



CÓMO IMPACTA LA RECURSIVIDAD EN LA ENSEÑANZA EN CIENCIAS DE LA COMPUTACIÓN

Mario Enrique Quintana¹

Jorge Enrique Sagula²

Florencio Isidro Monzón³

Educação Matemática no Ensino Superior

Resumen: Aquí se aplican estrategias recursivas como enfoque metodológico en la enseñanza de asignaturas de la Licenciatura en Sistemas de Información en la UCP; pilar del Proyecto de Investigación “La recursividad en contenidos curriculares de asignaturas de ciencias de la computación”. Para verificar abordaje e implementación de estrategias recursivas se seleccionaron tres espacios curriculares del I Cuatrimestre (2º a 4º año) en tal licenciatura, analizando contenidos, metodologías, criterios evaluativos y recursos utilizados. Se diseñó y efectuó una encuesta a los responsables de las asignaturas: Programación III, Programación en Ambientes Web y Análisis Matemático I, realizando un seguimiento y un monitoreo exhaustivo de las clases, para profundizar los resultados obtenidos y conocer las conclusiones sobre la aplicación de estrategias recursivas en contenidos propios de sus asignaturas. La encuesta reveló modificaciones sustanciales en el abordaje de los contenidos propios, tanto en formación específica del profesor como de los alumnos, mejorando la comprensión de temas como: paradigma funcional, sucesiones y series, algoritmos complejos, lenguajes y modelos de programación. Los profesores valoraron el uso de algoritmos recursivos para resolver problemas disciplinares dando herramientas para comprensión, programación, construcción de algoritmos, elaboración de fórmulas, etc., y manifestaron haber observado avances significativos en distintas evaluaciones en trabajos prácticos y otras instancias. Consideramos importante explorar y analizar espacios conexos en temas de Recursividad y afines en todas las asignaturas de la carrera.

Palabras Clave: Recursividad. Algoritmos Recursivos. Currículo. Programación Informática. Teoría de la Información.

1. INTRODUCCIÓN

La recursividad en ciencias de la computación, no sólo constituye una forma de resolver problemas, pues permite optimizar procedimientos y consecuentemente, evitar pérdidas innecesarias de tiempo y recursos.

¹Lic. en Enseñanza de la Matemática. Prof. en Matemática y Cosm. Esp. en Docencia Universitaria - F. de Ingeniería y Tecnología, Universidad de la Cuenca del Plata, Sede Formosa. Argentina. quintanamario_for@ucp.edu.ar

²Lic. en Matemática Aplicada. - F. de Ingeniería y Tecnología, Universidad de la Cuenca del Plata, Argentina. JorgeSagula@gmail.com

³Contador Público. Esp. en Docencia Universitaria. - F. de Ingeniería y Tecnología, Universidad de la Cuenca del Plata, Sede Formosa. Argentina. cpisi22@hotmail.com

Susanna Epp aporta: “La Recursividad es una de las ideas centrales en ciencias de la computación” (1995, p. 427). Además, “la potencia de la recursión se fundamenta en la posibilidad de definir un conjunto infinito de objetos con una declaración finita. Igualmente, un número infinito de operaciones computacionales puede describirse con un programa recursivo finito, aunque el programa no contenga repeticiones explícitas” (Wirth, Niklaus.1976, p.126).

Particularmente, la investigación comenzó aplicando estrategias de enseñanza de modo que los alumnos puedan de la Licenciatura en Sistemas de Información de la UCP puedan profundizar la comprensión de *Recursividad* y conceptos afines. Posteriormente, se planteó el abordaje y aplicación de estrategias recursivas en: Programación III, Programación en Ambientes Web y Análisis Matemático I, espacios curriculares desarrollados durante el I Cuatrimestre en distintos cursos de la carrera. Al efecto, se analizaron las metodologías aplicadas, contenidos, criterios evaluativos y recursos, efectuando un seguimiento acorde en las clases correspondientes y finalmente, se relevó la información de los docentes responsables de las asignaturas involucradas con el propósito de profundizar los resultados obtenidos y conocer las opiniones sobre la importancia de la aplicación específica de estrategias recursivas en los respectivos contenidos.

