

INSTITUTO/S: Tecnología e Ingeniería

CARRERA/S: Licenciatura en Informática / Tecnicatura en Informática

MATERIA: Programación Funcional

NOMBRE DEL RESPONSABLE DE LA ASIGNATURA: Mariano Rean

EQUIPO DOCENTE: -----

CUATRIMESTRE: Licenciatura: 1^{ro} / Tecnicatura: 3^{ro}

AÑO: Licenciatura: 2^{do} / Tecnicatura: 2^{do}

PROGRAMA N°: 13 (Aprob. Por Cons.Directivo fecha XX)

Instituto/s: Tecnología e Ingeniería

Carrera/s: Licenciatura en Informática/ Tecnicatura en Informática

Nombre de la materia: Programación Funcional

Responsable de la asignatura y equipo docente: Mariano Rean

Cuatrimestre y año: Licenciatura: 1^{ro} del 2^{do} año/ Tecnicatura: 3^{ro} del 2^{do} año

Carga horaria: 4 horas por semana

Programa N°: 13

Código de la materia en SIU: 761

Programación Funcional

1. Fundamentación

La programación funcional es uno de los paradigmas más difundidos de programación, entre los cuales también debe mencionarse a la programación imperativa y la programación orientada a objetos. Precede históricamente a las dos últimas y se destaca por su alto nivel de abstracción y sus robustos fundamentos matemáticos. La formación de los estudiantes en esta disciplina, intrínsecamente formal, posibilita la integración de conceptos matemáticos y lógicos, adquiridos en años previos, como por ejemplo propiedad, demostración, inducción, etc. con conceptos específicos de las ciencias de la computación (algoritmo, ejecución, etc.).

Hoy en día, importantes herramientas utilizadas en la industria se basan en, o adoptan conceptos de, la programación funcional.

2. Propósitos y/u objetivos

Objetivos

Se espera que el/la estudiante:

- Adquiera y comprenda conceptos fundamentales de la programación funcional y su importancia en la tarea de programar. Por ejemplo: abstracción mediante funciones y sistemas de tipos.
- Sea capaz de aplicar técnicas de transformación de programas en casos particulares.
- Pueda implementar programas sencillos en un lenguaje funcional. Y pueda demostrar propiedades sencillas de programas funcionales utilizando inducción estructural.

3. Programa sintético:

Nociones generales del paradigma funcional. Valores y expresiones. Las funciones como valores. Sintaxis. Sistema de Tipos Hindley-Milner. Tipos básicos. Constructores de tipos. Polimorfismo. Funciones de alto orden. Currificación. Inducción/Recursión. Definición inductiva de conjuntos. Definición recursiva de funciones sobre esos conjuntos. Demostraciones inductivas. Inducción estructural. Listas como tipo inductivo. Funciones básicas y de alto orden sobre listas. Patrón de recorrido, selección y recursión. Sistemas de Tipos. Ventajas y limitaciones de los lenguajes de programación con tipos. Asignación de tipos a expresiones. Algoritmo de inferencia. Tipos de datos recursivos. Transformación de Programas. Obtención de programas a partir de especificaciones.

4. Programa analítico

4.1 Organización del contenido:

Unidad 1: Nociones generales del paradigma funcional

Valores y expresiones. Las funciones como valores. Mecanismos de definición de expresiones y valores. Ecuaciones orientadas para definir funciones. Sintaxis. Sistemas de tipos Hindley-Milner. Tipos básicos. Constructores de tipos. Polimorfismo. Sintaxis para valores de cada tipo (caracteres, tuplas, listas, strings, funciones). Funciones parciales y totales. Funciones de alto orden. Currificación.

Unidad 2: Inducción y recursión

Definición inductiva de conjuntos. Definición recursiva de funciones sobre conjuntos. Demostraciones inductivas sobre dichas funciones. Ejemplos: programas, expresiones aritméticas, listas.

Unidad 3: Listas

Listas como tipo inductivo. Funciones básicas sobre listas (append, head, tail, take, drop, reverse, sort, elem, etc.). Funciones de alto orden sobre listas. Patrón de recorrido: map. Patrón de selección: filter. Patrón de recursión: foldr. Listas por comprensión. Demostración de propiedades de listas y funciones sobre listas.

