

Avances en la Práctica Docente de Algoritmos y Estructuras de Datos

Cristian A. Martinez, Diego A. Rodriguez, Carlos I. Orozco

Departamento de Informática – Universidad Nacional de Salta (UNSa)
Av. Bolivia 5051 - 4400 - Salta – Argentina

cmartinez@unsa.edu.ar, drodriguez@di.unsa.edu.ar,
iorozco@exa.unsa.edu.ar

***Abstract.** Algorithms and Data Structures is a subject of the Universidad Nacional de Salta that includes topics related to Mathematics and Programming necessary for students in their careers and professions. Our proposal is oriented to study, analyze and offer improvements on teaching practice of this subject in order to the teaching-learning process be efficient.*

***Resumen.** Algoritmos y Estructuras de Datos se dicta para dos carreras de la Universidad Nacional de Salta. Desarrolla fuertes contenidos de Programación y de Matemática que los estudiantes necesitan aprender y aplicarlos a lo largo de la carrera, como así también en el ámbito laboral. Nuestra propuesta está enfocada en estudiar, analizar y ofrecer mejoras en la práctica docente de dicha asignatura. A lo largo del trabajo se detallarán diferentes actividades que son fundamentales para que el proceso de enseñanza-aprendizaje sea eficiente.*

1. Introducción

La Comisión Nacional de Evaluación y Acreditación Universitaria (CONEAU) 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 Licenciatura en Análisis de Sistemas de la Universidad Nacional de Salta fue evaluada por la CONEAU en 3 oportunidades: en 2010, 2014 y 2018. Como resultado, el Departamento de Informática presentó un nuevo plan de estudios en conjunto con la Comisión de Carrera, como así también se llevaron a cabo acciones para mejorar en lo referente a equipamiento, bibliografía, infraestructura, formación y formación docente, como así también el rendimiento académico para alcanzar las respectivas acreditaciones.

Pertenciente al área de Algoritmos y Lenguajes, Algoritmos y Estructuras de Datos se dicta para las carreras de Licenciatura en Análisis de Sistemas y Tecnicatura Universitaria en Programación, tanto en el Departamento de Informática de la Facultad de Ciencias Exactas Sede Central como en Sede Regional Orán.

En este trabajo trataremos las acciones realizadas en la asignatura. Nuestra motivación consiste en mostrar avances en enseñanza y evaluación de los contenidos curriculares; en cómo se trabajó respecto a la articulación horizontal y vertical con otras asignaturas. Además, indicaremos las acciones que se están llevando a cabo y las que se quieren poner en práctica, de manera de continuar mejorando el proceso de enseñanza-aprendizaje.

El resto del trabajo se organiza de la siguiente manera: la sección 2 revisa el estado del arte. La sección 3 describe aspectos de la asignatura. En la sección 4 se analizan datos relacionados con la asignatura y su práctica docente. La sección 5 presenta la metodología de trabajo, contenidos curriculares y evaluación que se lleva a cabo en la actualidad. Finalmente, las conclusiones en la sección 6.

2. Estado del Arte

A continuación, listaremos algunos trabajos de asignaturas con temáticas similares y la práctica docente llevada a cabo.

Ferreira Szpiniak y Rojo (2006) muestran los resultados obtenidos al aplicar un nuevo enfoque en la enseñanza de la Programación en asignaturas de primer año de las carreras de Computación en la Universidad Nacional de Rio Cuarto. Proponen enseñar programación mediante el uso de máquinas abstractas y el lenguaje Pascal, para que el estudiante se inicie en la formalización de algoritmos y a continuación, los pueda programar mediante un lenguaje sencillo y a la vez, estructurado y eficiente. La nueva forma de trabajo permitió aumentar los porcentajes de regularidad del 20% en 2001, a 31% en 2004 y 30% en 2005.

Moreno y otros (2013) proponen el trabajo colaborativo para mejorar el proceso de enseñanza-aprendizaje en los cursos iniciales de Programación. En su trabajo, hacen una revisión de recursos tecnológicos que lo fomentan (Dropbox, Google Drive, Google Code, WebCT, Habi-Pro) y cómo se pueden combinar a través de una plataforma educativa (como Moodle) para llevar a cabo la práctica docente en temas relacionados con la programación.

