



MORENO, 24 JUN 2014

VISTO el Expediente N° UNM:0000127/2014 del Registro de la UNIVERSIDAD NACIONAL DE MORENO; y

CONSIDERANDO:

Que el REGLAMENTO GENERAL ACADEMICO, aprobado por Resolución UNM-R N° 37/10 y sus modificatorias, el que fuera ratificado por el Acta de la Sesión Ordinaria N° 01/13 del CONSEJO SUPERIOR de fecha 25 de Junio de 2013, establece el procedimiento para la aprobación de las obligaciones curriculares que integran los Planes de Estudios de las carreras que dicta esta UNIVERSIDAD NACIONAL.

Que conforme lo dispuesto en el citado REGLAMENTO GENERAL, se ha elevado una propuesta de Programa de la asignatura: "ENTORNO C y JAVA" (2043 R), del ÁREA: REDES, correspondiente al CICLO SUPERIOR de la carrera INGENIERÍA EN ELECTRÓNICA del DEPARTAMENTO DE CIENCIAS APLICADAS Y TECNOLOGÍA, de esta UNIVERSIDAD, aconsejando su aprobación con vigencia a partir del 2do. Cuatrimestre del Ciclo Lectivo 2014.

Que la SECRETARÍA ACADÉMICA de la UNIVERSIDAD ha

*Amey*

emitido opinión favorable, de conformidad con lo previsto en el artículo 3º de la Parte I del citado REGLAMENTO GENERAL, por cuanto dicho Programa se ajusta a las definiciones enunciadas en el artículo 4º de la Parte I del REGLAMENTO en cuestión, así como también, respecto de las demás disposiciones reglamentarias previstas en el mismo.

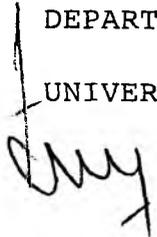
Que la SUBSECRETARÍA LEGAL Y TÉCNICA ha tomado la intervención de su competencia.

Que el CONSEJO del DEPARTAMENTO DE CIENCIAS APLICADAS Y TECNOLOGÍA, en sesión de fecha 6 de marzo de 2014, trató y aprobó el Programa propuesto, conforme lo establecido en el artículo 2º de la Parte I del REGLAMENTO GENERAL ACADÉMICO.

Por ello,

EL DIRECTOR GENERAL DEL DEPARTAMENTO DE CIENCIAS APLICADAS Y  
TECNOLOGÍA DISPONE:

ARTÍCULO 1º.- Aprobar el Programa de la asignatura: "ENTORNO C y JAVA" (2043 R), del ÁREA: REDES, correspondiente al CICLO SUPERIOR de la carrera INGENIERÍA EN ELECTRÓNICA del DEPARTAMENTO DE CIENCIAS APLICADAS Y TECNOLOGÍA, de esta UNIVERSIDAD, con vigencia a partir del 2do. Cuatrimestre del





03

~~ES COPIA~~  
UNIVERSIDAD NACIONAL DE MORENO

**Universidad Nacional de Moreno**  
**Departamento de Ciencias Aplicadas y Tecnología**

Lic. MARISA AMENEIRO  
JEFA DE DEPARTAMENTO  
MESA DE ENTRADAS  
UNIVERSIDAD NACIONAL DE MORENO

Ciclo Lectivo 2014, el que como Anexo I forma parte integrante de la presente Disposición.

ARTÍCULO 2º.- Regístrese, comuníquese, dese a la SECRETARÍA ACADÉMICA a sus efectos y archívese.-

DISPOSICIÓN UNM-DCAyT N° **03-14**

*Uney*

  
Mg. JORGE L. ETCARRÁN  
DIRECTOR GENERAL DEPARTAMENTO DE  
CIENCIAS APLICADAS y TECNOLOGÍA  
UNIVERSIDAD NACIONAL DE MORENO

ANEXO I

**Asignatura: ENTORNO C y JAVA (2043 R)**

**Carrera:** INGENIERÍA EN ELECTRÓNICA (Plan de estudios aprobado por Resolución UNM-R N° 21/10 -orientación en REDES-) y su modificatoria UNM-R N° 407/11)<sup>1</sup>