Unidad 3: Sistemas de tipos

Nociones básicas. Sistemas de tipado fuerte. Ventajas y limitaciones de los lenguajes de programación con tipos. Lenguaje de tipos. Asignación de tipos a expresiones. Propiedades interesantes de esta asignación. Algoritmo de inferencia. Mecanismos de definición de tipos nuevos y de funciones sobre ellos. Tipos algebraicos recursivos. Ejemplos: enumeraciones, listas, árboles binarios y generales.

Unidad 4: Transformación de Programas

Motivación. Obtención de programas a partir de especificaciones. Mejoramiento de eficiencia, con corrección por construcción. Técnicas particulares de transformación: tupling, eliminación de recursión, fusión. Transformación de listas por comprensión en expresiones que utilizan map, filter y concat.

Unidad 5: Lambda Cálculo

Definición del lenguaje. Sintaxis. Definición de sustitución. Modelo de computación. Nociones de alfa, beta y eta reducción. Semántica operacional. Lambda cálculo como modelo teórico de los lenguajes funcionales. Representación de booleanos, pares, números naturales, listas y otras construcciones.

4.1 Bibliografía y recursos obligatorios:

Bird, R. y Wadler, P. (1988). *Introduction to Functional Programming. First Edition.* Reino Unido, Oxford: Prentice Hall.

Bird, R. (1998). *Introduction to Functional Programming using Haskell. Second Edition* Reino Unido, Oxford: Prentice Hall.

A. Davie, A. (1992). *An Introduction to Functional Programming Systems using Haskell.* Reino Unido: Cambridge University Press.

Huday, P., Peterson, J. y Fasel, J. (2000). *A Gentle Introduction to Haskell. First Edition.* USA: Prentice Hall. Disponible en : <http://www.haskell.org/tutorial/>

4.2 Bibliografía optativa:

Lipovača, M. (2011). *Learn you a Haskell for great good. A Beginner's Guide.* Recuperado de: <http://learnyouahaskell.com> <http://learnyouahaskell.com/>

Thompson, S. (1996). *The Craft of Functional Programming.* USA: Addison Wesley.

Fejer, P. A., Simovici, D. (1991). *Mathematical Foundations of Computer Science. Volume I: Sets, Relations and Induction.* USA: Springer Verlag,

5. Metodologías de enseñanza:

Las clases se organizan como teórico-prácticas. Durante las clases se presenta el contenido formal y se ahonda en las características de la programación funcional.

En este sentido, en las primeras clases del curso se repasan los conceptos matemáticos de propiedad, demostración, prueba formal, inducción, en paralelo con la introducción de los conceptos y notaciones propios de la programación funcional. En ese contexto se discuten las similitudes y las diferencias del paradigma funcional respecto del procedural, y se introducen los conceptos característicos de la programación funcional, en particular, transparencia referencial, alto orden y sistemas de tipos.

Después de completado cada tema se resuelven ejercicios de programación haciendo hincapié en las características de la programación funcional mediante las computadoras del laboratorio.

En la práctica se plantean ejercicios que pueden ser realizados como trabajo experimental usando lenguajes funcionales (Haskell). A su vez, se hace uso de la plataforma Mumuki que promueve la educación de la programación y el pensamiento computacional de manera inclusiva, entretenida y durable.

Plan de trabajo en el campus:

El Campus Virtual es un espacio fundamental para el desarrollo de la asignatura. En el aula virtual se propondrá material educativo, apuntes de clase, bibliografía, así como también el programa y cronograma de la asignatura y las guías de Trabajos Prácticos y ejercicios.

Esto se complementa con el uso de Discord, que proporciona un espacio de actividades como foros de intercambio, entrega de actividades, socialización e intercambio de producciones entre estudiantes.

6. Actividades de investigación y extensión (si hubiera)

7. Evaluación y régimen de aprobación

7.1 Aprobación de la cursada

Consistirá en dos exámenes parciales con recuperatorios, según el cronograma previsto, de la totalidad de la materia descripta en el programa. Los mismos se realizarán en las fechas que establezcan en el cronograma correspondiente.