Arellano y otros (2014) presentan la experiencia realizada en el primer curso de Programación para las carreras de Ingeniería Electrónica de la Universidad Nacional de San Luis. Proponen el uso de TIMBA para introducir al estudiante en la noción de algoritmo, DIA para desarrollar diagramas de flujo, Pseint para que los estudiantes se inicien en la construcción de programas sin tener que lidiar con la sintaxis de los lenguajes de programación, y Moodle para ofrecer un canal de comunicación con los estudiantes.

En su tesis, Salazar (2015) propone un sitio web y estrategias didácticas para la enseñanza de la Programación mediante TICS para los estudiantes de la asignatura Informática de las carreras de Ingeniería Eléctrica, Electrónica y Agrimensura de la Universidad Nacional del Nordeste. Desarrolla un sitio web que incluye compilación de código C en línea, foro, chat con el plantel docente y recursos. En cuanto a lo didáctico, propone que los estudiantes trabajen de a pares para resolver ejercicios dispuestos por la Cátedra dentro del sitio web, disponiendo también de las devoluciones de parte de los docentes.

Soler Pellicer y otros (2018) presentan un ambiente visual para la enseñanza de Estructura de Datos de la Ingeniería Informática de la Universidad de Granma e Ingeniería en Sistemas de la Universidad Estatal del Sur de Manabí. Mediante mapas conceptuales, visualizan estructuras de datos con ejemplos para cada caso. Para los ejemplos, se muestra la complejidad del programa y ofrece la posibilidad de trazabilidad para las soluciones recursivas. Presentan una encuesta validada sobre la aceptabilidad de la nueva propuesta.

3. Algoritmos y Estructuras de Datos (AyED)

AyED se dicta para la Licenciatura en Análisis de Sistemas (LAS) Plan 2010 y Tecnicatura Universitaria en Programación (TUP) Plan 2012. Para LAS, desde el año 2011, en tanto que para TUP desde 2013. En ambas, se ubica en el primer cuatrimestre del segundo año, con una carga semanal de 8 horas (4 horas de teoría y 4 horas de práctica). En términos de carga horaria, cubre el 24% del total asignado al área de Algoritmos y Lenguajes en la carrera de grado, y el 12% para la misma área en la de pregrado. En base al análisis del régimen de correlatividades de ambos planes de estudio, la asignatura se relaciona con:

- LAS
 - 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
 - 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 (comunes a ambos planes) son: Teoría de las Estructuras Discretas. Definiciones y pruebas estructurales. Teoría de 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. Estrategias de diseño de algoritmos.

De acuerdo a los contenidos mínimos, se detecta un gran desarrollo teórico-práctico en Matemática y Programación. Por otra parte, de los regímenes de correlatividades, se aprecia una fuerte relación curricular con asignaturas de segundo y tercer año.

4. Estudio y Análisis de la Práctica Docente

Para la presentación de propuestas de mejora, fue necesario recolectar, procesar, analizar y estudiar datos de la asignatura en ambos planes de estudios. En particular, las fuentes fueron planes de estudios, programas analíticos, exámenes parciales, datos censales y de rendimiento académico, encuestas, entrevistas y observación externa.

4.1. Sobre la metodología de trabajo y el programa analítico en los inicios

Se indicarán aspectos relacionados con contenidos curriculares y metodología de trabajo propuestos en los inicios de la asignatura¹. En su primera etapa (año 2011), la forma de trabajo era muy similar a la realizada en Algorítmica, Estructuras de Datos y Matemática para Informática de la Licenciatura en Análisis de Sistemas Plan 1997: un enfoque matemático fuerte para los contenidos de Teoría de Números y un tratamiento

¹ El programa se encuentra disponible en <http://bo.unsa.edu.ar/cdex/R2011/R-CDEX-2011-179.PDF>

numérico-gráfico respecto a los demás temas del programa analítico (Contenedores Lineales, Grafos y Arboles). Por otra parte, las clases teóricas y prácticas se desarrollaban en aula y con uso de pizarrón.

