

Práctica Docente en la asignatura Algoritmos y Estructuras de Datos

Cristian A. Martínez^{*†}, Carlos Nocera^{*‡}, Diego Rodríguez^{*§}, Ismael Orozco^{*¶} y Eduardo Xamena^{*||}

^{*}Departamento de Informática

Universidad Nacional de Salta, Salta, Argentina

Emails: cmartinez@unsa.edu.ar[†], carlos.nocera@gmail.com[‡], drodriguez@unsa.edu.ar[§]

iorozco@exa.unsa.edu.ar[¶], examena@unsa.edu.ar^{||}

Abstract—La acreditación de carreras universitarias en nuestro ámbito ha permitido que muchas de ellas mejoren sus contenidos curriculares en pos de equiparar y uniformizar saberes fundamentales que un profesional debe poseer. En ese sentido, la acreditación de la Licenciatura en Análisis de Sistemas de la Universidad Nacional de Salta (Argentina) generó cambios positivos a nivel curricular, como así también en cuanto a formación de posgrado, tecnológico, de infraestructura, entre otros. En este trabajo se describen experiencias en el proceso de enseñanza-aprendizaje de una asignatura de 2do año de dicha carrera.

Index Terms—Acreditación; Algoritmos; Estructuras de Datos; Práctica Docente; TICS

I. INTRODUCCIÓN

La Comisión Nacional de Evaluación y Acreditación Universitaria (CONEAU)¹, dependiente del Ministerio de Educación, tiene como misión asegurar y mejorar la calidad de las carreras universitarias argentinas por medio de actividades de evaluación y acreditación. En ese sentido, la carrera de Licenciatura en Análisis de Sistemas (plan 1997) de la Universidad Nacional de Salta (UNSa) fue evaluada por la CONEAU. Como resultado de dicha evaluación y a los efectos de cumplir con las recomendaciones realizadas por dicha institución para alcanzar la acreditación de la carrera, el Departamento de Informática² y la Comisión de Carrera presentaron un nuevo plan de estudios.

En este trabajo, abordaremos las acciones realizadas en la asignatura Algoritmos y Estructuras de Datos, perteneciente a dicho plan.

La motivación del trabajo consiste en mostrar avances que se han logrado en la asignatura, en cuanto a la enseñanza y evaluación de los contenidos curriculares; en cómo la misma articula de manera horizontal y vertical con otras. Por otra parte, los avances y las acciones realizadas serán sustentadas con bibliografía específica. Pero además, indicaremos cuáles son los nuevos desafíos que se quieren afrontar para mejorar el proceso de enseñanza-aprendizaje.

El trabajo se organiza de la siguiente manera: La sección II está dedicada a describir aspectos de la asignatura a nivel curricular. En la sección III se mencionará la metodología de

trabajo y aspectos curriculares llevados a cabo en los inicios de la asignatura. En la sección IV abordaremos la situación actual. La sección V describe acciones que se esperan implementar en la asignatura durante el primer cuatrimestre de 2018 y a corto plazo. Finalmente, las conclusiones aparecen en la sección VI.

II. ALGORITMOS Y ESTRUCTURAS DE DATOS

En esta sección, describiremos ciertos aspectos de la asignatura que permitan conocer de manera introductoria, sus contenidos curriculares, relación con otras asignaturas y ubicación dentro de los planes de estudios de las carreras para las que se dicta.

Algoritmos y Estructuras de Datos es una asignatura que se dicta conjuntamente para las carreras de Licenciatura en Análisis de Sistemas (LAS) Plan 2010 (Res.CS 135/2010³, Res.CD-EXA 403/2012⁴) y Tecnicatura Universitaria en Programación (TUP) (Res.CS. 596/2011⁵). En ambas, está ubicada en el primer cuatrimestre del segundo año, con una carga horaria semanal de 8 horas (4 horas de teoría y 4 horas de práctica). De acuerdo al régimen de correlatividades para ambos planes, la Asignatura se relaciona con:

- LAS Plan 2010
 - Primer Año: Elementos de Programación, Matemática para Informática, Análisis Matemático I y Programación.
 - Segundo Año: Paradigmas y Lenguajes, y Teoría de la Computación II.
 - Tercer Año: Sistemas Operativos.
- TUP Plan 2011
 - Primer Año: Elementos de Programación, Matemática para Informática, Análisis Matemático I y Programación.
 - Segundo Año: Paradigmas y Lenguajes.
 - Tercer Año: Programación de Aplicaciones Web.

Los contenidos mínimos de la asignatura, propuestos para el plan de estudios LAS 2010 y luego adoptados para el de TUP 2011, son los siguientes: Teoría de las Estructuras Discretas. Definiciones y pruebas estructurales. Teoría de

¹www.coneau.gob.ar

²<http://di.unsa.edu.ar>

³<http://bo.unsa.edu.ar/cs/R2010/R-CS-2010-0135.pdf>

⁴<http://bo.unsa.edu.ar/cdex/R2012/R-CDEX-2012-403.PDF>

⁵<http://bo.unsa.edu.ar/cs/R2011/R-CS-2011-0596.pdf>

Números. Aritmética modular. Estructuras de control. Recursividad. Eventos. Excepciones. Concurrencia. Tipos abstractos de datos: definiciones. Especificación abstracta. Operaciones. Isomorfismo. Contenedores lineales. Estructuras de datos. Tipos de datos recursivos. Representación de datos en memoria. Estrategias de implementación. Manejo de memoria en ejecución. Algoritmos en grafos: Algoritmos de análisis y manipulación de grafos. Costos. Aplicación. Estructuras arbóreas: árboles generales y n-arios, binarios, balanceados. Estrategia de diseño de algoritmos.

