

CÓDIGO DE ASIGNATURA

1024

ASIGNATURA: ELEMENTOS DE PROGRAMACION

JEFE DE CÁTEDRA: Lic. Edgardo Javier Moreno

AÑO: 2016

CARGA HORARIA: 8

OBJETIVOS:

La asignatura Elementos de Programación forma parte del primer año de la currícula aprobada, está planteada dentro del plan general de estudios como una materia básica. Tiene un objetivo general que es el de introducir al alumno en el difícil arte de la programación, lo cual significa aprender a resolver problemas, planteando situaciones originales, utilizando las habilidades y ventajas de un computador. El mismo se desarrolla a través de tres sub-objetivos básicos perfectamente diferenciados y complementarios entre sí para el logro del objetivo.

- a) Aprender a interpretar, entender y encontrar los algoritmos necesarios para resolver los problemas propuestos, en distintos enunciados debidamente preparados para que el alumno, razone y adquiera nuevos hábitos de investigación para obtener los resultados correctos.
- b) Aprender a desarrollar lógicamente los procesos interpretados como correctos para resolución de problemas, con medios gráficos, como diagramas lógicos que permitan mostrar visualmente la solución a los problemas planteados. Verificar la lógica diseñada con lotes de prueba, asegurando el correcto funcionamiento de los problemas que pueden ser ejecutados a posteriori en un computador digital.
- c) Aprender a codificar, utilizando un adecuado lenguaje de programación como es el C, los problemas lógicamente desarrollados. Es este un lenguaje de aplicación en diversas áreas de la ingeniería.

Cumplidos estos sub-objetivos, podemos afirmar que aquellos alumnos que lo aprueben, estarán en condiciones de poder confeccionar un programa del tipo ó similar al que pueda controlar la gestión administrativa por ejemplo la de un videoclub, un almacén, un deposito, etc., no será el óptimo, por la limitación de los recursos con que disponemos en esta etapa de la carrera, pero puede funcionar muy bien.. Poseerá además una adecuada base para la comprensión de las nuevas tecnologías existentes en el mundo de la programación. Los conocimientos adquiridos serán afianzados y complementados con las otras dos materias de Programación a realizar en los años siguientes, los cuales introducirán a los alumnos en la Programación Orientada a los Objetos y en la programación Tipo Visual. Merece destacarse como otro objetivo adicional, durante el desarrollo de la materia, la búsqueda de la integración de la teoría con la práctica. El alumno será inducido a la búsqueda bibliográfica, aprender a trabajar en equipo y saber documentar lo realizado.

CONTENIDOS MÍNIMOS:

El papel de la computadora en la actualidad. Que es un Programa. Lenguajes de Programación. Etapas para solución de problemas. Diseño de estrategias. Algoritmos. Confección de Programas. Estructuras Básicas. Simbología utilizada para representar el proceso lógico de un programa. Selección Simple, Selección Múltiple, Proceso repetitivo Fijo, Proceso repetitivo Condicional. Resolución de Problemas. Lenguaje C. Generalidades. Elementos del Lenguaje. Codificación, programa fuente, Compilación, Objeto. Declaración de Variables, Distintas características, Tamaño según su contenido. Funciones de Entrada / Salida Sentencias de Asignación. Funciones de Librería. Estructura de Selección Simple/Múltiple. Operadores Lógicos. Variables. Contador/Acumulador/Bandera. Máximos. Mínimos. Funciones. Prototipos. Desarrollo, Argumentos. Arreglos Unidimensionales. Búsquedas. Arreglos Bidimensionales. Uso de Matrices. Estructuras. Datos estructuras. Archivos. Creación, Grabación, Consulta. Corte de Control. Ejercicios de Aplicación para resolver problemas.

PROGRAMA ANALÍTICO:

UNIDAD I – INTRODUCCIÓN GENERAL A LA PROGRAMACION

Objetivo: Introducir al alumno en la programación de las computadoras.

- Descripción general del principio de funcionamiento de un computador digital.
- Unidades componentes y sus características. Hardware y software.
- Sistemas de numeración binario y hexadecimal.
- Concepto de programa. Ejemplos de aplicación y ejecución.
- PRÁCTICA:* Desarrollo de un sencillo programa. Sistemas de numeración.