Para aprobar la cursada y obtener la condición de regular, el régimen académico establece que debe obtenerse una nota no inferior a cuatro (4) puntos y poseer una asistencia no inferior al 75% en las clases presenciales.

Podrán acceder al recuperatorio solo aquellos/as estudiantes que hayan obtenido una nota inferior o igual a 6 (seis) puntos en el examen parcial. La calificación que los/as estudiantes obtengan reemplazará la calificación obtenida en el examen que se ha recuperado y será la considerada definitiva a los efectos de la aprobación.

7.2 Aprobación de la materia

La materia puede aprobarse por promoción, evaluación integradora, examen final o libre.

Promoción directa: tal como lo establece el art°17 del Régimen Académico, para acceder a esta modalidad, el/la estudiante deberá aprobar la cursada de la materia con una nota no inferior a siete (7) puntos, no obteniendo en ninguna de las instancias de evaluación menos de seis (6) puntos, sean evaluaciones parciales ; recuperatorios o Trabajo práctico. El promedio estricto resultante deberá ser una nota igual o superior a siete (7) sin mediar ningún redondeo.

Evaluación integradora: tal como lo establece el art°18 del Régimen Académico, podrán acceder a esta evaluación aquellos/as estudiantes que hayan aprobado la cursada con una nota de entre cuatro (4) y seis (6) puntos.

La evaluación integradora tendrá lugar por única vez en el primer llamado a exámenes finales posterior al término de la cursada. Se realizará el mismo día y horario de la cursada y será administrado, preferentemente, por el/la docente a cargo de la comisión. Se aprobará tal instancia con una nota igual o superior a cuatro (4) puntos, significando la aprobación de la materia. La nota obtenida se promediará con la nota de la cursada.

Examen final: Instancia destinada a quienes opten por no rendir la evaluación integradora o hayan regularizado la materia en cuatrimestres anteriores. Son presenciales, individuales, sin material a la vista. La devolución de los resultados se realiza el mismo día en forma personalizada. Se evalúa la totalidad de los contenidos del programa de la materia y se aprueba con una calificación igual o superior a cuatro (4) puntos. Esta nota no se promedia con la cursada.

7.3 Criterios de calificación

La calificación de cada evaluación se determinará en la escala 0 a10, con los siguientes valores: 0, 1, 2 y 3: insuficientes; 4 y 5 regular; 6 y 7 bueno; 8 y 9 distinguido; 10 sobresaliente.

8. Cronograma

Semana	Tema	Contenidos	Modalidad
1	Introducción	Introducción al paradigma funcional y a Haskell	Presencial o virtual
2	Tipos	Introducción a los tipos de da-tos, algoritmo de inferencia de tipos	Presencial o virtual
3	Inducción/Recursión	Demostraciones por inducción y recursión en los naturales	Presencial o virtual
4	Inducción/Recursión	Demostraciones por inducción estructural y recursión en listas	Presencial o virtual

5	Mecanismo de reducción	Órdenes de evaluación, funciones totales y parciales	Presencial o virtual
6	Alto orden	Funciones como valores, funciones anónimas, composición	Presencial o virtual
7	Currificación	Funciones currificadas y descurrificadas, aplicación parcial	Presencial o virtual
8	Primer parcial		Presencial
9	Esquemas de recursión	Esquemas de recursión sobre listas: map, filter, fold	Presencial o virtual
10	Tipos algebraicos	Construcción de tipos algebraicos: enumerados, polimórficos, paramétricos, recursivos	Presencial o virtual
11	Esquemas de recursión	Esquemas de recursión sobre estructuras tipo árbol	Presencial o virtual
12	Transformación de programas	tupling, eliminación de recursión, fusión	Presencial o virtual
13	Lambda Cálculo	Definición del lenguaje, sustitución, Lambda cálculo como modelo teórico de los lenguajes funcionales.	Presencial o virtual
14	Segundo parcial		Presencial
15	Recupertorio primer parcial		Presencial
16	Recuperatorio segundo parcial		Presencial