En Análisis Matemático I se profundizó el concepto de recursividad, introduciendo “recurrencia” para definir una función. Así, desde la unidad temática “sucesiones y series”, se presentó como objetivo que los alumnos: constituyan una secuencia lógica, para establecer relaciones y regularidades entre números, construyan sucesiones y series por recurrencias, identifiquen el término enésimo, etc., considerando el 5º Axioma de Peano (Principio de Inducción Completa).

Los lenguajes diseñados para Programación Lógica y Programación Funcional ofrecen la recursión como medio de repetición directa para el programador, siendo necesario que el alumno comprenda la importancia del tema y adquiera las habilidades necesarias para programar. Desde esta perspectiva, el docente propone estos lenguajes, tal que la recursión de cola sea tan eficiente como la iteración, permitiendo a los alumnos expresar otras estructuras repetitivas en términos de recursión.

Actualmente existen diferentes lenguajes de programación para desarrollar en la web, surgidos por tendencias y necesidades de las plataformas. Aquí surge la necesidad de mostrar las ventajas del uso de la recursividad en lenguajes más

conocidos de la Programación Web, surgieron los lenguajes de programación para la web dinámica, permitiendo el uso de algoritmos recursivos.

2. MARCO TEÓRICO

Hay un sinnúmero de trabajos sobre aplicación de la recursividad en metodologías de enseñanza en asignaturas de ciencias de la computación. Buscando la mejor forma de entender “la recursividad” en la vinculación de ciencias de la computación con la matemática, se consideraron artículos abordados por otros autores, a saber:

Chesñear, Maguitman y González [1] en su trabajo “Tecnología informática en un curso de lenguajes formales y teoría de autómatas: un enfoque constructivista”, aplican la teoría de lenguajes recursivos, despertando un significativo interés en los alumnos y obteniendo generadores de discusiones e intercambios de ideas, en la Universidad Nacional del Sur. En la misma universidad, Chesñear [2] presenta consideraciones y ejercicios motivadores para la enseñanza de la recursión, concluyendo que “La recursión es uno de los temas que mayor fascinación ejerce sobre los estudiantes que adquieren sus primeras vivencias en programación a nivel universitario” y, a la vez, evidencia a la recursión como herramienta de la programación.

Di Mare [3] presenta tres ejemplos sencillos de programas que ayudan al alumno que programa a entender rápidamente “el significado de la recursividad y cómo funciona”. El primer ejemplo consiste en una aplicación muy simple de recursividad para crear un comando para el sistema operativo DOS/pc; el segundo corresponde al cálculo factorial escrito en Pascal, y el último corresponde al clásico recorrido PID (Proceso-Izquierda-Derecha) para árboles. Este ejemplo se justifica en componentes diferentes del concepto de recursión.

Rubio Sánchez [4], en la Universidad Rey Juan Carlos trabajó la recursividad como concepto básico de programación mediante problemas combinatorios equivalentes, desempeñando un rol importante en la adquisición de competencias asociativas a la abstracción funcional y descomposición de problemas a través del concepto de inducción. Esta estrategia no sólo aportó a la mejora de la comprensión de la recursividad, sino también de la programación.

Lacave, Molina y Giralt [5], en la Universidad de Castilla desarrollaron el trabajo “Identificando algunas causas del fracaso en el aprendizaje de la recursividad: análisis

experimental en las asignaturas de programación”; más allá que el objetivo principal consistió en identificar las necesidades del alumnado al enfrentarse a la asimilación del concepto de recursividad, demuestran que la recursividad es una poderosa herramienta para resolver problemas complejos, pero es un concepto difícil de entender para los alumnos en el proceso de aprendizaje de programación. Los autores describen una experiencia desarrollada en las asignaturas Fundamentos de Programación I y Metodología de la Programación en la Escuela Superior de Informática en Ciudad Real.

Sanabria Codesal e Hinarejos [6] en “Una experiencia de coordinación entre las asignaturas de Análisis Matemático y Programación”, describen una actividad transversal entre las asignaturas Análisis Matemático y Programación, obligatorias en el primer curso de Ingeniería en Informática de la Escuela Técnica Superior de Ingeniería Informática (ETSINF) de la Universidad Politécnica de Valencia (UPV). Si bien el objetivo fue fomentar el trabajo cooperativo y el aprendizaje activo de los contenidos de dos asignaturas básicas de la titulación, los alumnos realizaron un trabajo grupal, relacionando los contenidos de Programación con los contenidos propios del Análisis Matemático logrando mejorar la comprensión.

