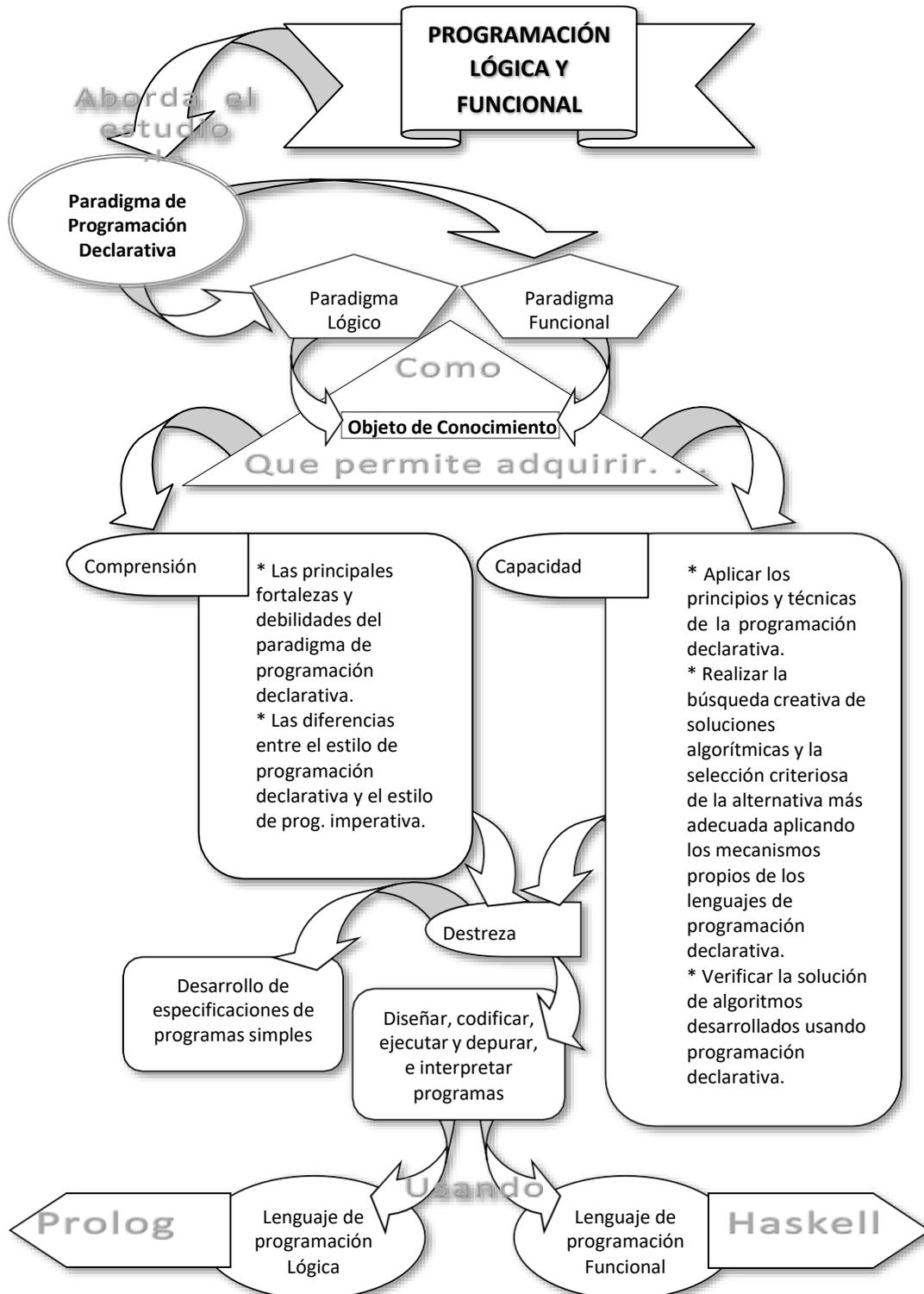


Figura 2. Articulación temática de Programación Lógica y Funcional.



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.

Objetivos.

Que los alumnos logren...

- Aplicar los principios y técnicas de la Programación Funcional.
- Realizar, en grupo e individualmente, con respeto y con responsabilidad ética, la búsqueda creativa de soluciones algorítmicas y la selección criteriosa de la alternativa adecuada aplicando los mecanismos propios de lenguajes de Programación Funcional, tales como Haskell y Python, seleccionando y procesando la información necesaria para resolver los problemas planteados y verificar las soluciones encontradas.