De acuerdo a los contenidos mínimos, se desprende que la asignatura tiene una fuerte base teórica en Matemática y Programación. Por otra parte, es muy fuerte la relación que tiene con asignaturas de segundo y tercer año de ambas carreras. Respecto a planes de estudios anteriores, debemos decir que Algoritmos y Estructuras de Datos se relaciona (al menos, en base a ciertos contenidos curriculares) con la asignatura Algorítmica, Estructuras de Datos y Matemática para Informática (AEDMI), perteneciente a LAS Plan 1997 (Res. CS. 185/96).

III. LA ASIGNATURA EN SUS INICIOS

En su primera fase (año 2011), la forma de trabajo era similar a la realizada en Algorítmica, Estructuras de Datos y Matemática para Informática, asignatura del plan de estudio anterior. Esto, en parte se debió a que el plantel docente de ambas asignaturas era el mismo.

A continuación, listaremos el programa analítico, el programa de trabajos prácticos y el régimen de regularidad de la asignatura (aprobados por Res.CD-EXA 179/2011⁶). Posteriormente, realizaremos un análisis de los mismos.

A. Programa Analítico

Unidad 1: Teoría de las Estructuras Discretas

Los algoritmos, la matemática y las estructuras de datos. Naturaleza de los números naturales, operaciones fundamentales. Teorema de la división, divisibilidad, números primos, algoritmos. Máximo Común Divisor. Teorema Fundamental de la Aritmética. Aritmética Modular. Congruencias. Introducción a la Teoría de Grupos: Los grupos aditivos y multiplicativos de Z_p .

Unidad 2: Los Algoritmos y Lenguajes de Programación
Estructuras de control. Flujo de control. Eventos. Excepciones. Representación de los datos en memoria. Memoria estática y dinámica. Manejo de la memoria en ejecución. Algoritmos para memorias dinámicas. Gestión dinámica de la memoria. Liberación explícita de la memoria. Recursividad. Caracterización de la recursividad. Subprogramas recursivos. Metodología de diseño de soluciones recursivas. Concurrencia: concepto. Estrategia de diseño de algoritmos.

Unidad 3: Tipos Abstractos de Datos

Tipos Abstractos de Datos. Concepto. Definición. Especificación abstracta, interfaz, implementación, encapsulamiento, ocultación de información. Operaciones: generadora, constructora, accesoria, modificadoras, asignación y copia. Taxonomía

de los tipos abstractos de datos y la estructura de datos. Isomorfismo.

Unidad 4: Contenedores Lineales

Introducción. Representación de datos en memoria. Organización secuencial de los datos, listas, listas secuenciales, listas ordenadas, listas enlazadas, listas doblemente enlazadas, pilas, colas, colas con prioridad. Aplicaciones de contenedores lineales.

Unidad 5: Grafos

Organización en red, grafos orientados, grafos no orientados, caminos, recorridos. Grafos con costo. Algoritmo de análisis y manipulación de grafos. Aplicaciones de grafo.

Unidad 6: Árboles

Organización jerárquica. Árboles generales y n-arios. Relaciones de parentesco y taxonómicas. Recorridos. Árboles binarios. Equivalencia entre árboles generales y binarios. Árboles basados en la ordenación de sus elementos: árboles ABB, árboles balanceados: AVL, 2-3. Aplicaciones de árboles.

B. Programa de Trabajos Prácticos

- Trabajo Práctico 1: Introducción al diseño de algoritmos. Algoritmos recursivos.
- Trabajo Práctico 2: Teoría de Números. División.
- Trabajo Práctico 3: Teoría de Números. Divisibilidad y primos. Máximo Común Divisor.
- Trabajo Práctico 4: Teoría de Números. Aritmética Modular.
- Trabajo Práctico 5: Grupos y Cuerpos.
- Trabajo Práctico 6: Estructuras de Control. Concurrencia.
- Trabajo Práctico 7: Tipo abstracto de datos.
- Trabajo Práctico 8: Contenedores lineales.
- Trabajo Práctico 9: Grafos I.
- Trabajo Práctico 10: Grafos II.
- Trabajo Práctico 11: Árboles I. Organización jerárquica. Árboles binarios.
- Trabajo Práctico 12: Árboles II. ABB, AVL, 2-3. Aplicaciones.

C. Régimen de Regularidad

Para regularizar, el Estudiante debe:

- Aprobar los 2 exámenes parciales o sus respectivas recuperaciones.
- Alcanzar un 75% de asistencia a clases prácticas.

D. Análisis y Discusión

En relación a los contenidos curriculares y al listado de trabajos prácticos, se puede apreciar una gran cantidad de temas a abordar (como se observa en el programa de trabajos prácticos), con cierta descoordinación entre ellos (por ejemplo, recursión, concurrencia y técnicas de diseño de algoritmos en una misma unidad y al inicio del cursado) y además, poca profundidad en el tratamiento de los mismos.