3. HIPÓTESIS DE TRABAJO

- Implementar estrategias, aplicando conceptos de recursividad y procedimientos recursivos en la enseñanza de: Programación III, Programación en Ambiente Web y Análisis Matemático I pertenecientes al currículo de Licenciatura en Sistemas de Información posibilitará mejorar la comprensión de contenidos centrales de tales asignaturas.
- Los docentes responsables de los espacios curriculares, deben concientizarse sobre la utilidad de los algoritmos recursivos para acceder a resolver problemas inherentes a su disciplina brindando herramientas para comprensión, programación, construcción de algoritmos y elaboración de fórmulas, en pro de mejorar la formación del futuro egresado.

4. METODOLOGÍA

Considerando que este artículo forma parte del Proyecto de Investigación “*Enseñanza de las funciones recursivas como eje en las ciencias de la computación*”, en la Licenciatura en Sistemas de Información de la UCP, se decidió

estudiar sucesivamente la inserción de Recursividad, Relaciones Recursivas y Algoritmos Recursivos, esenciales en Ciencias de la Computación. En este sentido Schiavoni, Fava y Rosso [7] presentan una herramienta para mejorar el aprendizaje de algoritmos y estructuras de datos en alumnos de primer año de la carrera, posibilitando realizar el seguimiento y análisis de los llamados recursivos, pudiendo inspeccionar el estado de las variables en cada llamado.

El estudio reviste un enfoque cuali-cuantativo por la naturaleza de la situación.

Así, se conceptualizó aplicar estrategias recursivas en: Programación III, Programación en Ambientes Web y Análisis Matemático I, con seguimiento integral de las clases específicas relacionadas con recursividad.

Previamente, se concretaron reuniones con los profesores responsables de tales asignaturas, para acordar metodologías, procedimientos, contenidos, criterios evaluativos y recursos, y luego, se encuestó a dichos profesores responsables y así evaluar dicho abordaje y conocer sus conclusiones sobre la aplicación de estrategias recursivas en contenidos de sus asignaturas.

4.1 Enfoque en Análisis Matemático I

En Análisis Matemático I se introduce el concepto de recurrencia para definir una sucesión como función.

Específicamente, se enfatizó en la importancia de conocer un dato específico y así poder calcular los restantes, concepto esencial en la definición por recurrencia. La actividad se expandió de modo de formalizar una definición de sucesiones y series, para comprender, luego, la importancia del enésimo término.

Análogamente, se propusieron algunas sucesiones conocidas, como la Sucesión de Fibonacci, para incentivar a los alumnos en la construcción de sucesiones desde la recursividad, presentando:

“Problema de los conejos: ¿Cuántas parejas de conejos tendremos a fin de año si, comenzando con una, produce cada mes una pareja que procrea a los dos meses de vida?”

Luego, se requirió responder la consigna:

- *A partir de la relación que encuentres escribe en notación matemática una definición de la función “dado un mes cualquiera obtener el número de parejas de conejos”.*

Varios alumnos, a partir del desconocimiento de la sucesión, emplearon algoritmos recursivos para resolver la situación. Por caso, se detectaron estrategias de resolución como se observa en la **Figura I**, para $n = 5$:

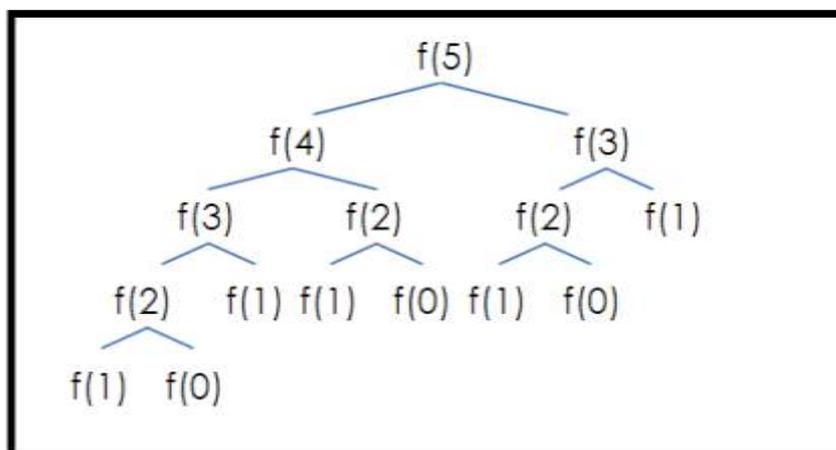


Figura I. Árbol de llamadas del cálculo recursivo.