- Desarrollar especificaciones de programas simples aplicando la Programación Declarativa y usando diversos entornos de programación para diseñar, codificar, ejecutar, depurar e interpretar programas utilizando lenguajes de Programación Funcional, como Haskell y Python.
- Integrar participativamente el desarrollo de actividades de resolución de problemas aportando sus habilidades y conocimientos al equipo de trabajo, desenvolviéndose de manera colaborativa para favorecer la resolución de los problemas planteados y el desarrollo de la dinámica grupal.
- Trabajar de forma autónoma desarrollando inquietud para la eficiencia y el rigor, cumpliendo con las modalidades y tiempos de presentación de las actividades prácticas, reaccionando positivamente frente al error e identificando y verificando nuevas estrategias para alcanzar el logro exigido a fin de completar el desarrollo de la tarea recibida.

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).

Criterios de evaluación y aprobación

- Presentación en tiempo y forma de las tareas individuales y grupales, de acuerdo a lo solicitado en cada actividad planteada, desenvolviéndose con respeto y responsabilidad ética y participando colaborativamente en la resolución de los problemas planteados y el desarrollo de la dinámica grupal.
- Correcta aplicación de la Programación Declarativa en diversos entornos de programación para diseñar, codificar, ejecutar, depurar e interpretar programas utilizando lenguajes de Programación Funcional, como Haskell y Python.

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.

Objetivos.

- Aplicar los principios y técnicas de la Programación Lógica.
- Realizar, en grupo e individualmente, con respeto y con responsabilidad ética, la búsqueda creativa de soluciones algorítmicas y la selección criteriosa de la alternativa adecuada aplicando los mecanismos propios del lenguaje de Programación Lógica, Prolog, seleccionando y procesando la información necesaria para resolver los problemas planteados y verificar las soluciones encontradas.
- Desarrollar especificaciones de programas simples aplicando la Programación Declarativa para diseñar, codificar, ejecutar, depurar e interpretar programas utilizando el lenguaje de Programación Lógica, Prolog.
- Integrar participativamente el desarrollo de actividades de resolución de problemas aportando sus habilidades y conocimientos al equipo de trabajo, desenvolviéndose de manera colaborativa para favorecer la resolución de los problemas planteados y el desarrollo de la dinámica grupal.
- Trabajar de forma autónoma desarrollando inquietud para la eficiencia y el rigor, cumpliendo con las modalidades y tiempos de presentación de las actividades prácticas, reaccionando positivamente frente al error e identificando y verificando nuevas estrategias para alcanzar el logro exigido a fin de completar el desarrollo de la tarea recibida.

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).

Criterios de evaluación y aprobación

- Presentación en tiempo y forma de las tareas individuales y grupales, de acuerdo a lo solicitado en cada actividad planteada, desenvolviéndose con respeto y responsabilidad ética y participando colaborativamente en la resolución de los problemas planteados y el desarrollo de la dinámica grupal.



- Correcta aplicación de la Programación Declarativa para diseñar, codificar, ejecutar, depurar e interpretar programas utilizando el lenguaje de Programación Lógico, Prolog.

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	GRADO DE PROFUNDIDAD EN EL TRATAMIENTO
Identificación, formulación y resolución de problemas de informática	• Taller de Programación Funcional. • Taller de Programación Lógica.	<ul style="list-style-type: none">• Identificar, plantear y resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad.• Seleccionar y procesar la información necesaria para la resolución de problemas y verificar las soluciones encontradas.• Realizar la búsqueda creativa de soluciones algorítmicas y la selección criteriosa de la alternativa adecuada aplicando los mecanismos propios de los lenguajes de Programación Declarativa.	Alto
Concepción, diseño y desarrollo de proyectos de informática		<ul style="list-style-type: none">• Desarrollar especificaciones de programas simples aplicando la Programación Declarativa.	Medio
Gestión, planificación, ejecución y control de proyectos de informática		<ul style="list-style-type: none">• Diseñar, codificar, ejecutar, depurar e interpretar programas utilizando lenguajes de programación propios del Paradigma Declarativo, tales como Haskell, Python y Prolog, entre otros.	Medio



Tabla 3: Formación en Ejes Transversales (continuación)