Aprendizaje a lograr: Concepto de programa y sus objetivos. Identificar las unidades del computador a usar en la programación.: Memorias y almacenamiento de programas.

UNIDAD II - INTERPRETACION DE PROBLEMAS – DIAGRAMACION ESTRUCTURADA - ESTRATEGIAS

Objetivo: Aprender a efectuar un análisis de los problemas y elaborar la estrategia más adecuada para su resolución. Poder diferenciar los distintos tipos de problemas

- Compresión de problemas. Enunciados e interpretación. Análisis de ideas.
Relación entre datos, incógnitas y resultados
- Metodología a desarrollar para su resolución. Diseño de estrategias.
- Análisis de los diversos tipos de problemas. Algoritmos, concepto y definición. Características, limitaciones. Análisis del problema y diseño del algoritmo. Diseño descendente top-down.
- Definición y aplicación de los diagramas de lógica. Simbología.
- Teorema fundamental de la Programación Estructurada. Estructuras básicas.
Combinación de las distintas estructuras. Precauciones.

f. Programación modular. Técnicas de construcción y seguimiento de diagramas. Prueba y depuración de algoritmos. Juegos de prueba
g. *PRÁCTICA*: Confección de algoritmos. Juegos de prueba. Análisis de resultados. Confeccionar los diagramas de lógica de sencillos problemas de aplicación utilizando las estructuras básicas. Juego de prueba y análisis de resultados.

Aprendizaje a lograr: Confeccionar los diagramas de lógica de sencillos problemas de aplicación utilizando las estructuras básicas. Juego de prueba y análisis de resultados.

UNIDAD III – ELEMENTOS BÁSICOS DEL LENGUAJE “C”

Objetivo: Conocer los elementos básicos de un lenguaje de programación.

Enseñar a confeccionar sencillos programas, codificando diagramas. Utilización de las sentencias básicas en la resolución de nuevos problemas.

- a. El lenguaje “C”, breve historia y alcances. El entorno de la programación.
- b. Estructura general de un programa. El preprocesador, la función del main().
- c. Elementos básicos del lenguaje. Alfabeto. Palabras reservadas. Tipos de datos. Constantes, variables y operadores. Código ASCII.. Expresiones, diferentes tipos.
- d. Sentencias de asignación. Sintaxis, reglas y prioridades.
- e. Sentencias de entrada y salida con formato. Funciones printf(...) y scanf(..).
- f. Codificación de diagramas desarrollados como ejemplos.
- g. Confección de sencillos programas. Juegos de prueba. Depuración de programas.
- h. *PRÁCTICA*: Codificación en C de los diagramas de lógica. Edición, compilación y ejecución del programa. Prueba y depuración. Programación interactiva.

Aprendizaje a lograr: Estar capacitado para desarrollar sencillos programas en lenguaje C utilizando las sentencias básicas del lenguaje. Corrección de errores.

UNIDAD IV – ESTRUCTURAS DE CONCATENACION

Objetivo: Conocer y aplicar los conceptos de contador y acumulador. Conocer y aplicar las Funciones de Biblioteca ó Standard del "C". Aplicación y alcance de las estructuras de concatenación.

- a. Conceptos de contador y acumulador. Autoincremento.
 - b. Funciones standard del "C" , diversos tipos.
 - c. Combinación de operadores.
 - d. Casos típicos de error en ciertas operaciones. Errores de almacenamiento.
 - e. Sentencias de comentario. Estilo en la programación.
 - f. *PRÁCTICA:* Confección de programas utilizando las funciones de biblioteca.
- Pruebas de exceso de almacenamiento. Errores posibles en ciertas funciones.

Aprendizaje a lograr: Estar capacitado para desarrollar programas en lenguaje C utilizando las funciones de biblioteca. Estudio y corrección de errores.

UNIDAD V – ESTRUCTURAS DE SELECCIÓN SIMPLE Y MULTIPLE

Objetivo: Conocer y aplicar las estructuras de selección simple y de selección múltiple. Poder aplicar estas estructuras y sus combinaciones en la solución de los problemas y su posterior codificación en lenguaje C. Conocer y aplicar los operadores lógicos.