Otros alumnos emplearon funciones recursivas propiamente dichas.

En las clases subsiguientes se propusieron actividades de abordaje de progresiones aritméticas y geométricas para obtener cualquier elemento de estas sucesiones utilizando el valor previo.

En base a las construcciones de estas progresiones se propuso hallar una fórmula (expresión algebraica) para encontrar la suma sin necesidad de utilizar todos los términos de la progresión tratada.

En las clases siguientes se analizó la convergencia o divergencia de sucesiones y series aplicando criterios de convergencia para ambos casos hasta operar con series de potencias. Estas consideraciones permitieron comprender la deducción de las series de Taylor y Mac Laurin, importantes en Análisis Matemático I.

Observando las estrategias de aprendizaje de los alumnos y las estrategias de enseñanza de los profesores responsables, se destacaron estos aspectos:

- Exploración e investigación constante sobre la temática.
- Interacción permanente entre compañeros de clase.
- Promoción y/o Estimulación del proceso de trabajo sin que el profesor resuelva todo.
- Incentivación al estudiante como responsable del conocimiento del tema en estudio.

- Descubrimiento de la funcionalidad de cada problema planteado de parte de los alumnos.
- Vislumbramiento de estrategias de aprendizaje innovadoras y creativas en los alumnos.

En el espacio curricular se trabajó esta temática desde la bibliografía básica propuesta por el docente (Rabuffetti, H. [8] y Repetto, C. [9]).

4.2 Enfoque en Programación III

En programación “no todas las funciones pueden llamarse a sí mismas”, sino que deben diseñarse especialmente para ser recursivas, evitando conducir a bucles infinitos, o que el programa termine inadecuadamente. Además, no todos los lenguajes de programación permiten usar recursividad.

En Programación III se aborda el uso de la lógica como lenguaje de programación; retomándose un programa como un conjunto de fórmulas lógicas correspondientes a la especificación del problema a resolver, para dar curso a la programación funcional.

Analizando el Paradigma Funcional, los alumnos infieren que:

- Del paradigma se construye el lenguaje de Programación Funcional.
- Se basa en descripciones al evaluar funciones o en la aplicación de funciones a valores conocidos.
- El mecanismo básico es evaluar o llamar una función; contemplando:
 - La propia evaluación de la función.
 - La transferencia de valores como parámetros a la función.
 - La obtención de valores resultantes como valores devueltos de las funciones.
- Que no involucra variables o asignación a variables, sólo constantes, parámetros y valores.

Es menester aclarar que los alumnos en esta etapa demostraron dominio y destreza en el manejo de funciones en general, y en particular, funciones recursivas.

Luego, se construyó la función recursiva, desde la perspectiva de la programación, así:

- Una función recursiva se resuelve con una llamada a sí misma, cambiando el valor de un parámetro en la llamada a la función.

- Mediante las sucesivas llamadas recursivas a la función se obtienen valores que, computados, sirven para lograr el valor de la función llamada originalmente.

- El proceso de llamadas recursivas siempre debe finalizar en una llamada a la función resuelta directamente, sin necesidad de invocar nuevamente a la función.

Ante esta conceptualización y caracterización de las funciones recursivas se propuso construir procedimientos o algoritmos en base a estas consignas a los alumnos:

“Una función es recursiva cuando se llama a sí misma”

Consigna 1: Programar un algoritmo recursivo que calcule:

(a) Factorial de un número.

$0! \rightarrow 1$

$1! \rightarrow 1$

...

$n! \rightarrow n \cdot (n - 1)!$

Los alumnos utilizaron esta lista para implementar la función:

```
(DEFINE FACTORIAL
  (LAMBDA (N)
    (IF (< N 2)
      1
      (* N (FACTORIAL (- N 1))))))
```

(b) Un número de la serie de Fibonacci.

$F(0) \rightarrow 0$

$F(1) \rightarrow 1$

...