EJE	ACTIVIDADES	RESULTADOS DE APRENDIZAJE	GRADO DE PROFUNDIDAD EN EL TRATAMIENTO
Utilización de técnicas y herramientas de aplicación en la informática	<ul style="list-style-type: none">Taller de Programación Funcional.	<ul style="list-style-type: none">Aplicar los principios y técnicas de la Programación Declarativa identificando las principales características del Paradigma Declarativo.Desarrollar programas aplicando el Paradigma Declarativo utilizando diversos entornos de programación.	Alto
Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.	<ul style="list-style-type: none">Taller de Programación Lógica.	<ul style="list-style-type: none">Diseñar, codificar, ejecutar, depurar e interpretar programas utilizando lenguajes de programación propios del Paradigma Declarativo, tales como Haskell, Python y Prolog, entre otros.	Alto
Fundamentos para el desempeño en equipos de trabajo	<ul style="list-style-type: none">Tareas de Estudio.Taller de Programación Funcional.Taller de Programación Lógica.	<ul style="list-style-type: none">Integrar participativamente el desarrollo de actividades de revisión teórica y actividades de resolución de problemas aportando sus habilidades y conocimientos al equipo de trabajo.Desarrollar los trabajos asignados de manera colaborativa favoreciendo la resolución de los problemas planteados y el desarrollo de la dinámica grupal.	Alto
Fundamentos para la comunicación efectiva	<ul style="list-style-type: none">Tareas de Estudio.	<ul style="list-style-type: none">Expresar de manera clara, comprensible y libre de errores de ortografía y gramática, en forma oral y escrita, los conceptos teóricos relacionándolos adecuadamente con su práctica.Enfocar la comunicación oral y escrita en el intercambio y transmisión de los conocimientos.	Medio
Fundamentos para la acción ética y responsable.	<ul style="list-style-type: none">Tareas de Estudio.Taller de Programación Funcional.Taller de Programación Lógica.	<ul style="list-style-type: none">Trabajar de forma autónoma desarrollando inquietud para la eficiencia y el rigor, cumpliendo con las modalidades y tiempos de presentación de las actividades teóricas y prácticas.Reaccionar frente al error identificando y verificando nuevas estrategias para alcanzar el logro exigido a fin de completar el desarrollo de la tarea recibida.Realizar la búsqueda creativa de soluciones algorítmicas en grupo e individualmente, con respeto y con responsabilidad ética.	Medio



Tabla 3: Formación en Ejes Transversales (continuación)

EJE	ACTIVIDADES	RESULTADOS DE APRENDIZAJE	GRADO DE PROFUNDIDAD EN EL TRATAMIENTO
Fundamentos para el aprendizaje continuo	<ul style="list-style-type: none">Taller de Programación Funcional.Taller de Programación Lógica.	<ul style="list-style-type: none">Desarrollar y apropiarse de nuevas habilidades y conocimientos como resultado del estudio y la práctica aplicando el Paradigma Declarativo.	Medio

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 usarR: Una guía paso a paso	Alonso, J. C. & Ocampo, M. P.	Editorial Universidad Icesi.	eLibro	2023
	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.

Las exposiciones teóricas se complementan con el desarrollo del Taller de Lenguaje de Programación Funcional y el desarrollo del Taller de lenguaje de programación Prolog a los efectos de lograr una práctica de formación experimental que permita encontrar las 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 wikis y cuestionarios evaluativos y de diagnóstico, así como la presentación digital de las actividades y 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 wiki, encuestas 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.

Se evaluará:

- Participación del alumno en clase y en las actividades individuales y grupales.
- Disposición y desempeño del alumno en la resolución de las actividades propuestas.
- Presentación en tiempo y forma de las actividades propuestas.
- Capacidad de resolución y de análisis de los problemas de carácter teórico y práctico que se le presenten al alumno.



8.3. Evaluación Parcial

8.3.1. Programa de Evaluaciones Parciales

EVALUACIÓN	TEMA	MODALIDAD	SEM	FECHA	DEVOLUCIÓN DE RESULTADO
Parcial 1	Paradigma Funcional	Individual. Cuestionario multiple choice mediante plataforma del CUV.	7	25/09	25/09
Recuperatorio 1			8	02/10	02/10
Parcial 2	Paradigma Lógico		13	06/11	06/11
Recuperatorio 2			14	13/11	13/11

Tabla 6: Descripción y cronograma de Evaluaciones Parciales

8.3.2. Criterios de Evaluación

ESTRUCTURA	OBJETIVO	CRITERIOS DE EVALUACIÓN
Preguntas teóricas de los siguientes tipos: <ul style="list-style-type: none">• Verdadero-Falso.• Respuesta múltiple.• Completar gráficas.• Completar frases.• Relacionar conceptos con sus definiciones.	Determinar si los alumnos han adquirido las competencias para: <ul style="list-style-type: none">• Identificar las principales características y las fortalezas y debilidades del paradigma de programación declarativa estudiado.• Reconocer las diferencias entre el estilo de programación declarativa aplicado y el estilo de programación imperativa.	<ul style="list-style-type: none">• Capacidad para reconocer aspectos de relevancia en la temática planteada.• Habilidad para relacionar, contrastar y vincular los contenidos estudiados.

Tabla 7: Estructura, objetivos y criterios de Evaluaciones Parciales

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 100.

El puntaje mínimo para aprobar los parciales es de cincuenta (50) puntos, sobre una calificación máxima de 100.



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)

No se prevé.

8.5.2. Condiciones para lograr la Regularidad de la Asignatura.

- Aprobar las evaluaciones parciales de teoría o sus correspondientes recuperatorios.
- Presentar todas las actividades de los talleres y aprobar los proyectos propuestos en las prácticas de formación experimental.
- Presentación en tiempo y forma de la evaluación integradora.

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