- a. Sentencia de selección simple, if(...). Sentencias anidadas. Operador condicional.
 - b. Casos particulares en la combinación de las estructuras.
 - c. Operadores lógicos. Prioridades y combinación. Reducción de los programas.
 - d. Estructura de selección múltiple switch(...). Utilización de caracteres.
 - e. Sentencia break., importancia de su utilización.
 - f. Confección de programas. Juegos de prueba. Comprobación de resultados.
 - g. *PRÁCTICA:* Confección de programas utilizando las sentencias mencionadas.
- Codificación. Confección de completos juegos de prueba. Pruebas de escritorio.
- Comprobación practica de los casos mencionados precedentemente.

Aprendizaje a lograr: Estar capacitado para desarrollar programas en lenguaje C utilizando las estructuras de selección simple y múltiple, con todas sus variantes. Seguimiento de programas con los juegos de prueba. Análisis de variantes.

UNIDAD VI – ESTRUCTURAS DE ITERACION

Objetivo: Adquirir la capacidad para aplicar en la solución de los problemas las estructuras iterativas, en sus variantes. Poder programar algoritmos de búsqueda de máximos y mínimos. Aprender el uso de la pantalla graficando caracteres.

- El concepto de ciclo. La sentencia de iteración definida `for(...)`
- Variantes en la utilización de los parámetros del `for(...)`. Casos particulares.
- Ciclos anidados. Poder de iteración y sus variantes.
- Sentencia de iteración condicionada. Sentencias `while(...)` y `do while(...)`. Comparación entre ambas. Ciclos 0-n y 1-n.
- Estructuras de iteración combinadas, sus ventajas.
- Sentencias `"continue"` . Sentencia `"exit"` .
- Graficación en la pantalla utilizando caracteres. Sentencias `gotoxy(...)` y `windows(...)`. Otras funciones que facilitan la operatividad de las ventanas.
- PRÁCTICA:** Confección de programas utilizando las sentencias de iteración. Prueba de programas con utilización práctica de ventanas. Facilitar el diálogo con los usuarios de los programas. Pruebas de aplicación directa.

Aprendizaje a lograr: Estar capacitado para desarrollar programas en lenguaje C utilizando las estructuras iterativas en todas sus variantes. Manejo sencillo de las ventanas. Ventajas. Seguimiento de programas con los juegos de prueba.

UNIDAD VII – PROGRAMACIÓN MODULAR – FUNCIONES

Objetivo: Aprender a dividir los programas en partes menores, que llamaremos "funciones". Adquirir la capacidad para poder construir

programas mayores en base a la construcción y utilización de funciones. Funciones pre-programadas.

- a. Los subprogramas. Las funciones en C. Forma general
- b. Declaración, prototipos y definición. Estructura y codificación.
- c. Variables globales y locales. Llamadas a una función.
- d. Declaración de parámetros. Parámetros formales y reales. Sentencia return.
- e. Llamada a funciones, argumentos. Parámetros por valor y por variable.
- f. Diagramas sintácticos.
- g. Los punteros, definición y utilización como parámetros de funciones.
- h. Reglas de ámbito. Variables externas. Algunas funciones standard Concepto de recursividad. Aplicaciones.
- i. *PRÁCTICA*: Confección de funciones con y sin parámetros. Inclusión de funciones pre programadas. Aplicación en el diseño y modularización de los programas. Usar funciones recursivas.

Aprendizaje a lograr: Capacitación necesaria para desarrollar programas en lenguaje C definiendo, confeccionando y utilizando funciones con diversos tipos de parámetros. Utilizar funciones pre programadas.

UNIDAD VIII – DATOS ESTRUCTURADOS – LOS ARRAYS

Objetivo: Aprender a utilizar en los programas datos con estructura como son los vectores y las matrices. Familiarizarse con la utilización de tablas, búsquedas y ordenamiento. Comprender la ventaja de su utilización, cadena de caracteres y funciones adecuadas para procesamiento de caracteres.

- a. Conceptos básicos sobre estructuras de datos tipo array. Declaración.
- b. Arrays unidimensionales, los vectores. Inicialización. Comprobación de límites.
- c. Arrays multidimensionales, tablas y matrices.
- d. Los arrays como parámetros de funciones.
- e. Operaciones de ordenamiento y búsqueda en arrays. Distintos métodos.
- f. Representación de los strings por cadenas de caracteres, conceptos.
- g. Funciones standard para manejo de cadenas : strlen(...), strcpy(...), strcat(...), strcmp(...). Uso de null

h. **PRÁCTICA:** Confección de programas que ingresan o exhiben datos con estructura tipo arrays. Aplicación de los algoritmos de ordenamiento y búsqueda. Creación de funciones que reciben arrays, su utilización, funciones con cadenas.