El desarrollo de los contenidos curriculares se dividía en dos partes, igualmente equilibradas en tiempo; la primera, referida a Teoría de Números y la segunda, a Estructuras de Datos y Algoritmos. Los contenidos de Teoría de Números

⁶<http://bo.unsa.edu.ar/cdex/R2011/R-CDEX-2011-179.PDF>

eran abordados desde un enfoque matemático fuerte y una carga horaria un tanto desproporcional respecto de los demás incluidos en el programa analítico. Respecto a la segunda parte, éstos eran desarrollados generalmente con un enfoque teórico y con poca carga horaria destinada al desarrollo de programas (en lenguaje Pascal). Tanto las clases teóricas como las prácticas eran impartidas en aulas, desarrollándose así los contenidos teóricos y los ejercicios prácticos.

Por otra parte, si bien no consta en el anterior programa analítico, para aprobar la materia el estudiante debía rendir un examen teórico-práctico de los temas vistos.⁷

IV. LA ASIGNATURA EN LA ACTUALIDAD

En esta sección describiremos la forma de trabajo que lleva a cabo la Cátedra para el desarrollo de las actividades académicas de la asignatura. La misma, tiene dos períodos fuertemente marcados: 2012-2013 (de transición) y 2014 a la actualidad (de consolidación).

La Cátedra se compone de 1 Profesor Adjunto (dedicación exclusiva desde 2013) y 2 Jefes de Trabajos Prácticos (dedicación simple y semi-exclusiva desde 2014). Esta organización se mantiene estable desde 2013.

Los cambios que se han realizado en la asignatura desde 2012 a la fecha, se sustentan en los siguientes ejes: articulación, desarrollo de contenidos y metodología de trabajo, y evaluación.

A. Articulación

Uno de los cambios más sustanciales fue conocer y articular contenidos curriculares con las Cátedras con las cuales Algoritmos y Estructuras de Datos tiene relación directa. Esta acción se sustenta en el proceso de enseñanza-aprendizaje de Ausubel[9] a partir de conocimientos previos del estudiante. Pero además, para que el proceso de enseñanza-aprendizaje sea eficiente, es necesario que la instrucción sea formal, organizada y explícita[8]. Es decir, es fundamental identificar los conocimientos previos y a partir de allí, organizar los contenidos curriculares que se impartirán.

Para los contenidos relacionados con Matemática Discreta, desde la asignatura Matemática para Informática se introduce al estudiante en Lógica, Conjuntos, y Estructuras algebraicas. Para los contenidos relacionados con Contenedores Lineales, Recursión, Tipo Abstracto de Datos y Tipos de Datos Recursivos, desde la Asignatura Programación se introduce en Recursividad, Tipos abstractos de datos, y Técnicas de diseño de algoritmos (divide y vencerás, visto desde la modularidad de programas); a su vez, se introduce en tipos de datos dinámicos (lista enlazada simple) y en análisis de eficiencia, los cuales se utilizarán para desarrollar los contenidos de Árboles y Grafos.

Estos conocimientos previos nos permiten entonces, tener en claro de qué manera desarrollar los contenidos de AyED promoviendo así el *aprendizaje significativo* en el estudiante[9].

⁷En la mayoría de las carreras que se dictan en la UNSa, los estudiantes al cursar una materia pasan por dos instancias de evaluación. La primera, de regularización, al final del cursado. La segunda, de aprobación o promoción, luego de haber de haber regularizado la asignatura.

Por otra parte, los conocimientos que el estudiante adquiere durante el cursado, le permiten proseguir el proceso de enseñanza-aprendizaje en los siguientes temas en la asignatura Paradigmas y Lenguajes: Clase, Encapsulación, Reusabilidad, Recursión y Algoritmos concurrentes. También, permiten tratar los siguientes en la asignatura Teoría de la Computación II (TCII): Representación de datos: objetos de datos, datos simples y estructurados, estructuras estáticas y dinámicas; Métodos formales de descripción de sintaxis: grafos sintácticos, gramáticas libres de contexto, y análisis sintáctico. Además, se relaciona en la asignatura Programación de Aplicaciones Web con: Programación del lado del cliente y del lado del servidor. Para ambas unidades que están fuertemente vinculadas al desarrollo de aplicaciones, se precisa un buen manejo de estructuras de datos para un mejor control de la información ingresada, como también para el procesamiento de datos proveniente de la base de datos. Finalmente, en la asignatura Sistemas Operativos con los temas: algoritmos de planificación de procesos, problemas clásicos de la comunicación entre procesos, administración de la memoria con listas enlazadas, algoritmo de reemplazo de páginas óptimo, y algoritmo del banquero para uno y varios recursos.

En lo referente a la articulación horizontal, es importante mencionar la gran interacción existente con la asignatura Programación Orientada a Objetos, en cuanto a la coordinación de conceptos y desarrollo de ejercicios de las guías de trabajos prácticos (por ejemplo en temas como Programación Orientada a Objetos (POO), Tipo Abstracto de Datos (TAD) y Contenedores Lineales). En menor medida, existe interacción con Teoría de la Computación I (TCI), fundamentalmente por los pocos contenidos en común entre ambas asignaturas y porque TCI tiene un enfoque más teórico que práctico.

Como puede apreciarse, AyED es fundamental para las dos carreras. Las reuniones con los docentes responsables de las otras Cátedras, ha permitido que la asignatura siga evolucionando con cambios que han resultado positivos para una mejor enseñanza y formación del estudiante.

