

**SEMINARIO DE LENGUAJES (OPCIÓN
C)**

Año 2021

Carrera/ Plan:

Licenciatura en Informática Plan 2003-07 / Plan 2012 / Plan 2015
Licenciatura en Sistemas Plan 2003-07 / Plan 2012 / Plan 2015
Analista Programador Universitario Plan 2007 / Plan 2015
Analista en TICs Plan 2017

Año: 2**Régimen de Cursada:** *Semestral***Carácter:** *optativa***Correlativas:** Algoritmos, datos y programas**Profesor/es:** Christian Rodriguez**Hs. semanales:** 6**FUNDAMENTACIÓN**

Dentro del marco de la carrera, esta opción introduce el lenguaje C, un lenguaje sencillo y a la vez poderoso, cuya historia se entrelaza con los sistemas operativos, redes y bases de datos. El aprendizaje del lenguaje C en esta instancia de la carrera contribuye favorablemente con la formación general del alumno, dado que lenguajes modernos heredan sus características, así como también es utilizado por otras materias avanzadas de la carrera como lenguaje de base.

OBJETIVOS GENERALES

Profundizar los conocimientos obtenidos por el alumno en los primeros cursos vinculados con Algoritmos y Programación, permitiéndole desarrollar un estudio teórico-práctico de algún lenguaje de programación (el lenguaje puede variar con los cambios tecnológicos), poniendo énfasis en el análisis formal de las características del lenguaje y su comparación con los que el alumno conociera a ese momento (típicamente Pascal).

RESULTADOS DE APRENDIZAJE

- 1.3 Describir los avances informáticos actuales e históricos y demostrar cierta visión sobre tendencias y avances futuros (Básico)
- 3.1. Definir y diseñar hardware/software informático/de red que cumpla con los requisitos establecidos (Básico)
- 3.3. Elegir y utilizar modelos de proceso adecuados, entornos de programación y técnicas de gestión de datos con respecto a proyectos que impliquen aplicaciones tradicionales, así como aplicaciones emergentes (Básico)
- 3.4. Describir y explicar el diseño de sistemas e interfaces para interacción persona-ordenador y ordenador-ordenador (Básico)
- 3.5. Aplicar las correspondientes competencias prácticas y de programación en la creación de programas informáticos y/u otros dispositivos informáticos (Adecuado)
- 6.1. Organizar su propio trabajo de manera independiente demostrando iniciativa y ejerciendo responsabilidad personal (Básico)
- 6.3. Planificar su propio proceso de aprendizaje autodidacta y mejorar su rendimiento personal como base de una formación y un desarrollo personal continuos (Básico)

CONTENIDOS MINIMOS

Estudio de un lenguaje de programación en el que se desarrollen aplicaciones concretas. En lo posible la oferta de lenguajes será variable y actualizada con el cambio tecnológico.

PROGRAMA ANALÍTICO

1. Características Generales
 1. Un poco de historia
 2. El proceso de la compilación y la ejecución de un programa en C.
2. Sintaxis Básica
 1. Estándares de codificación
 2. Tipos predefinidos
 3. Declaración de variables
 4. Estructuras de Control
 5. El formato de un programa de C
3. La Entrada y Salida estándar
4. El preprocesador
5. Definición de funciones
 1. Pasaje de parámetros en C
 2. Definición de prototipos
6. Manejo de expresiones
 1. Operadores
 2. Precedencia entre operadores
 3. Conversiones de tipo
7. Arreglos y Punteros
 1. Arreglos en C
 2. Manejo de punteros
 1. Introducción a las variables de tipo puntero
 2. Declaración de punteros
 3. Aritmética de punteros
 4. Punteros y parámetros
 5. Arreglos de punteros
 6. Funciones para manejo de memoria
8. Estructuras
 1. Definición y manipulación de estructuras
 2. Estructuras recursivas
9. Archivos
 1. Streams y descriptores
 2. Funciones de acceso en serie, bloque, directo y secuencial. Con y sin formato.
10. Modificación del alcance y tiempo de vida de variables
 1. Uso de extern y static
11. Punteros a funciones
12. Uso de librerías. Presentación de la libc.