Aprendizaje a lograr: Capacitación necesaria para desarrollar programas que utilizan datos estructurados. Práctica en la utilización de los algoritmos básicos que usan arrays. Utilización de funciones que operan sobre arrays.

UNIDAD IX – LAS ESTRUCTURAS DE DATOS

Objetivo: Aprender a definir y utilizar las estructuras de datos en forma similar a los arrays, salvando sus diferencias.

- a. Definición de las estructuras como nuevo tipo de datos. Campos.
- b. Procesamiento. Tipos de datos definidos por el usuario. struct (...)
- c. Longitud de las estructuras. Estructuras anidadas. Procesamiento.
- d. Las estructuras como parámetros de funciones.
- e. Arrays de estructuras. Concepto de uniones. Almacenamientos temporarios.
- f. **PRÁCTICA:** Confección de programas y funciones que definen y utilizan estructuras. Generación de arrays de estructuras y su operatividad.

Aprendizaje a lograr: Capacitación necesaria para desarrollar programas que utilizan estructuras de datos. Aprender a utilizar las estructuras como almacenamiento temporario de datos. Utilización en las funciones.

UNIDAD X – LOS ARCHIVOS

Objetivo: Aprender a definir y utilizar los archivos de datos como medio de almacenamiento permanente de la información. Conocer los diferentes tipos.

Aprender a guardar información y recuperarla. Métodos diversos.

- a. Conceptos generales sobre archivos en "C". Tipos permitidos, de texto y binarios
- b. Definición, creación, lectura y grabación. en archivos binarios secuenciales.

- c. Funciones standard para su operatoria. Generación del nombre de los archivos.
- d. Archivos como parámetros de funciones.
- e. Métodos de búsqueda de información en un archivo binario.
- f. *PRÁCTICA*: Confección de programas y funciones que generan y utilizan archivos binarios. Recupero de información. Tamaño y límite de los archivos.

Aprendizaje a lograr: Capacitación necesaria para desarrollar programas que utilizan búsqueda archivos binarios, utilizando diversos métodos de generación y de la información. Identificación de los archivos dentro del computador. Directorios.

UNIDAD XI – USO DE ARCHIVOS – CORTE DE CONTROL

Objetivo: Aprender manejo integral de información para su correcto almacenamiento. Agregar Borrar, Modificar registros. Corte de Control.

- a. Comandos relacionados con archivos.
- b. Búsqueda de datos y relación de campos.
- c. Corte de Control, Niveles de Cortes.
- d. Características de la Información.
- e. *PRACTICA*: Confección de programas y practica de manejo de archivos utilizando Corte de Control.

Aprendizaje a lograr: Apertura y almacenamiento de archivos, presentar informas y acumuladores con use de corte de control.

BIBLIOGRAFÍA:

BIBLIOGRAFÍA BÁSICA

(Debe existir en Biblioteca)

Autor	Título	Editorial	Año	Edición
Guillermo Correo Uribe	Desarrollo de Algoritmos y sus Aplicaciones	McGraw-Hill		Tercera
B.Kernighan/D.Ritchie	EL LENGUAJE DE PROGR. C++	Prentice Hall.		
Docentes	Apuntes de la Cátedra	Fotocopia		

BIBLIOGRAFÍA COMPLEMENTARIA

Autor	Título	Editorial	Año	Edición
Claudio F. Freijedo. Alicia Cortagerena	Tecnologías de la Información y las comunicaciones.	Ediciones Macchi		
Byron S. Gottfried	Programación en C	McGraw-Hill.		

METODOLOGÍA DE ENSEÑANZA:

El alumno será inducido a: generar métodos de análisis para interpretar y resolver problemas propuestos, aprender a trabajar en equipo, saber evaluar las distintas alternativas algorítmicas para generar procesos que satisfagan problemáticas distintas, realizar una correcta presentación gráfica, a través de diagramas de lógicas que representen desarrollos y procesos determinados. Las clases tendrán básicamente contenido teórico / práctico. En la parte práctica se guiará a los alumnos en la resolución de situaciones que tienen como finalidad la fijación de conocimientos teóricos.