En relación a los contenidos curriculares, se observa una gran cantidad de temas a abordar, con cierta descoordinación entre ellos (Concurrencia y Diseño de algoritmos son desarrollados al comienzo del cursado) y además, poca profundidad en el tratamiento de los mismos (como ocurre con los temas Contenedores Lineales, Grafos y Arboles, que requieren de mucha práctica en computadora).

4.2. Sobre las encuestas

Se realizaron 3 encuestas a estudiantes que cursaron la asignatura en 2017. La primera, para recolectar datos personales de índole socio-económico, familiar, de conocimientos previos, entre otros, los cuales se usaron para un estudio mediante técnicas de Minería de Datos, Arboles de Decisión y Redes Neuronales. La segunda, se enfocó en los conocimientos adquiridos por el estudiante de los temas incluidos en el primer examen parcial, como así también la valoración de la práctica docente del Responsable y de los Jefes de Trabajos Prácticos, uso de bibliografía, consulta, y el taller de JAVA que se realiza de manera simultánea al cursado. La tercera, se realizó para conocer los conocimientos adquiridos por el estudiante, la valoración de la práctica docente del Responsable y de los Jefes de Trabajos Prácticos, del trabajo práctico final, uso de TICS (ver sección 5 para más detalles), entre otros. Se realizó la clase previa al examen parcial y la completaron 28 estudiantes de manera voluntaria. Algunos resultados de la tercera encuesta se listan en Tablas 1, 2 y 3.

Tabla 1. 3ra encuesta: datos recolectados (en promedio) del estudiante

Ítem	Resultado
Materias cursadas incluida AyED	3
Horas de estudio para el parcial	9.2
Valoración (1-10) del conocimiento de temas del parcial	6.5
Porcentaje de desarrollo de los TPs incluidos en parcial	47%
Porcentaje de asistencia a clases teóricas	93%
Porcentaje de asistencia a clases practicas	88%

Tabla 2: 3ra encuesta: datos (en promedio) sobre la práctica docente

Ítem	Resultado
Valoración (1-10) de los conocimientos del Responsable sobre temas del parcial	9.7
Valoración (1-10) de la enseñanza del Responsable sobre temas del parcial	9.4
Valoración (1-10) de los conocimientos del JTP sobre temas del parcial	8.4
Valoración (1-10) de la forma de enseñar del JTP sobre temas del parcial	9.1
Valoración (1-10) sobre la interacción del plantel docente	8.9
Valoración (1-10) del tiempo dedicado por la Cátedra a los temas del parcial	7.2

Tabla 3: 3ra encuesta: datos recolectados (en promedio) sobre acciones propuestas por la Cátedra

Ítem	Resultado
Valoración (1-10) del TP Final	8.6
Valoración (1-10) del uso de SAROS	8.8
Valoración (1-10) del uso de Linux	8.8
Valoración (1-10) de lo aprendido durante el Cursado	7.9
Valoración (1-10) de los contenidos curriculares para su profesión	9.3

De las tablas se aprecia que al final del cuatrimestre los estudiantes cursan 3 asignaturas (incluyendo AyED), dedican (hasta el momento de la encuesta) 9.2 horas de estudio para el examen, sus conocimientos son adecuados pero el desarrollo de los trabajos prácticos es bajo (47%). La asistencia es alta (cerca del 90%). La valoración que realizaron sobre los conocimientos y la práctica docente del plantel docente también es alta (por encima de 8).

Cuando se consultó sobre ciertos aspectos de la práctica docente, la respuesta es positiva: el trabajo práctico final tiene alta aprobación (8.6 sobre 10), al igual que el uso de SAROS como herramienta colaborativa de desarrollo (8.8) y Linux como sistema operativo (en vez de Windows) de base. Asimismo, consideran relevante lo aprendido durante el Cursado (7.9), en tanto que el tiempo destinado para los temas del segundo parcial es adecuado (7.2), aunque consideran necesario una mayor carga horaria.

4.3. Sobre las entrevistas

Se diseñaron 2 entrevistas dirigidas a estudiantes para conocer de manera directa las opiniones sobre la práctica docente. La primera, dirigida a aquellos que no regularizaron (no pudieron aprobar los 2 exámenes o sus recuperatorios) y la segunda para los que alcanzaron la regularidad al finalizar el cursado. Por falta de tiempo, se les solicitó a dos estudiantes que cursaron en 2017 que envíen sus respuestas por correo electrónico.