**Área:** Redes

**Trayecto curricular:** Ciclo Superior

**Periodo:** 7° y 8° Cuatrimestre - Año 4

**Carga horaria:** 96 (noventa y seis) horas

**Vigencia:** A partir del 2° Cuatrimestre 2014

**Clases:** 32 clases (treinta y dos)

**Régimen:** de regularidad o libre

**Responsable de la asignatura:** RODRIGUEZ FERNANDO GABRIEL

**Programa elaborado por:** Fernando Gabriel RODRIGUEZ, Marcelo Roberto TASSARA y Daniel Emilio RIGANTI

**FUNDAMENTACIÓN DEL PROGRAMA:**

La materia, junto con su predecesora forman un bloque que da las herramientas cognitivas para poder diseñar y programar los dispositivos que utilizan tecnologías orientadas a objetos. Como son las nuevas familias de microcontroladores y los dispositivos Android, entre otros.

**OBJETIVOS GENERALES:**

- Formar al alumno en el conocimiento del lenguaje orientado a Objetos C++.
- Formar al alumno en el conocimiento del lenguaje JAVA.
- Desarrollar actividades prácticas para aplicar los conocimientos adquiridos.

**CONTENIDOS MÍNIMOS:**

El lenguaje C++ y el framework de .net. Creación de Programas Simples en C++. Programación orientada a objetos. Clases y Objetos. Herencia y polimorfismo. Constructores: definición, llamada, constructor por defecto. Creación de objetos (operador new). Operaciones de conversión (explícita, uso de as). Definir y Utilizar una Clase. Hacer Override a un Método. Introducción a Java. Herencia y métodos virtuales. Concepto de herencia. Clase base, clase hija, sintaxis de clases derivadas. Polimorfismo. Métodos genéricos. Operador is. Arreglos y Colecciones. Clase

<sup>1</sup> Se encuentra autorizado por Resol. 2287/13 y 2288/13 del MINISTERIO DE EDUCACIÓN

*[Handwritten signature]*

StringBuilder. Arrays y Serialización. Arrays unidimensionales y multidimensionales y anidados. Serialización XML. Vectores. Matrices. Alcance de objetos y Reciclado de Memoria. Control de Acceso. Variables y Métodos. This y super. Métodos Getters y Setters. Argumentos y valores de retorno. Ámbito. Clase Abstractas. Interfaces. Arrays de datos primitivos y de objetos. Serialización de Objetos. Construcción de GUI en Java. Generación de interfaces de usuario con NetBeans. AWT (Abstract Windows Toolkit). SQL. Hilos. Procesos y hebras. Aplets.

## PROGRAMA

### UNIDAD 1:

Programación Orientada a Objetos. Clases y Objetos. Herencia. Polimorfismo. Encapsulación. Clases. Tipos de Clases. Variables y Métodos de Instancia. Ámbito de una variable. Métodos. Argumentos y valores de retorno. Métodos Constructores. Sobrecarga y sobrescritura de métodos. Alcance de Objetos y Reciclado de Memoria. Herencia. Control de Acceso. Variables y Métodos. this y súper. Métodos Getters y Setters. Argumentos y valores de retorno. Ámbito. Clase Abstractas. Interfaces.

### UNIDAD 2:

Arrays. Arrays de datos primitivos y de objetos. Listas, Ordenar listas, Criterios de Ordenamiento. Serialización de Objetos. Excepciones y aserciones

### UNIDAD 3:

El lenguaje C++ y el framework de .net. Common Language Runtime (CLR). Biblioteca de clases de .net. Fundamentos del lenguaje C#. Identificadores: consideraciones al definirlos. Palabras reservadas. Tipos de datos. Literales. Variables. Constantes. Métodos o funciones. Sintaxis de la llamada a las funciones definidas. La función principal main() y los programas en C#. Expresiones aritméticas. Expresiones relacionales, lógicas y booleanas. Operadores. Conversión de tipos (Type casting) Entrada y salida. Salida con formato: Console.WriteLine. Entrada con formato: Console.ReadLine. Creación de Programas Simples en C++. Crear una Aplicación simple de Consola con C#. Utilizar un Namespace. Instrucciones que Controlan el flujo del Programa. Instrucciones. Concepto. Bloques de instrucciones. Instrucciones básicas: definición de variables locales, asignaciones, llamadas a métodos, instrucciones

Amey



03

COPIA  
UNIVERSIDAD NACIONAL DE MORENO

Lic. MARISA AMENEIRO  
JEFA DE DEPARTAMENTO  
MESA DE ENTRADAS  
UNIVERSIDAD NACIONAL DE MORENO

**Universidad Nacional de Moreno**  
**Departamento de Ciencias Aplicadas y Tecnología**

condicionales (instrucción if, instrucción switch), instrucciones iterativas (instrucción while, instrucción do while, instrucción for).