B. Desarrollo de contenidos curriculares y metodología de trabajo

A continuación, listaremos los contenidos curriculares propuestos en un nuevo programa analítico (Res.D-EXA 633/2012⁸) como así también el desarrollo de las guías de trabajos prácticos. En ambos casos, puede apreciarse un desarrollo completo, gradual y real de los contenidos mínimos de la asignatura para ambos planes de estudios.

1) *Programa Analítico*: Unidad 1: Introducción a la teoría de las estructuras discretas

Matemática discreta: definiciones previas. Conjuntos numéricos: operaciones fundamentales. Teoría de números: introducción, divisibilidad, teorema de la división, algoritmo de Euclides, máximo común divisor, números primos. Aritmética modular. Congruencia. Introducción a la teoría de grupos.

⁸<http://bo.unsa.edu.ar/dex/R2012/R-DEX-2012-633.PDF>

Unidad 2: Recursividad

Definiciones previas. Tipos de recursividad. Propiedades de la recursividad. Tipos de datos recursivos. Manejo de memoria en ejecución. Funciones y procedimientos recursivos. Aplicaciones.

Unidad 3: Introducción a la abstracción de datos

Analogía. Niveles de abstracción de datos. Tipo de dato abstracto: definiciones, especificaciones, operaciones, implementación, ventajas, isomorfismo. Aplicaciones.

Unidad 4: Contenedores lineales: pilas y colas

Estructura lineal: definición, notación, operaciones, representación de datos en memoria. Listas. Pilas: definición, representación gráfica, operaciones, implementación. Colas: definición, representación gráfica, operaciones, implementación. Colas circulares. Colas con prioridad. Aplicaciones de contenedores lineales.

Unidad 5: Contenedores lineales: listas enlazadas

Listas enlazadas: definición, representación de datos en memoria, ventajas, costos, operaciones, manejo de memoria, implementación, consideraciones sobre la depuración de programas. Listas ordenadas. Listas circulares. Listas doblemente enlazadas. Implementación usando listas enlazadas. Aplicaciones.

Unidad 6: Contenedores no lineales: árboles y grafos

Grafos: definiciones previas. Propiedades. Clasificación. Representación/implementación. Algoritmos de análisis y manipulación. Problemas de grafos: Camino más corto, Arbol mínimo, Flujo máximo, Viajante de comercio, Coloreo, Apareo. Aplicaciones. Árboles: definición, terminología, características, propiedades, operaciones. Árboles binarios. Árboles binarios de búsqueda. Árboles balanceados. Árboles n-arios. Implementación de árboles. Aplicaciones.

Unidad 7: Técnicas de diseño de algoritmos

Técnicas clásicas de diseño de algoritmos: Divide y reinarás, Algoritmos greedy, Backtracking y Programación dinámica. Definiciones. Características. Aplicaciones. Consideraciones sobre prueba y depuración de programas. Pruebas sobre estructuras de control de programas. Pruebas unitarias. Manejo de excepciones. Manejo de eventos. Introducción a la programación concurrente.

2) *Programa de Trabajos Prácticos*: En Tabla I se muestra el listado de trabajos prácticos, el cual incluye los temas cubiertos en cada uno de ellos y el tiempo dedicado (en horas) en clases prácticas.

3) *Metodología de Trabajo*: Los contenidos curriculares se desarrollan en clase teórica y práctica. En clase teórica, mediante la proyección de diapositivas, uso del pizarrón y software se tratan contenidos teóricos y ejemplos prácticos. El uso de estas herramientas permite un mejor afianzamiento de los temas, la implementación de ejemplos y ejecución de los mismos en clase. Así también, se fomenta el intercambio y el debate de ideas, necesarios para que el estudiante pueda entender conceptos desde diferentes ópticas, tal como lo plantea Bruner[3] en el postulado perspectivista. La clase práctica se desarrolla en laboratorio. En la misma se pretende que el estudiante consolide la lógica y la abstracción

TABLE I
DESARROLLO DE TRABAJOS PRÁCTICOS

TP	Temas incluidos	Horas
1	Un. 1: Divisibilidad y números primos	6
2	Un. 1: Máximo común divisor y mínimo común múltiplo	6
3	Un. 1: Aritmética modular	6
4	Un. 2: Recursividad	4
5	Un. 3: TAD	4
6	Un. 4: Contenedores lineales: pilas y colas	4
7	Un. 5: Contenedores lineales: listas enlazadas	8
8	Un. 6: Contenedores no lineales: árboles y grafos	14
9	Un. 7: Técnica de diseño de algoritmos	4

necesaria para que pueda afrontar de manera supervisada y a posterior en forma autónoma, la resolución de problemas computacionales de diferente índole y complejidad. Para ello, el docente desarrolla ejemplos prácticos usando pizarrón y supervisa en el desarrollo de los ejercicios propuestos en las guías de trabajos prácticos a través del uso de la computadora. Desde Contenedores Lineales en adelante, los estudiantes disponen de una librería propia desarrollada (en Pascal; a partir de 2014, en Java) y ofrecida por la Cátedra. Este recurso o modo de mediación, según Wertsch[13] permite que se puedan desarrollar soluciones a problemas computacionales en menos tiempo (aumentando así la productividad[1]); es extensible, sirve como material de estudio y de referencia para el desarrollo de software.