La totalidad de alumnos de la cátedra contarán con la posibilidad de acceder al foro de cátedra a través de la página de la universidad. En este foro podrán formular y responder preguntas relacionadas con los contenidos de la asignatura y temas de cátedra. Los docentes intervendrán exclusivamente cuando alguna de las informaciones vertidas no se ajusten a la realidad o generen posibles confusiones.

EXPERIENCIAS DE LABORATORIO/ TALLER / TRABAJOS DE CAMPO:

Para fortalecer y verificar la correcta presentación de las distintas soluciones, propuestas por el alumno, se cuenta con Laboratorios en los cuales se proceden a codificar los programas desarrollados en clase. Fijando en proceso de conocimiento. Para cada una de las unidades temáticas se desarrollan distintas actividades que fortalecen las mismas. Consideramos que la práctica en los laboratorios provee un entendimiento no tan abstracto ya que se pueden visualizar los resultados en forma inmediata luego de su correcta codificación.

METODOLOGÍA DE EVALUACIÓN:

La evolución del proceso de enseñanza aprendizaje tiene carácter de permanente e integral y contempla la adquisición de conocimientos (aspecto este que se observa de manera específica en cuatro momentos durante el ciclo lectivo), la formación de actitudes, el desarrollo de capacidades de análisis, destrezas y habilidades para encontrar información y resolver situaciones que se le presentan.

1. El dictado de la materia se dividirá en clases teóricas y de realización de trabajos prácticos en clase y laboratorio.
2. En las clases prácticas los alumnos, orientados por los docentes a cargo de las mismas, resolverán problemas de aplicación de los temas vistos en clase. Las guías de trabajos prácticos estarán disponibles con anterioridad a la fecha de realización de cada uno de los trabajos. Dado el tiempo disponible, los docentes procederán al planteo de los objetivos de cada trabajo y orientarán al alumnado en cuanto a la metodología de realización. Los trabajos prácticos a desarrollar en el curso permitirán una evaluación continua.

3. Aquel alumno que obtenga **2 (dos) calificaciones menores a 4 (cuatro) puntos**, consecutivas o alternadas, pierde su condición de regular debiendo recurrar la asignatura. En este caso la nota que se consignara en la planilla de calificaciones para el ciclo lectivo es **ausente**.
4. En razón de la modularidad establecida para las evaluaciones parciales, los días en que se rinden éstas, (tiempo aproximado 120 minutos) no habrá suspensión de actividades. Con posterioridad a la realización del parcial, los docentes explicaran y / o desarrollaran según corresponda cada uno de los temas evaluados. El resto de la clase será dedicado a actividades prácticas o teóricas según necesidad. Para el caso de recuperatorios, podrán desarrollarse en un día habitual de cursada. La instancia de recuperación tendrá lugar la última semana de curso normal (se pueden rendir hasta **un parcial** durante el horario normal de clase y manteniendo los límites de tiempo preestablecidos para cada parcial).
5. La condición del alumno con respecto a la asignatura, deberá definirse como máximo al concluir el periodo de cursada que nos ocupa.
6. Para los recuperatorios, la nota obtenida en el parcial que se recupera reemplaza a la original, registrando como calificación del parcial, el nuevo valor obtenido en el recuperatorio.
7. La aprobación de la materia (**aprobó**) se obtendrá si se cumplen los siguientes requisitos:
 - Aprobar la totalidad de exámenes parciales (directamente o a través de recuperatorio) con una nota igual o superior a 7 (siete) puntos.
 - Asistir como mínimo al 75 % de las clases.
8. La condición de alumno regular (**cursada**, habilita para rendir examen final) se obtendrá si se cumplen los siguientes requisitos:
 - Aprobar la totalidad de exámenes parciales (directamente o a través de recuperatorio) con una nota igual o superior a 4 (cuatro) puntos.
 - Asistir como mínimo al 75 % de las clases.
9. Quienes no cumplan con el requisito de asistencia (presencia igual o superior al 75 %) perderán su condición de alumno regular y merecerán la condición de **ausente**.
10. Aquellos alumnos que habiendo satisfecho el requerimiento de asistencia, no logren la aprobación o regularidad (ver puntos 9 y 10) merecerán la condición de **reprobados o ausentes**. La condición de **reprobado** se consigna cuando se haya presentado en la instancia de recuperación y no satisfizo los objetivos de aprobación o regularidad (ver puntos 9 y 10). Para el caso que no se haya presentado a la instancia de recuperación teniendo necesidad de hacerlo, la condición a consignar es **ausente**.-
11. **Exámenes libres**. Aquellos alumnos que consideren tener los conocimientos suficientes de los temas que se imparten podrán optar por este tipo de examen. La evaluación con esta característica será del mismo nivel de complejidad que las correspondientes al examen regular aunque más extensas. El alumno comenzara con