**UNIDAD 4:**

Programación orientada a objetos. Clases y Objetos. Principios de la Orientación a Objetos. Conceptos del Diseño Orientado a Objetos - herencia y polimorfismo. Declaración de clases y sus elementos: visibilidad y accesibilidad de una clase y sus miembros. Concepto de Clases. Elementos de una clase: campos, Atributos y Operaciones (métodos). Definición de clase: concepto de clase y objeto. Sintaxis de definición de clases. Miembros de una clase. Acceso a los campos. Sintaxis de métodos. Sintaxis para llamar a los métodos. Implementación de una Clase. Clase System.Object. Métodos: concepto, definición, llamada, tipo de parámetros (entrada y salida). Métodos externos (modificador extern). Constructores: definición, llamada, constructor por defecto. Propiedades. Concepto, definición, acceso, implementación (get y set). Ejercicios. Enumeraciones. Concepto, definición, uso. Clase System. Enum.

**UNIDAD 5:**

Operaciones de creación de objetos. Creación de objetos (operador new). Operaciones de conversión (explícita, uso de as). Constructores. Constructor por defecto. Referenciar al objeto actual: uso de this. Sobrecarga de métodos. Trabajando con Clases. Definir y Utilizar una Clase. Utilizar los Modificadores de Acceso: Definir NameSpaces. Declarar una Clase Abstracta. Derivar una Clase Concreta de una Clase Abstracta. Utilizar una Clase Abstracta y su Clase Derivada Concreta. Trabajando con Métodos Implementar Constructores. Hacer Override a un Método. Sobrecargar un Método. Herencia. Encapsulamiento. Introducción a Java. Origen de Java. Características de Java (Simple - Orientado a Objetos - Distribuido - Robusto - De Arquitectura Neutral - Seguro - Portable - Interpretado). Conceptos Básicos de Java. Programación en Java. Comentarios Identificadores. Palabras clave. Palabras reservadas. Variables, Constantes, Tipos de Datos, operadores. Literales. Estructuras de control, condicionales, Bucles

**UNIDAD 6:**

Herencia y métodos virtuales. Concepto de herencia. Clase base, clase hija, sintaxis de clases derivadas. Constructores de clases hijas. Modificadores de acceso (public, private, static, protected, internal). Polimorfismo. Métodos genéricos. Operador

*[Handwritten signature]*



03

COPIA  
UNIVERSIDAD NACIONAL DE MORENO  
Lic. MARISA AMENEIRO  
JEFA DE DEPARTAMENTO  
MESA DE ENTRADAS  
UNIVERSIDAD NACIONAL DE MORENO

**Universidad Nacional de Moreno**  
**Departamento de Ciencias Aplicadas y Tecnología**

is. Arreglos y Colecciones. Clase StringBuilder. Tratamiento de array: Clase System.Array (miembros). Tratamiento de string: System.string (miembros). Arrays y Serialización. Arrays unidimensionales y multidimensionales y anidados - Colecciones. Diccionarios. Generics (aplicados a colecciones). Serialización XML. Vectores. Matrices. Trabajando con Arreglos y Colecciones. Crear un Tipo Colección: Utilizar un Tipo Colección. Funciones. Listas. Instrucción for, instrucción foreach.

**UNIDAD 7:**

Construcción de GUI en Java. El manejo de eventos en GUI. Generación de interfaces de usuario con NetBeans: adición de contenedores, adición, alineación y anclaje de componentes individuales y de varios componentes. Aplicaciones basadas en GUI. AWT (Abstract Windows Toolkit). Acceso a Bases de Datos (JDBC). Cargar controlador. Realizar conexión. Sentencias SQL. Hilos. Procesos y hebras. Concurrencia y sincronización. Aplets.

**BIBLIOGRAFÍA:**

Se le brindará al alumno los materiales necesarios para que puedan desarrollar los temas utilizando métodos de búsqueda en internet. La gran cantidad de información gratuita disponible en la red permitirá al alumno desarrollar los contenidos..