Tanto en clase teórica como en práctica, el manejo de la autoridad de parte del Docente para un mejor desarrollo de la motivación por el aprendizaje (modelo TARGET indicado por Huertas et al.[6]) es el correspondiente a un profesor democrático; es decir, se intenta ser facilitador de la realización de las tareas, promoviendo también la participación de los estudiantes en la toma de decisión relativas a las mismas así como en las responsabilidades inherentes, siempre bajo un cierto grado de control.

A partir de 2013, los estudiantes desarrollan un Trabajo Práctico Final grupal. Con esto, se pretende que apliquen los conocimientos aprendidos a diferentes situaciones reales, específicas y novedosas (buscando así capturar la atención de ellos[6]) como así también, se fomenta el trabajo en equipo y que entre compañeros se ayuden a aprender (de acuerdo al postulado interaccional de Bruner[3]). Para ello, al comienzo de la segunda mitad del cuatrimestre, cada docente de la Cátedra propone un tema diferente de complejidad media y realiza una instrucción teórico-práctico explícita del mismo; La propuesta de múltiples temas para el trabajo final grupal adhiere a lo indicado en el parámetro tarea del modelo TARGET [6], mientras que la instrucción explícita resulta fundamental para promover la metacognición en el estudiante, según Mateos[8]. Luego, en horarios de consulta y a través de los foros disponibles en el sitio Moodle de la asignatura, los docentes guían y supervisan (es decir, se aplica la práctica guiada propuesta por Mateos[8]) el desarrollo por parte de los grupos a su cargo. Teniendo en cuenta que se realiza

en simultáneo con el desarrollo de las clases y analizando la calidad de los trabajos presentados y la opinión de los estudiantes, la actividad es muy satisfactoria y productiva. Es importante mencionar algunos aspectos que hacen que esta actividad sea enriquecedora. En primer lugar, se intenta acercar a los estudiantes a una práctica profesional a través de problemas de interés real y actual (*pequeñas obras*, como indica Bruner[3] en el postulado de la externalización). En segundo lugar, la práctica guiada es una actividad que permite a los estudiantes aprender según sus tiempos y modalidades, pero con la supervisión del docente. En tercer lugar, se fomenta la práctica cooperativa [8] fundamental para su práctica profesional, donde el desarrollo de software involucra el trabajo en equipo y que es necesario practicarlo fuertemente a lo largo de la carrera. Respecto a los temas tratados en los trabajos prácticos finales debemos mencionar: compresión de datos sin pérdida (Huffman), evaluador de expresiones matemáticas, detección de bordes en imágenes digitales, criptografía (RSA y DES), String Matching, Auto-suggest, Sudoku (resuelto por backtracking), revisor ortográfico, entre otros.

Desde 2014 se realiza un taller de JAVA. Se desarrolla en horario extra-áulico por un período de 5 clases y al inicio del cuatrimestre. La motivación de esta práctica docente es la de introducir al estudiante de LAS en dicho lenguaje y en el paradigma orientado a objetos. De esta manera, se busca equiparar saberes y experiencias respecto de los estudiantes de TUP quienes cursan una asignatura específica (POO). Además, y como se observa en el programa analítico, dichos saberes no están incluidos pero son necesarios desarrollarlos, dadas las ventajas que ofrecen los mismos en la actualidad y la conveniencia de integrarlos a los contenidos desarrollados en la materia.

A partir de 2012, la Cátedra utiliza la plataforma Moodle. El sitio web de la Cátedra funciona como repositorio de material (guías de trabajos prácticos, apuntes, exámenes parciales y finales de cursadas anteriores, herramientas, software, entre otros) y como medio de información y consulta para los estudiantes. Además, dispone de 2 foros, los cuales son de uso frecuente para responder consultas en común realizadas por estudiantes y para comunicación entre ellos. Es importante destacar que los estudiantes de AyED tienen experiencia previa con el uso de la herramienta. De esta manera, se dispone de otro medio de comunicación. En este caso, virtual.

En cuanto a las consultas presenciales, se procura que las mismas se realicen en laboratorio. Esto permite que el Docente pueda explicar un tema específico a un grupo de alumnos que tengan las mismas consultas, como así también que actúe como tutor de varios estudiantes sobre problemáticas diferentes.

C. Metodología de Evaluación

Las pruebas de evaluación de los aprendizajes son valiosas si nos permiten conocer la manera y el grado de apropiación que los estudiantes han realizado de un conocimiento considerado importante[4]. En ese sentido, se evalúa al estudiante en dos momentos puntuales durante el cursado: al finalizar la

Unidad 2 (Recursividad) y al finalizar la Unidad 7 (Técnicas de diseño de algoritmos). En el primer momento, se evalúa a través de un examen parcial mientras que en el segundo, mediante un examen parcial y un Trabajo Práctico Final. Posterior al cursado, por medio de un examen final que abarca todos los contenidos curriculares del programa analítico vigente.