$F(n) \rightarrow F(n - 1) + F(n - 2)$

Entonces, los alumnos utilizaron la siguiente lista:

```
(DEFINE FIBO
  (LAMBDA (N)
    (COND
      ((= N 0) 0)
      ((= N 1) 1)
      (ELSE (+ (FIBO (- N 1)) (FIBO (- N 2))))))
```

Consigna 2: Se pidió a los alumnos explorar otras relaciones, operaciones, etc., donde sea posible utilizar algoritmos recursivos; las principales propuestas fueron:

:

- Programar un algoritmo recursivo que permita invertir un número.
- Programar un algoritmo recursivo que permita sumar los dígitos de un número.
- Programar un algoritmo recursivo que calcule el Máximo Común Divisor de dos números.

Consigna 3: Se propusieron ejercicios algo más complejos:

- Plantear un programa mediante un algoritmo recursivo para sumar los elementos de un vector.
- Plantear un programa mediante un algoritmo recursivo para multiplicar los elementos de un vector.
- Plantear un programa mediante un algoritmo recursivo para sumar los elementos de una matriz.

Seguidamente, se observan algunas soluciones (**Figura II** y **Figura III**):

```
int suma_vec(int v [], int n) {
  if (n == 0) {
    return v [n];
  }
  else {
    return suma_vec(v, n - 1) + v [n];
  }
}
```

Figura II. Suma de Vectores

```
int suma (int fila, int col, int orden, int mat [] [])
{
  if (fila == 0 && col == 0)
    return mat [0] [0];
  else
    if (col < 0)
```

Figura III. Suma de Matrices

Después, se realizó un estudio completo sobre el lenguaje de programación funcional SCHEME, deduciendo que su sintaxis es muy reducida, en comparación con

muchos otros lenguajes; no necesita reglas de precedencia, pues, esencialmente, carece de operadores: usa notación prefijada para todas las llamadas a función.

Al familiarizarse los alumnos con este lenguaje, detectaron que SCHEME se relaciona con algoritmos recursivos por:

- Abstracción de Datos: pueden construirse datos complejos desde datos más simples.
- Reconocimiento de Patrones: SCHEME es capaz de reconocer la función a la que llama a partir de la misma.
- Funciones de Orden Superior: las funciones pueden ser argumentos de otras funciones.

Las conclusiones obtenidas por los alumnos sobre estas características, se relacionaban con conocimientos previos sobre recursividad.

Esta forma de trabajar en las clases permitió que los alumnos realicen y propongan sus propias estrategias con escasa intervención del docente; así, los alumnos se responsabilizan sobre su propio aprendizaje construyendo algoritmos o programando algoritmos desde la recursión.

La bibliografía básica utilizada en la asignatura fue: Lenguajes de Programación de Loudon Kenneth [10] y de Allen B. Tucker y Robert Noonan [11].

4.3 Enfoque en Programación en Ambiente Web

Aquí se estudia la recursividad como una técnica de programación originada en cálculos matemáticos, consistente en describir cálculos o acciones autoalusivamente, esto es, resolver problemas describiéndolos en términos de situaciones más sencillas. Esta técnica puede entenderse particularmente en Programación Web con subprogramas, planteando la resolución de un problema en términos de subproblemas más sencillos.

Se abordó la estrategia de enseñanza-aprendizaje empleando las Torres de Hanoi (**Figura IV**); tal que los alumnos propusieron soluciones recursivas relativamente sencillas, sobre los conocimientos de Recursividad para resolver problemas cuya solución es más compleja iterativamente y luego, programaron en Lenguaje Pascal.