BIBLIOGRAFÍA

- Kernighan, Ritchie (1991), El lenguaje de Programación C. 2° Edición. Prentice Hall. ISBN 0-13-110362-8.
- Reese, Richard (2013) Understanding and Using C Pointers. Ed. O'Reilly. ISBN 978-1449344184
- Perry (2000) C con ejemplos. Ed. Prentice Hall. ISBN: 9879460073
- Weiss. (1995) Efficient C Programming. A Practical Approach. Ed. Prentice Hall. ISBN 0-13-362658-X

METODOLOGÍA DE ENSEÑANZA

La metodología del presente año considera una adaptación por el contexto de pandemia en consecuencia del COVID-19. Si bien se adaptan cuestiones de virtualidad la metodología no varía sino la modalidad del dictado y los encuentros con los alumnos.

La metodología es de tipo taller por el carácter propio de la materia de enseñar un lenguaje. Por ello, las clases teóricas constan de aquellos aspectos conceptuales del lenguaje, donde se utilizan presentaciones del tipo slideshow online y muchos ejemplos prácticos compartiendo la pantalla de la terminal del docente.

La presentación utilizada en las teorías están disponibles online desde el comienzo de las clases y las mismas contienen ejemplos de código que pueden descargarse. Los ejemplos son completos, y funcionales, permitiendo al alumno resolver distintas situaciones problemáticas ilustrando conceptos complejos.

En la modalidad presencial, se utiliza un proyector en el aula, y en la modalidad virtual se comparte la pantalla del docente y además graba la sesión de la clase disponibilizando cada encuentro en la plataforma Moodle empleada por el curso. La modalidad virtual ofrece un registro en video de cada clase dictada y la visualización asincrónica de los alumnos.

Las clases prácticas por su parte, se realizan en la sala de PC de la facultad cuando la modalidad es presencial. En este caso, los estudiantes plantean sus dudas y trabajan con los ayudantes, quienes los acompañan en este proceso. El JTP es quien realiza explicaciones de práctica al inicio de cada trabajo práctico, haciendo hincapié en los ejercicios más importantes y puntos a evaluar. En total se deben completar 4 prácticas. Los trabajos prácticos incluyen ejercicios adicionales. Las prácticas incluyen evaluaciones de aprobación obligatoria, que se pueden recuperar al final de la cursada. Cada entrega incluye una instancia de coloquio donde el docente a cargo del grupo realiza distintas preguntas sobre la temática abordada. La entrega se realiza a través de la plataforma virtual.

En la modalidad virtual, las clases prácticas mantienen la misma metodología pero en vez de encontrarse en un aula, el encuentro se produce en la plataforma Moodle, donde los docentes son moderadores y los alumnos pueden consultar. La propia plataforma permite mantener chats privados para las consultas. El JTP siendo moderador, así como los ayudantes pueden exponer los temas de explicación práctica, y como sucede con las teorías cada clase deja un registro en video para su visualización asíncrona.

Dentro de la plataforma Moodle se incluyen autoevaluaciones de carácter optativo para reforzar temas y obligatorios. También se plantean entregas con fecha límite donde los alumnos pueden subir la versión finalizada de sus trabajos para la posterior corrección del pantele docente.

Cada ayudante tiene a cargo un grupo de alumnos, y será su función seguirlo en el desenvolvimiento de la cursada, intentado identificar los puntos problemáticos para poder resolverlos en forma rápida y no provoque el abandono de la cursada.

Las entregas que realizan los alumnos se complementan con coloquios entre el estudiante que expone la tarea realizada en forma individual y el docente. Es en esta instancia donde se evalúa no sólo el desarrollo, sino además los conocimientos, claridad de su entrega, organización y expresión. Esto se refleja en planillas que conforman documentación de evaluación del coloquio.

