

EXP-UNC 17547/2017

VISTO

El contenido de la Res. HCD N° 207/2002, donde se detalla la nómina de materias optativas para la Licenciatura en Ciencias de la Computación; y

CONSIDERANDO

Que la Coordinadora de la Comisión Asesora de Computación, Laura BRANDÁN BRIONES, propone incorporar la materia "Conceptos Avanzados de Lenguajes de Programación" a la nómina de materias optativas de la Licenciatura en Ciencias de la Computación;

Que mediante Res. HCS N° 122/2002 se ha delegado en este cuerpo la facultad de modificar la nómina de materias optativas del Plan de Estudios de la Licenciatura en Ciencias de la Computación;

Que se cuenta con el aval del Consejo de Grado.

Por ello,

EL CONSEJO DIRECTIVO
DE LA FACULTAD DE MATEMÁTICA, ASTRONOMÍA, FÍSICA Y COMPUTACIÓN

RESUELVE:

ARTÍCULO 1°: Modificar la nómina de materias optativas del Plan de Estudios de la Licenciatura en Ciencias de la Computación, incorporando a la misma la materia "Conceptos Avanzados de Lenguajes de Programación".

ARTÍCULO 2°: Fijar como programa, correlativas y carga horaria de la materia, los detallados en el Anexo que forma parte de esta Resolución.

ARTÍCULO 3°: Remitir a la Secretaría de Asuntos Académicos de la Universidad esta Resolución para su conocimiento y efectos, en cumplimiento con lo establecido en el Art. 2° de la Res. HCS N° 122/2002.

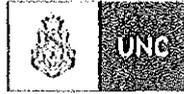
ARTÍCULO 4°: Notifíquese, publíquese y archívese.

DADA EN LA SALA DE SESIONES DEL CONSEJO DIRECTIVO DE LA FACULTAD DE MATEMÁTICA, ASTRONOMÍA, FÍSICA Y COMPUTACIÓN A VEINTICUATRO DIAS DEL MES DE ABRIL DE DOS MIL DIECISIETE.

RESOLUCIÓN CD N° 119/2017


Dra. SILVIA PATRICIA SILVETTI
SECRETARIA GENERAL
FaMAF


Dra. Ing. MIRTA IRIONDO
DECANA
FaMAF



Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía, Física y
Computación

EXP-UNC 17547/2017

RESO CD: 119/2017

ANEXO

PROGRAMA DE ASIGNATURA

ASIGNATURA: Conceptos Avanzados de Lenguajes de Programación	AÑO: 2017
CARACTER: Optativa	UBICACIÓN EN LA CARRERA: 5° año 2° cuatrimestre
CARRERA: Licenciatura en Ciencias de la Computación	
REGIMEN: Cuatrimestral	CARGA HORARIA: 120 Horas.

FUNDAMENTACIÓN Y OBJETIVOS

Los lenguajes de programación modernos vienen equipados con muchas capacidades y abstracciones para facilitar la construcción de programas. El objetivo de esta materia es exponer a las y los estudiantes a un conjunto variado de dichas capacidades, pudiendo de esta forma conocer con mayor profundidad y detalle el amplio espectro de lenguajes de programación existentes o por existir.

Puntualmente, se establecerán los siguientes ejes para conocer y comparar los lenguajes:

- **Paradigma:** Cada paradigma (funcional, imperativo, de objetos, concurrente, etc.) está diseñado con abstracciones específicas para trabajar sobre un cierto espacio de problemas.
- **Reutilización de código:** Es crucial entender las abstracciones que provee un lenguaje, más allá de aquellas provistas por el propio paradigma, para facilitar el reuso de código. Esto permite menos redundancia, y por ende mayor mantenimiento del código.
- **Tipado:** Los tipos codifican los valores que un programa recibe y retorna, y permiten, en mayor o menor medida, garantizar propiedades de seguridad, es decir, que un programa se comporte adecuadamente.
- **Codificado:** Algunos lenguajes son muy verbosíacos, haciendo que los programas sean largos y verbosos. Otros, en cambio, permiten escribir programas complejos de forma muy concisa, a veces incluso ocultando detalles importantes.