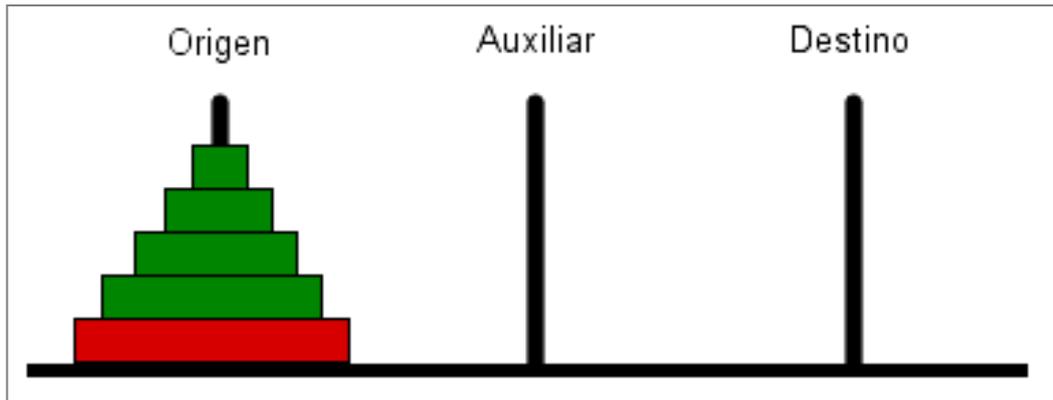


Figura IV. Torres de Hanoi

Se presentó la Función de Ackermann, definiendo:

$$Ack(0; n) = n + 1$$

$$Ack(m; 0) = Ack(m - 1; 1); \text{ si } m > 0$$

$$Ack(m; n) = Ack(m-1; Ack(m; n - 1)) \text{ si } m; n > 0$$

Se pidió a los alumnos que empleen tal función para hallar la imagen para ciertos pares (m,n).

Los alumnos construyeron algoritmos y/o procedimientos, destacando su creatividad en materia recursiva (**Figura V**), y luego programaron la función en el Lenguaje Pascal.

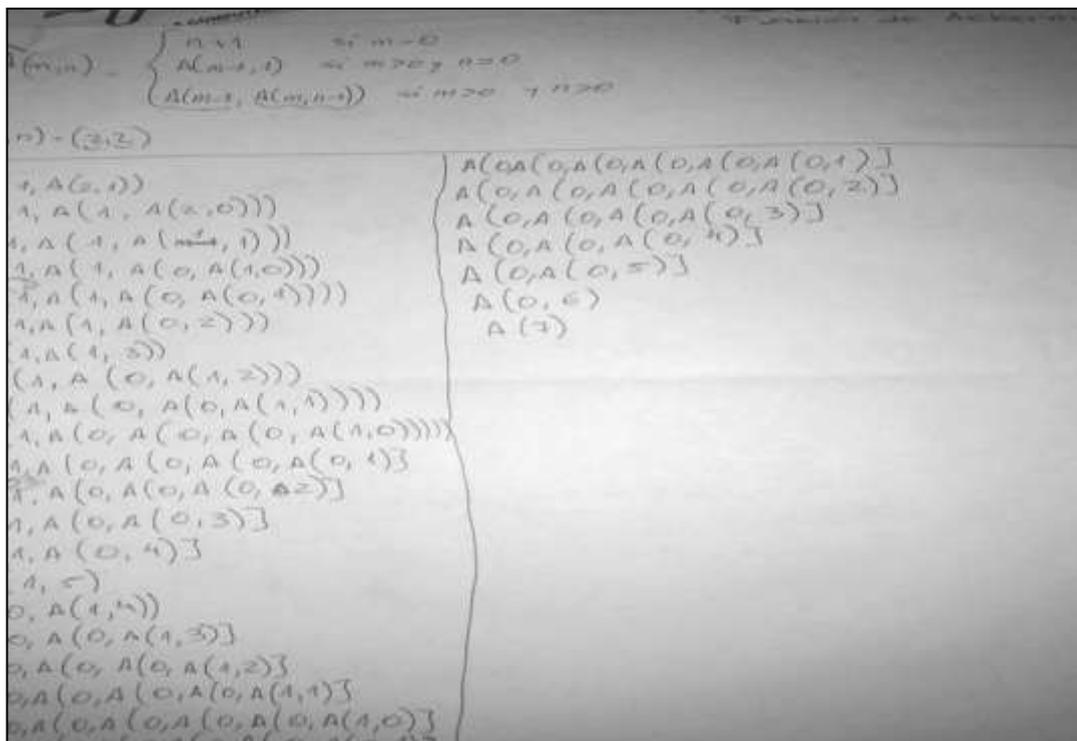


Figura V. Función de Ackermann para m=n=2

Después, se abordaron los temas: Corrección de subprogramas recursivos, Principio de Inducción Completa, Recursividad mutua o cruzada, destacando el muy buen desempeño de los alumnos, mostrando un destacado nivel de comprensión.