Respecto al estudiante que no regularizó la materia, nos interesaba conocer sus deficiencias en el aprendizaje como así también las de la Cátedra. De sus respuestas se puede apreciar que estuvieron principalmente enfocadas en su imposibilidad por aprender el lenguaje JAVA en tiempo y forma. A pesar de haber asistido al taller, no pudo superar ciertas deficiencias en el manejo del lenguaje impidiéndole entender los contenidos curriculares de la materia los cuales se basan en este lenguaje. Finaliza el cuestionario indicando que aprenderá más sobre el lenguaje para que no sea un impedimento durante la próxima cursada.

En cuanto a las respuestas brindadas por el segundo estudiante, enfatiza haber aplicado sus conocimientos en Programación Numérica y Programación Orientada a Objetos, las cuales no son correlativas con AyED. Por otra parte, valora el trabajo en grupo y el desarrollo del trabajo práctico final. Asimismo, considera que la asignatura Análisis Matemático I no debería ser correlativa de la materia. Finalmente, sugiere a los futuros estudiantes que presten atención en clase y consulten sus dudas en el momento que ocurran.

4.4. Sobre la observación externa

El registro de clase práctica fue llevado a cabo por una docente externa del Departamento. El tema de la clase fue Algoritmo de Kruskal y TSP (Unidad 6 del programa analítico vigente). La clase inició a las 16:00 y finalizó a las 18:00 horas. Asistieron 21 estudiantes y el docente contó con el apoyo del auxiliar docente. A continuación, se indican algunos de los aspectos más relevantes del registro de clase:

- Acerca de la motivación: la introducción de la clase tiene relación con el tema propuesto. Para ello, el docente realiza un sondeo de conocimientos previos al tema a tratar. Transmite interés en la clase, presentando material de soporte y utilizados de manera correcta.
- Acerca de la clase: el docente explica los temas con claridad, siguiendo una secuencia lógica, enfatizando cuando es necesario. Hace referencia a contenidos anteriores para vincularlos, explicando los temas con ejemplos y ejercicios.
- Acerca de las habilidades pedagógicas del docente: el docente maneja vocabulario técnico, tiene buen manejo de los contenidos desarrollados y se muestra abierto a comentarios y preguntas de los estudiantes.
- Acerca del plan de clases: las consignas propuestas por el docente son claras y facilitan la tarea, propone actividades adecuadas para cada una de las fases de la clase con buen manejo del tiempo.
- Acerca de los estudiantes: manejan un vocabulario técnico acorde, se muestran disciplinados en clase, aunque su producción en clase es regular.

Análisis de clase: el docente divide la clase en dos. En la primera hora, plantea una clase expositiva para recordar y fijar conceptos vistos en clase teórica. En la segunda, trabaja con grupos pequeños, resolviendo una guía de trabajos prácticos, siendo pocas las intervenciones hacia la clase. Como fortalezas, motiva a sus estudiantes durante la clase, utiliza diferentes recursos para que aprenda los contenidos con un óptimo manejo del tema. Como debilidad, hizo un análisis excesivo del código fuente presentado a los estudiantes.

4.5. Sobre datos censales, becas y rendimiento académico de estudiantes de AyED

Esta sección describe el proceso aplicado para obtener información de relevancia sobre datos académicos, socio-económicos de nuestros estudiantes y de su grupo de familiar.

El dataset de estudio consistió de 469 registros referidos a estudiantes de LAS y TUP que cursaron la asignatura entre 2011 y 2017. Los datos provienen del SIU Guaraní (datos personales, de exámenes de asignaturas previas e información censal), de la Secretaría de Bienestar Universitario de la UNSa (datos de becas) y de la asignatura (notas de parciales). Los datos considerados para el estudio son: edad al inicio de la cursada, sexo, tipo de colegio secundario en que se graduó (privado o público), situación laboral del estudiante y de los padres, si cobraba alguna beca mientras cursaba la asignatura, condición en la materia (primer cursado o recursante), si aprobó la asignatura Matemática para Informática, si aprobó la asignatura Programación, si aprobó el primer parcial de la asignatura durante el cursado, situación al finalizar el cuatrimestre (regular, libre o abandonó).