La evaluación de exámenes parciales y trabajos finales es procesual (tal como sugiere el parámetro Evaluación del modelo Target [6]). En los exámenes parciales, se evalúan los temas incluidos, con ejercicios fragmentados y evitando solapamiento de contenidos curriculares. Además, se intenta que los ejercicios sean de aplicación de conocimientos (poniéndose de manifiesto el interés por el aprendizaje significativo y no el maquinal), que no sean repetitivos y con consignas claras y concisas. Pero también, que sean similares a los propuestos en las guías de trabajos prácticos en cuanto a consigna y esfuerzo de desarrollo. La duración de los exámenes parciales es de 3 horas. En cuanto al espacio físico, se realiza en una única aula y todo el plantel docente está presente. Esto permite reducir los tiempos de entrega y recolección de parciales. Además, facilita al momento de comunicar aclaraciones u observaciones a los estudiantes sobre ciertas consignas del examen. Por otra parte, el plantel docente practica la *evaluación dinámica*[4]; esto es, circular en ciertos momentos por los pasillos, observar el desarrollo de los exámenes y brindar algunas pistas que sean disparadores para que los estudiantes puedan sortear dificultades presentes durante la resolución de los ejercicios. Respecto a los trabajos finales, se evalúa lo desarrollado por cada integrante como así también la interacción con el resto del grupo. Además, se permite la re-entrega de manera de mejorar la calidad de los mismos.

Algo a destacar es el reconocimiento de lo desarrollado por el estudiante con lineamientos similares a los indicados en el modelo Target. Las evaluaciones de los exámenes incluyen anotaciones de los docentes que indican si lo desarrollado es correcto o en caso contrario, lo que es incorrecto y el por qué. Esto permite que los estudiantes analicen sus exámenes y consulten sobre su producción. De manera similar, la evaluación del Trabajo Final también incluye un momento para indicar aspectos a mejorar o destacar lo realizado. En ambos casos, el elogio del esfuerzo y progreso como así también las recomendaciones para superar saberes no aprendidos, son importantes para la motivación del estudiante y ocurren en privado.

Para regularizar la Asignatura, el estudiante debe simultáneamente:

- Aprobar cada uno de los dos exámenes parciales o sus respectivas recuperaciones, con nota mayor o igual a 60/100 o el examen global, en caso que haya tenido que recuperar ambos.
- Aprobar el Trabajo Práctico Final propuesto por la Cátedra.

Para promocionar la Asignatura:

- Los estudiantes regulares, deben rendir un examen oral o

escrito sobre los contenidos del programa vigente.

- Los estudiantes libres, deben aprobar un examen práctico, y luego rendir el examen teórico.

D. Análisis de porcentajes de regularidad

La práctica docente desarrollada en la asignatura ha sido detallada en la sección anterior (período 2011) y en subsecciones previas (período 2012-2017). Respecto a este último período, debemos destacar dos momentos claramente diferenciados por la variedad de acciones realizadas: de transición (2012-2013) y de consolidación (2014-2017). Por otra parte, la migración del lenguaje Pascal a Java (año 2014) y el dictado en simultáneo para las carreras LAS y TUP (2013), son eventos cruciales en la breve historia de Algoritmos y Estructuras de Datos.

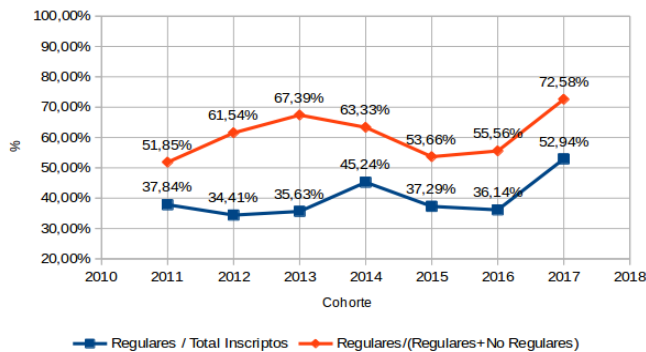


Fig. 1. Resultados por Cohorte

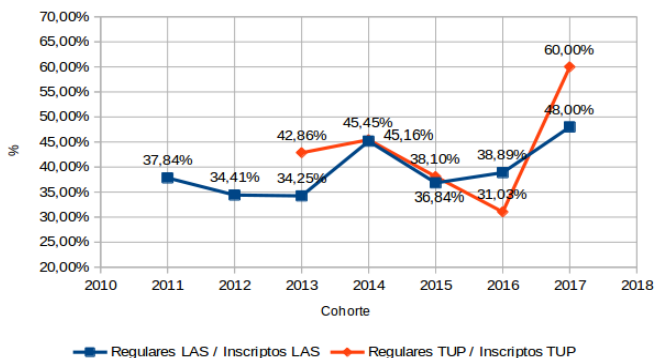


Fig. 2. Resultados por Carrera y Cohorte

En la Fig. 1 se muestran los porcentajes de regularidad por Cohorte. La gráfica de color azul muestra porcentajes de regularidad sobre el total de estudiantes inscriptos, mientras que la de color rojo, sobre el total de estudiantes regulares y no-regulares. Si se analizan los valores año a año, estos no muestran grandes cambios mas allá de la población estudiantil inscripta (a excepción de la Cohorte 2017). En promedio, el porcentaje de regularidad sobre el total de inscriptos es del 39.7%, en tanto que respecto del total de regulares y no regulares es del 61.9%. Respecto a otros estudios realizados sobre cátedras con contenidos similares, nuestros resultados

son alentadores. La Red Martinez et al.[7] informan porcentajes de regularidad cercanos al 26%, en tanto que Szpiniak y Rojo[11], 30.5%.