Seguidamente, en las clases de Programación Web, el docente presentó a los alumnos diferentes actividades y ejercicios para afianzar estos conceptos:

Ejercicios

Consigna 1: Escribir una función recursiva para calcular el término enésimo de la secuencia de Lucas: 1, 3, 4, 7, 11, 18, 29, 47, ...

Consigna 2: Dado el programa:

Program Invertir (input, output);

{Se lee una línea del input y se escribe invertida}

procedure InvertirRec;

var

c: char;

begin

Read(c);

if c <> '.' then begin

InvertirRec;

Write(c)

end

end; {InvertirRec}

begin {Invertir}

WriteLn('Teclee una cadena de caracteres ("." para finalizar)');

InvertirRec

end. {Invertir}

Analizar su comportamiento y estudiar el resultado para la secuencia de entrada "aeiou", describiendo la evolución de la pila recursiva. Obsérvese que el uso de esta pila recursiva permite recuperar los caracteres en orden inverso al de su lectura.

3. Definir subprogramas recursivos para los siguientes cálculos:

$$a) \left(1 + \frac{1}{2} + \dots + \frac{1}{n}\right) = \left(1 + \frac{1}{2} + \dots + \frac{1}{n-1}\right) + \frac{1}{n}$$

$$b) x_0 = 1$$

$$x^n = (x * x)^{n/2} ; \text{ si } n > 0 \text{ y es par}$$

$$x^n = x * (x^{n-1}); \text{ si } n > 0 \text{ y es impar}$$

En el resto de los ejes temáticos de la asignatura se introdujeron, elementalmente, las estructuras de datos recursivas como concepto importante de la memoria dinámica, tales como: listas, pilas y colas y árboles.

Es importante aplicar la recursividad en las pilas, pues los alumnos abordan la recursividad al activar una pila, en cada llamada recursiva.

Los alumnos abordaron estas temáticas desde Aplicaciones WEB (Martínez, D.; Valderas Aranda, P. y López, O. [12]).

5. ENCUESTAS Y ENTREVISTAS A DOCENTES

Analizadas las encuestas relevadas del trabajo de los docentes se destaca que los mismos reafirman la importancia de la recursividad en el “mundo de la computación” permitiendo ratificar su valor en sus asignaturas.

Los docentes entrevistados manifestaron que los alumnos, a partir del conocimiento previo sobre Recursividad construyeron sus propias estrategias de aprendizaje para comprender otros contenidos en la misma asignatura; y, todos acordaron que las características inherentes a la recursividad, permiten, también buscar algoritmos iterativos.

Sobre las estrategias de enseñanza de la temática, los docentes expresaron que la Recursividad fue fundamental en la enseñanza de los temas de su asignatura, en pro de construir algoritmos para resolver problemas específicos.

6. RESULTADOS OBTENIDOS

Se destaca la solvencia de los alumnos en las distintas asignaturas en contenidos estrechamente relacionados con la temática. En nuestra opinión, y luego de haber presenciado el dictado de algunos temas centrales, observando las estrategias de resolución de los alumnos, los resultados de las estrategias de

enseñanza aplicadas inicialmente fueron conducentes a este logro, concluyendo que los alumnos profundizaron la comprensión del tema.

Durante las clases, se percibió en los alumnos:

- Dominio de los algoritmos recursivos.
- Aplicación correcta de procedimientos para programar en cualquier lenguaje.
- Pensamiento recursivo en resolución de situaciones y ejercicios.
- Construcción adecuada de la sintaxis de programación.
- Elaboración deductiva de conceptos y propiedades desde los conceptos recursivos.
- Creatividad en la resolución de situaciones y ejercicios, en base a algoritmos y procedimientos recursivos.

Sin embargo, el valor más saliente es destacar la importancia de la Recursividad y sus temas afines, y su tratamiento como eje en numerosas asignaturas de la carrera.

Destácase que este estudio sólo cubrió espacios curriculares desarrollados durante el I Cuatrimestre Lectivo; dejando pendiente promover, en futuras investigaciones, la exploración en otras asignaturas de la Licenciatura en Sistemas de Información.