Algunos resultados a nivel univariado/bivariado: el promedio de edad es 21.2 años, de los cuales el 80% son varones. Respecto al establecimiento secundario, el 58% provienen de instituciones públicas. El 83% de los estudiantes aprobaron Matemática

para Informática, y el 66% Programación. El 70% no trabaja, aunque el 22% admitió estar buscando trabajo. Además, el 8.5% fue beneficiario de alguna beca mientras cursaba la asignatura. Sobre sus progenitores, el 64% de los padres y el 60% de las madres completaron el secundario; sólo completaron estudios universitarios el 2.6% de los padres y el 4.7% de las madres. En cuanto a lo laboral, el 14.28% de los padres y el 31.77% de las madres están desocupados.

4.5.1 Aplicación de Redes Neuronales

Las Redes Neuronales han sido aplicadas en diferentes campos de estudio (Aggarwal, 2018). En este trabajo las usaremos para predecir el estado final de cursado de los alumnos (Regular, No regular, Abandonó). Para ello, proponemos el uso de arquitectura de red neuronal multicapa y un entrenamiento supervisado de ADAM para actualización de los pesos. Luego de varios experimentos empíricos, el modelo propuesto en el presente artículo está formado por 6 capas completamente conectadas. La función de activación empleada es RELU (rectified linear unit) en las 5 primeras capas y sigmoide en la capa de salida para garantizar que las salidas del modelo estén comprendidas en el rango [0, 1]. La cantidad de neuronas por cada capa del modelo es la siguiente: primera capa: 8000 neuronas, segunda: 90 neuronas, tercera: 62 neuronas, cuarta: 32 neuronas, quinta: 10 neuronas, y sexta: 3 neuronas.

El MLP fue implementado en Python 2.7.3, utilizando la librería Keras (2015) para el manejo de redes neuronales. Para evaluar la performance del clasificador, calculamos la precisión (Ec. 1) definida como las muestras clasificadas correctamente divididas por el número total de muestras del dataset. Esto es:

$$Precision = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Donde TP corresponde al número de verdaderos positivos detectados, TN a verdaderos negativos, FP a falsos positivos y FN a falsos negativos respectivamente.

Tabla 4: k-fold Cross validation con k=4. Cada iteración usa 3 fold para el entrenamiento, la restante para validación. La segunda columna muestra la precisión por cada fold.

K-fold	Precisión
1	72.84%
2	72.64%
3	73.02%
4	71.98%
Promedio	72.62%

Finalmente aplicamos k-fold Cross validation con k=4, dividiendo el dataset en 4 grupos de 117 muestras cada uno. La Tabla 4 muestra la precisión promedio luego de 10 corridas por cada fold. La precisión final del clasificador es de 72%, es decir que de 100 casos acierta en 72, un número aceptable considerando la cantidad de información recolectada. O sea, mediante el uso de Redes Neuronales aplicado a datos censales y de

rendimiento académico, se puede predecir la situación (al final del cursado) de un estudiante de Algoritmos y Estructuras de Datos con una precisión del 72.62%.

4.5.2 Aplicación de Árboles de Decisión

Los Árboles de Decisión son un conjunto de condiciones o reglas organizadas de manera jerárquica de manera tal que la decisión final se puede determinar siguiendo las condiciones desde la raíz del árbol hasta algunas de las hojas (Rokach y otros, 2014).

A los efectos de predecir (con otra técnica) la situación final de cursado, se propuso aplicar Árboles de Decisión. Para ello, se utilizó la librería *party* (Zhao, 2013) incluida en R versión 3.2.3 ejecutada sobre Linux. Para llevar a cabo el experimento numérico, se incluyeron todas las variables de estudio (indicadas previamente) de manera de obtener el mejor modelo posible, usando un 70% de los datos para entrenamiento y un 30% para pruebas. En Fig.1 se muestran los resultados alcanzados. De ellos, se destacan dos situaciones de interés: los estudiantes que aprobaron el primer parcial y cuya edad no supere los 23 años, tienen una probabilidad superior al 80% de regularizar la asignatura; por otra parte, los estudiantes que no aprobaron el primer parcial y no aprobaron el examen final de Programación, tienen una probabilidad superior al 70% de abandonar el cursado. Finalmente, de lo obtenido por los árboles de decisión, se aprecia que la situación de cursado del estudiante es independiente del entorno familiar como así también del establecimiento secundario de procedencia.