**Bibliografía Obligatoria:**

Como Programar En C/C++ Y Java

Autor: DEITEL HARVEY M., DEITEL PAUL J.

Editorial: PEARSON ADDISON-WESLEY

1152 paginas, 1970 gramos, encuadernación rústica.

Edición: 2004 | Idioma: Castellano

ISBN: 9789702605317

Como Programar C#

Autor: DEITEL HARVEY M., DEITEL PAUL J.

Editorial: PEARSON ADDISON-WESLEY

1080 paginas, 1870 gramos, encuadernación rústica.

Edición: 2007 | Idioma: Castellano

ISBN: 9789702610564

<https://developers.google.com/edu>

**Bibliografía complementaria**

El lenguaje de programación C++. Autor: Bjarne Stroustrup.

Pensando en C++. Autor: Bruce Eckel.

Pensando en Java. Autor: Bruce Eckel.

*Handwritten signature*



03



**Universidad Nacional de Moreno**  
**Departamento de Ciencias Aplicadas y Tecnología**

**OBJETIVOS DE APRENDIZAJE:**

Al finalizar el curso, el alumno deberá comprender el paradigma de Orientación a Objetos y poseer las bases para desarrollar en cualquier lenguaje orientado a objetos. Obtendrá bases específicas para desarrollar tanto en C++ como en Java y C#.

**OBJETIVOS PEDAGÓGICOS:**

Comprender el funcionamiento de circuitos amplificadores realimentados y fuentes de alimentación.

- Diseñar y desarrollar aplicaciones utilizando el paradigma de orientación a objetos.
- Poseer criterio para seleccionar el lenguaje de programación adecuado según la aplicación.
- Comprenda que el rol del ingeniero es contribuir al desarrollo y mejora de la calidad de vida de la comunidad en la que desarrolla su actividad.
- Integre los conocimientos adquiridos contemporánea y previamente al curso.
- Genere nuevos conocimientos a partir de su propia experiencia.
- Tome contacto con dispositivos que requieran del desarrollo de aplicaciones.
- Aprenda la importancia de autocapacitación utilizando las herramientas gratuitas disponibles. En especial con lo efímero de los conocimientos tecnológicos actuales.
- Incorpore el soporte de herramientas informáticas y software de simulación.

**METODOLOGÍA DE TRABAJO:**

La asignatura está constituida por 7 Unidades las cuales se dictarán durante dos cuatrimestres en clases teóricas y prácticas, con resolución de problemas a cargo de los alumnos.

Las guías de trabajos prácticos, propuestos por el docente, se resolverán en forma individual. Se utilizarán las herramientas informáticas adecuadas para la resolución de los problemas y la justificación de las respuestas obtenidas.

**EVALUACIÓN Y APROBACIÓN:**

Evaluación:

La evaluación consta de dos exámenes parciales y un examen final. Los exámenes parciales se aprobarán con una nota mínima de cuatro (4), y una vez comprobada y aprobada la entrega de los TP's el

*[Handwritten signature]*



alumno tendrá derecho a rendir el examen final que se aprobará con un mínimo de cuatro (4).

El alumno podrá "recuperar" sus exámenes parciales en 3 (tres) fechas destinadas a tal efecto. Cada parcial podrá ser recuperado un máximo de 2 (dos) veces. Asimismo el alumno podrá rendir el examen final en 3 (tres) fechas, no consecutivas, destinadas a tal efecto.

**RÉGIMEN DE APROBACIÓN:**

- Asistencia mínima del 80% (ochenta por ciento)
- Regularización y examen final: Aprobación de las dos instancias de evaluación con mínimo de 4 (cuatro) puntos.
- Asistencia menor al 80% (ochenta por ciento), en este caso el alumno deberá recuperar la totalidad de sus exámenes parciales.
- El alumno deberá aprobar los TP's de la cátedra.
- La asignatura podrá ser "promocionada" en el caso que los exámenes parciales tengan nota 7 (siete) como mínimo, cada uno. No promociona el alumno que tenga notas menores a 7 en cada uno de los parciales. No se promediarán las notas de los parciales para lograr la promoción. El régimen de promoción hace que el alumno, habiendo cumplido los requisitos anteriormente mencionados, no tenga que rendir examen final para aprobar la asignatura.