En la Fig. 2 se muestran los porcentajes de regularidad (total regulares/total inscriptos) discriminados por carrera. Aquí se puede apreciar el rendimiento creciente de los estudiantes de LAS (gráfico color azul) y decreciente (gráfico color rojo) de los estudiantes de TUP, pero con un importante repunte durante 2017. Los porcentajes de regularidad de LAS (en promedio, 38.3%) son similares a los de TUP (en promedio, 44.5%).

V. MEJORAS A IMPLEMENTAR EN LA ASIGNATURA

En esta sección detallaremos algunas acciones que se aplicaron durante el primer cuatrimestre de 2017, como así también mejoras a futuro. En ambos casos, estas propuestas surgen como inquietud de la Cátedra en pos de mejorar el proceso de enseñanza-aprendizaje de manera continua y evolutiva.

La manera de transferir conocimientos y experiencias, depende en gran parte de la formación académica y experiencia laboral del Docente. Una mejor formación académica contribuirá a motivar mejor a sus estudiantes, motivándose primero él mismo (evitando el nopodernimiento, indicado por Abramowsky[1]). Desde la Cátedra, se brindará el apoyo necesario para que todo el plantel tenga formación de posgrado en Ciencias Informáticas o de la Computación. Actualmente, el Docente Responsable de la asignatura y 1 Jefe de Trabajos Prácticos tienen título de Doctor en Ciencias de la Computación.

Si bien los porcentajes de regularidad son adecuados, es importante reducir los niveles de deserción (en promedio, cercanos al 35%). Esto significa, reducir el número de estudiantes que abandonan el cursado. Una acción superadora es la de evaluar al estudiante recursante sobre los saberes no alcanzados. Esta acción ha sido implementada durante el último cuatrimestre, exceptuando de presentar un nuevo Trabajo Práctico Final a aquellos que lo aprobaron anteriormente. Para el próximo año se eximirá a los estudiantes recursantes de rendir el primer examen parcial, si lo aprobaron en la cursada anterior. Esta decisión se sustenta en que los contenidos más relevantes de la asignatura se sitúan a partir de la segunda mitad del cuatrimestre.

Otra acción que ha sido implementada (por ahora, parcialmente) durante el primer cuatrimestre de 2017 está relacionada con la *práctica guiada*. Siguiendo el método propuesto por Schoenfeld[8], los Jefes de Trabajos Prácticos en sus respectivas comisiones y en laboratorio⁹, piden a sus estudiantes que participen de manera colectiva¹⁰ en la resolución de un problema computacional, de manera similar a lo indicado por Gallego y Gortázar[5] en cuanto al uso de EclipseGavab. En nuestro caso, en clases prácticas se ha usado el plugin de

⁹En los últimos años, ambas comisiones de trabajos prácticos tienen aproximadamente 30-35 estudiantes quienes asisten regularmente a clases.

¹⁰Soler y Lezcano [10] proponen el uso de Cmaptools para tratar el tema TAD.

Eclipse Saros¹¹ el cual tuvo una muy buena recepción en los estudiantes. Es importante destacar que una acción de este tipo no se ha llevado a cabo previamente en ninguna de las dos carreras informáticas que se dictan en el Departamento de Informática. Por otra parte, se auguran excelentes resultados para el próximo año debido al impacto que esto genera: fomentar la práctica guiada, cooperativa y con un buen uso de las TICS.

Otra mejora realizada en el cuatrimestre anterior, ha sido el uso de guías de lectura. Por el momento, las mismas se usaron para abordar las técnicas de diseño de algoritmos (de la Unidad 7 del programa). Consideramos al uso de este recurso una medida positiva por varias razones. En primer lugar, acerca al estudiante a la lectura de un libro, de manera de esquivar el mito de la autosuficiencia de la clase[12]. En segundo lugar, las guías propuestas hacen referencia a algoritmos antes tratados en clase, lo cual favorece al aprendizaje significativo. Por último, las guías acompañan al estudiante en su aprendizaje autónomo.

VI. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se han presentado diferentes acciones implementadas en los últimos años en la asignatura Algoritmos y Estructuras de Datos que se dicta para las carreras Licenciatura en Análisis de Sistemas y Tecnicatura Universitaria en Programación del Departamento de Informática de la UNSa.

A continuación, enumeramos algunos aspectos que consideramos fundamentales en el mejoramiento general de la asignatura:

- Contenidos curriculares
 - Articulados en forma horizontal y vertical.
 - Organizados por contenidos comunes y desarrollados en forma gradual, promoviendo el aprendizaje significativo.
 - Los contenidos no desarrollados durante el cursado deben ser abordados en asignaturas optativas posteriores.
- Metodología de trabajo
 - Clases teórico-prácticas a cargo del Docente Responsable con fuerte uso de tecnología.
 - Clases prácticas con uso intensivo de la computadora.
 - Taller extra-curricular (por ejemplo, para manejo de tecnología a aplicar durante el cursado).
 - Trabajo final integrador supervisado por los docentes.
 - Consultas presenciales y virtuales a cargo de la Cátedra.
 - Discusión y debate de los contenidos vistos en clase. Participación activa del estudiante.
- Tecnología
 - Fomento del software libre.
 - Lenguaje de programación acorde para aplicar los contenidos vistos en clase.