una evaluación teórica y al concluir la misma continuará con una evaluación práctica. Para lograr la aprobación de la asignatura deberá satisfacer ambas instancias. Este tipo de examen tendrá lugar exclusivamente en la primera fecha de cada llamado a examen final.

CRONOGRAMA ORIENTATIVO DE ACTIVIDADES

DESARROLLO DE CONTENIDOS 1ER. CUATRIMESTRE - ELEMENTOS DE PROGRAMACION						
Inicio Cuatrimestre 04 de Abril, Fin Cuatrimestre 16 de Julio				2016	1º Cuatrimestre	
MES	Sem ana	Nro .Cla	SEM- FECHA	TEMAS	Observaciones	
MARZ O ABRIL	1	1	LU-28	Introducción General/Conoc.Básicos Computador	Unidad 1	
		2	SA-02	Interpretación de problemas	Unidad 1	
A B R I L	1	1	LU-04	Diseño de Estrategias /Algoritmos	Unidad 1	
		2	SA-09	Estructuras Básicas /Combinaciones de Estruc	Unidad 2	
	2	3	LU-11	Repetir-Mientras-Selección- Funciones/ejer.	Unidad 2 / Unidad3 -LABORATORIO	
		4	SA-16	Lenguaje C -Estructura Gral./Variables-Tipos	Unidad 3 / Unidad 4	
	3	5	LU-18	Estructura de Selección Simple -IF	Unidad 5 -LABORATORIO	
		6	SA-23	Contador,Acumulador/Concatenación	Unidad 5	
	4	7	LU-25	Swich/Ejercitación Aplicada	Unidad 5 -LABORATORIO	
		8	SA-30	Iteración Definida / Ejercicios	Unidad 6	
M A Y O	5	9	LU-02	FOR-1 -Max/Min	Unidad 6 -LABORATORIO	
		10	SA-07	Uso de Señales/Banderas- Iteración Condicio	Unidad 6	
	6	11	LU-09	While / Ejercicios de Aplicación	Unidad 6 -LABORATORIO	
		12	SA-14	Do While / Estr.Combinadas	Unidad 6	
	7	13	LU-16	Programación Modular/Prototipo/parametros	Unidad 7 -LABORATORIO	
		14	SA-21	Programación Modular/Prototipo/parametros	Unidad 7 -LABORATORIO	
	8	15	LU-23	Arrays / Matrices	Unidad 8	
		16	SA-28	Matrices /funciones	Unidad 8 -LABORATORIO	
	J U N I O	9	17	LU-30	Ejercicios de Aplicación	Unidad 8
		10	18	SA-04	Ejercicios de Aplicación	Unidad 8
19			LU-06	1er . Parcial	Parcial	
20			SA-11	estructuras	Unidad 9	
11		21	LU-13	Array estructuras	Unidad 9	
		22	SA-18	Archivo	Unidad 10	
12		23	LU-20	Archivo	Unidad 10	
		24	SA-25	Archivo -Corte de Control	Unidad 11	
13		25	LU-27	Archivo Ejercicios	Unidad 11	
		14	26	SA-02	Archivo Ejercicios	Unidad 11
27	LU-04		2do. Parcial	Parcial		
28	SA-09		Aclaraciones para Recuperatorio			
29	LU-11		Recuperatorios 1ro o 2do Parcial	Recuperatorio		
J U L I O	15	30	SA-16	Entrega Notas - cierre Planillas		

CONDICIONES DE CURSADA Y APROBACIÓN

Según lo establecido en la RHCS 054/2011 (Régimen académico integrado)

“Declaro que el presente programa de estudios de la asignatura Elementos de Programación, es el vigente para el ciclo lectivo 2016, guarda consistencia con los contenidos mínimos del Plan de Estudios”

Firma

Edgardo Javier Moreno

Aclaración

29/03/2016

Fecha