A fin de mejorar la comunicación entre los estudiantes y la cátedra, se utilizarán la plataforma con foros de notificaciones unidireccionales de la cátedra hacia los alumnos y por canales de consultas bidireccionales donde los alumnos realizan consultas y la cátedra responde. De esta forma la comunicación fluye de forma ágil y automática.

EVALUACIÓN

La ***aprobación de la cursada*** estará dada por la aprobación del 80% de los trabajos prácticos y el trabajo final integrador. Este trabajo final integrador tendrá la forma de “desafíos” que deben convertirse en “ideas proyecto” y posteriormente en potenciales desarrollos del alumno. Se trata de que alumno logre abstraer una serie de pasos que respondan a una metodología clásica de investigación y lo ayuden a formarse en esta competencia

- Búsqueda de bibliografía actualizada sobre el tema.
- Abstracción del desafío/problema como una “idea proyecto a resolver”.
- Expresión sintética de la especificación del proyecto, con recursos humanos requeridos y plan de tareas
- Implementación y defensa oral/escrita de la solución al desafío.

La cátedra acompaña el proceso del alumno, para consolidar sus habilidades para esta competencia. Además, se incluye material sobre los estándares del lenguaje y aplicaciones del lenguaje relacionadas con los cambios tecnológicos.

La evaluación de estas competencias se refleja en una planilla detallada, donde se indica la capacidad del alumno para desarrollar su aprendizaje y la formulación de la solución al desafío en forma autónoma.

En la cátedra se pone énfasis en la capacidad del alumno para conocer técnicas y herramientas de aplicación en Informática, en lo posible siguiendo las tendencias marcadas por el software libre, y en la aplicación efectiva de las mismas. La cátedra acompaña el proceso con materiales para que el alumno estudie casos, valore la selección y empleo eficiente de herramientas, y técnicas determinadas para cada problema. La evaluación de esta competencia forma parte de las evaluaciones de trabajos prácticos.

La ***aprobación de la materia*** estará dada por la aprobación de la cursada y un trabajo final.

CRONOGRAMA DE CLASES Y EVALUACIONES

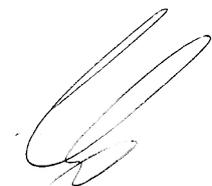
Clase	Fecha	Contenidos/Actividades
1	8/3	Historia de C. Características generales. Compilación de un programa. Tipos de datos. Variables. Operadores
2	15/3	Estructuras de control: bloques, condicionales simples y compuestos, iteraciones. Estructura de un programa. Funciones y prototipos.
3	22/3	Entrada y salida estándar.
4	29/3	El preprocesador: qué es, sus características. directivas.
5	5/4	El preprocesador: macros y condicionales. Punteros introducción.
6	12/4	Punteros: operadores y pasaje por referencia. Arreglos. Aritmética de punteros.
7	19/4	Strings. Asignación dinámica de memoria. Arreglos de punteros. Estructuras.
8	26/4	Definición de tipos. Estructuras dinámicas
9	3/5	Variables extern, static y register. Archivos. Funciones de archivos con streams.
10	10/5	Funciones de archivos con file descriptors
11	17/5	Punteros a funciones. Punteros a void
12	24/5	Argumentos variables
13	31/5	La glibc: una introducción acerca de cómo seguir aprendiendo C

Evaluaciones previstas	Fecha
Evaluación 1: contenidos hasta clase 7	26/4
Evaluación 2: contenidos hasta clase 13	7/6
Entrega 1 del trabajo integrador	21/6
Entrega 2 del trabajo integrador	12/7
Entrega final del trabajo integrador	2/8

Las fechas son aproximadas y pueden desfasarse por la extensión de algún tema pudiendo agregar más semanas para cubrir los contenidos mencionados

Contacto de la cátedra (mail, sitio WEB, plataforma virtual de gestión de cursos):

- Email del profesor: car@info.unlp.edu.ar
- Email de la cátedra: seminarioc@info.unlp.edu.ar
- Plataforma Moodle: <https://catedras.linti.unlp.edu.ar/>
- Teorías: http://catedra_c.gitlab.io/teoria/


Lic. Christian Rodriguez