- Framework de desarrollo que facilite la programación y el testing de programas.
- Herramienta colaborativa (plugin, librería) para el desarrollo de contenidos en clase y para el trabajo en equipo por parte de los estudiantes.
- Librería extensible que sirva de marco de referencia y como herramienta para aumentar la productividad en el estudiante.
- Plataforma virtual educativa como herramienta de soporte.

- Recursos didácticos

- Apuntes (de temas específicos, de clase, para el trabajo final).
- Código fuente de ejercicios propuestos en las guías de trabajos prácticos.
- Guías de lectura.
- Papers, tesis (para el trabajo final).

- Metodología de evaluación

- Evaluación en forma procesual, no maquina.
- Evaluación individual y grupal.
- Evaluación de contenidos teóricos, prácticos y de experiencias.
- Valorar en el estudiante la aplicación del conocimiento adquirido y la búsqueda de nuevas soluciones a problemas vistos en clase.

- Plantel docente

- Docentes con dedicación horaria acorde.
- Preferentemente con formación de posgrado e investigación en temáticas afines a la asignatura.
- Organizado como equipo y no de manera vertical.

Todas las acciones implementadas estuvieron orientadas a enseñar los contenidos de la asignatura de manera completa, organizada, actualizada y fundamentalmente, motivadora. Es decir, se mejoró en contenidos curriculares (mejor integración y profundidad de los contenidos desarrollados con uso intensivo de la Programación Orientada a Objetos), en tecnología (por ejemplo, la migración de Pascal a Java¹²) y TICS, en evaluación de saberes y aplicación de los mismos (Trabajo Práctico Final). Todo esto permitió que nuestros estudiantes adquieran conocimientos y experiencias de manera sólida, sin resentir los porcentajes de regularidad y con un alto nivel de aprobación sobre las acciones llevadas a cabo por la Cátedra, según se observa en las encuestas estudiantiles.

Sin embargo, aún queda mucho por hacer más allá de las acciones que se han comenzado a implementar durante el primer cuatrimestre de 2017. Mejorar el porcentaje de regularidad es una de las tareas pendientes. Para ello, se está desarrollando una tesis de posgrado que analiza la práctica docente llevada a cabo por la Cátedra de manera de mejorar los valores actuales. Por otra parte, el abandono estudiantil en el segundo año de ambas carreras es alto, por lo que merece también un mejor estudio de dicho fenómeno y un tratamiento

¹¹<http://www.saros-project.org/>

¹²Szpiniak y Rojo[11] tratan una situación similar de migración.

acorde. Todo esto en pos de formar mejores estudiantes y futuros profesionales.

ACKNOWLEDGEMENT

Los autores agradecen las observaciones y sugerencias realizadas por los evaluadores las cuales ayudaron a mejorar la calidad del trabajo presentado.

El trabajo ha sido parcialmente financiado por Proyecto CIUNSa 2364 y Proyecto Redes IX 46-148-237.

REFERENCES

- [1] A. Abramowsky, *Variaciones del nopodermiento escolar: perezas y desmotivaciones en Interés, motivación y deseo*, 1ra ed. Buenos Aires, Argentina: Noveduc, 2007.
- [2] G. Bachelard, *La formación del espíritu científico. Contribuciones a un psicoanálisis del conocimiento objetivo*, 1ra ed. DF, México: Siglo XXI, 1999.
- [3] J. Bruner, *La educación, puerta de la cultura*, 1ra ed. Madrid, España: Visor, 1997.
- [4] S. Celman, *Es posible mejorar la evaluación y transformarla en herramienta de conocimiento?* en *La evaluación de los aprendizajes en el debate didáctico contemporáneo*, Buenos Aires: Argentina: Paidós, 1998.
- [5] M. Gallego y F. Gortázar, *EclipseGavab, un entorno de desarrollo para la docencia online de la programación* Trabajos de las XV JENUI, Barcelona, 1–8, 2009.
- [6] J. Huertas et al., *Principios para la intervención motivacional en el aula en Motivación. Querer aprender*, 1ra ed. Buenos Argentina: Aique, 1996.
- [7] D. La Red Martínez, M. Giovannini, N. Pinto, M. Frissone y M. Báez, *Determinación de Perfiles de Rendimiento Académico en la UTN-FRR* Trabajos del III CONAIIISI, Buenos Aires, 20–28, 2015.
- [8] M. Mateos, *Metacognición y educación*, 1ra ed. Buenos Aires, Argentina: Aique, 2001.
- [9] J. Pozo, *Teorías cognitivas del aprendizaje*, 1ra ed. Madrid, España: Morata, 2001.
- [10] Y. Soler y M. Lezcano, *Consideraciones sobre la tecnología educativa en el proceso de enseñanza-aprendizaje. Una experiencia en la asignatura Estructura de Datos* Revista Iberoamericana de Educación, 49-2, 1–9, 2009.
- [11] A. Szpiniak y G. Rojo, *Enseñanza de la programación* Revista TE&ET, Nro. 1 Vol. 1, 71–78, 2006.
- [12] J. Steiman, *Qué y cómo enseña la Universidad Es un problema que tiene que plantearse el docente universitario?* Trabajos de las Primeras Jornadas de Pedagogía Universitaria, Buenos Aires, 10–20, 2009.
- [13] J. Wertsch, *Propiedad de la acción mediada en La mente en acción*, 1ra ed. Buenos Aires, Argentina: Aique, 1999.