7. ANÁLISIS Y CONCLUSIONES

Nuestro principal objetivo fue demostrar por qué la recursividad es importante en ciencias de la computación. Este estudio confirma hallazgos previos y lo supuesto inicialmente; a pesar que en esta etapa, sólo se abordaron tres asignaturas: Programación III, Programación en Ambiente Web y Análisis Matemático I.

Conforme a las consideraciones expuestas en “Resultados Obtenidos”, se concluye que la comprensión de la Recursividad como herramienta y su gran potencia, favorece la resolución de problemas naturalmente, llegando a corroborar la solución del problema, en cualquier asignatura.

Considerando las afirmaciones de los docentes encuestados, concluimos que emplear estrategias recursivas para abordar ejes centrales en ciencias de la computación ha mejorado ostensiblemente la enseñanza de otros contenidos.

Reafirmamos la necesidad de abordar y trabajar Recursividad en otras asignaturas de la Licenciatura en Sistemas de Información y así profundizar el

conocimiento de la temática para establecer y confirmar la importancia de las Funciones Recursivas en Ciencias de la Computación.

8. REFERENCIAS

CHESÑEVAR, C.; MAGUITMAN, A.; GONZÁLEZ, M. y COBO M. Tecnología Informática en un curso de lenguajes formales y teorías de autómatas: un enfoque constructivista. **Actas** del II Workshop de Tecnología Informática Aplicada a la Educación - IX CACIC. Pp. 245-255. UNSur, Bahía Blanca. Buenos Aires. Argentina. 2003.

CHESÑEVAR, C. Algunas consideraciones y ejercicios motivadores para la enseñanza de la recursión. Trabajo presentado en el **II Ateneo de profesores universitarios de computación**. UNSur. Bahía Blanca. Buenos Aires. Argentina. 1994.

DI MARE H. Tres formas diferentes de explicar la recursividad. **Revista Ingeniería de la Universidad de Costa Rica**. Vol. 6, Nº 2. Pp. 31-44. 1996.

RUBIO SANCHEZ, (2011). Enseñanza de la recursividad mediante problemas combinatorios. Trabajo presentado en **III Seminario de Investigación en tecnologías de la Información aplicadas a la Educación**. BN: 978-84-9982-245-7. ISBN electrónico: 978-84-15454-71-7. Biblioteca de Educación del MECyD. Madrid. España. 2011.

LACAVE, C.; MOLINA, A. y GIRALT, J. Identificando algunas causas del fracaso en el aprendizaje de la recursividad. Jornadas de Enseñanza Universitaria de la Informática. **Libro de Actas** SBN: 978-84-695-8051-6 . V XIX. Pp. 225-232. Ciudad Real: Universitat Jaume I. ESTyCE. Castillon. España. 2013.

MARTÍNEZ HINAREJOS, C.; SANABRIA CODESAL, Esther “Una experiencia de coordinación entre las asignaturas de Análisis Matemático y Programación de la ETSINF”. **Actas** de la VIII Jornadas de Redes de Investigación en Docencia Universitaria, Alicante. Libro de Actas ISBN 978-84-693-6845-9. V 8. Pp 1050. España. 2010.

SCHIAVONI, M.; FAVA, L.; ROSSO, J. Una herramienta educativa para mejorar la comprensión de algoritmos y estructuras de datos. “X Congreso sobre Tecnología en Educación & Educación en Tecnología”. **Libro de Acta** ISBN: 978-950-656-154-3. Pp. 617-623. Corrientes. Argentina. 2015.

9. BIBLIOGRAFÍA DE ASIGNATURAS

RABUFETTI, H. **Introducción al Análisis Matemático** - Cálculo I. Buenos Aires, El Ateneo. 2001.

REPETTO, C. **Manual de Análisis Matemático**. Primera y segunda partes. Buenos Aires, Macci. 1981.

LOUDEN KENNETH C. **Lenguajes de Programación**. Principios y Prácticas.. Edt. Cengage Learning Thomson Internacional. 2004.

ALLEN B. y NOONAN R. **Lenguajes de Programación**. Principios y Paradigmas.. Edt. Mc Graw-Hill. 2003.

MARTÍNEZ, D.; VALDERAS ARANDA, P.; LÓPEZ, O. **Aplicaciones WEB Un Enfoque Práctico**. Editorial: Alfaomega Ra-Ma. 3ª edición. 2005.