Desde λ Prolog hasta Rust hay una variedad de paradigmas. Desde los templates de C++ hasta los functors de OCaml hay todo un rango de métodos para reutilizar código. Desde Python hasta Idris hay toda una variedad de sistemas de tipos. Desde Java hasta Crystal existen muchas herramientas para escribir código conciso y claro.

La materia "Paradigmas de Programación" ya inicia parte del camino, enfocándose, como su nombre indica, en los paradigmas. El objetivo de esta materia es profundizar el mismo camino, y a la vez ver todos aquellos aspectos avanzados que incorporan los lenguajes, más allá de sus paradigmas.

Al finalizar la materia las y los estudiantes estarán en condiciones de evaluar qué lenguaje, o conjunto de lenguajes, le permitirá resolver de manera mas efectiva un desarrollo de software.

CONTENIDO

1. Repaso de los paradigmas imperativo y funcional desde una mirada unificadora. Valores por referencia vs. por copia. Tipado básico. Tipos abstractos de datos. Manejo de excepciones. Semántica de un lenguaje ilustrativo con características imperativas y funcionales.
2. Repaso del paradigma orientado a objetos como una extensión del imperativo. Registros y subtipado. Semántica de un lenguaje ilustrativo orientado a objetos. Tipos abstractos de datos vs. objetos.



EXP-UNC 17547/2017

RESO CD: 119/2017

3.
Inferencia de tipos.
4.
Sobrecarga de operadores como inferencia de funciones. Interfaces y definiciones.
5.
Insertando características imperativas en un lenguaje funcional puro. Mónadas. Transformadoras de mónadas.
6.
Modularización. Módulos en ML: estructuras, signaturas, funtores, módulos de primera clase. Módulos vs. clases.
7.
Tipos avanzados. Tipos refinados. Tipos dependientes.

BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

Types and Programming Languages. Benjamin C. Pierce.
Real World Haskell. Bryan O'Sullivan, Don Stewart, and John Goerzen.
Real World OCaml: Functional Programming for the Masses. Anil Madhavapeddy, Jason Hickey, and Yaron Minsky.
Practical Foundations for Programming Languages (2nd ed.). Robert Harper.

BIBLIOGRAFÍA COMPLEMENTARIA

Manuales de referencia, tutoriales, libros, y páginas web de los distintos lenguajes a ver.
Una posible lista de lenguajes incluye (orden alfabético):
λProlog, C++, Clojure, Crystal, F*, Haskell, Idris, Julia, OCaml, Ruby, Rust, Scala, Smalltalk

EVALUACIÓN

FORMAS DE EVALUACIÓN

FORMAS DE EVALUACIÓN

Las y los estudiantes deberán completar dos evaluaciones parciales, una de las cuales podrá ser recuperada. Además, deberán entregar cuatro trabajos de laboratorio que realizarán en grupo, pudiendo recuperar uno.

REGULARIDAD

La o el alumno deberá:

-aprobar al menos dos evaluaciones parciales o sus correspondientes recuperatorios, aprobar al menos el 60 % de los Trabajos Prácticos o de Laboratorio.

PROMOCIÓN

La o el alumno deberá:

-aprobar todas las evaluaciones parciales con una nota no menor a 6 (seis), y obteniendo un promedio no menor a 7 (siete),
-aprobar todos los Trabajos de Laboratorio con una nota no menor a 6 (seis), y obteniendo un promedio no menor a 7 (siete).

CORRELATIVIDADES

Para cursar:

-Paradigmas de lenguajes de programación (regularizada).

Para rendir:

-Paradigmas de lenguajes de programación (regularizada).