5. Propuesta de mejora de la Práctica Docente

Describiremos la metodología de trabajo de la Cátedra la cual se compone de un Profesor Adjunto con dedicación exclusiva, dos Jefes de Trabajos Prácticos con dedicación simple y exclusiva respectivamente, y un Auxiliar Docente de 2da categoría. Los cambios en la asignatura desde 2012 a la fecha, se sustentan en los siguientes ejes: articulación, desarrollo de contenidos y metodología de trabajo, y evaluación.

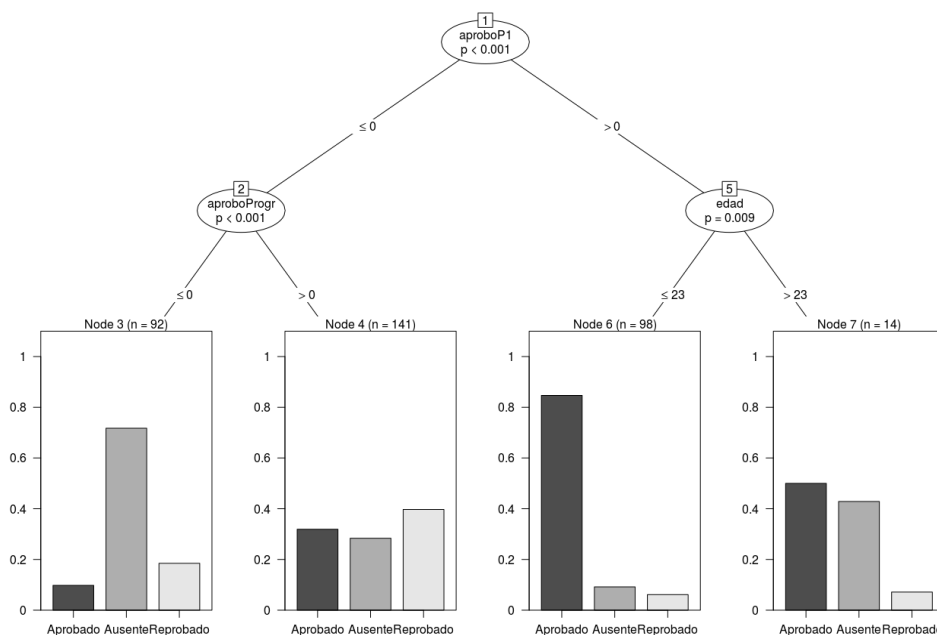


Figura 1: Árbol de decisión obtenido sobre el dataset mediante R

5.1 Articulación

Una de las acciones más sustanciales para la mejora de la práctica docente fue conocer y articular contenidos curriculares con las Cátedras con las cuales Algoritmos y Estructuras de Datos tienen relación directa. Esta acción se sustenta en el proceso de enseñanza-aprendizaje de Ausubel (Pozo, 2006) 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 (Mateos, 2001).

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.

En lo referente a la articulación horizontal, es importante mencionar la gran interacción 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.

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 Paradigmas y Lenguajes: Clase, Encapsulación, Reusabilidad, Recursión y Algoritmos concurrentes. También, permiten tratar los siguientes en 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 con Programación de Aplicaciones Web en: Programación del lado del cliente y del lado del servidor. Finalmente, en la asignatura Sistemas Operativos con: algoritmos de planificación de procesos, problemas clásicos de la comunicación entre procesos, administración de memoria con listas enlazadas, algoritmo de reemplazo de páginas óptimo, y algoritmo del banquero para uno y varios recursos.

5.2 Desarrollo de contenidos curriculares y metodología de trabajo

5.2.1 Programa Analítico

A partir de los conocimientos previos que el estudiante adquiere en Matemática para Informática y Programación, se desarrollan los contenidos de las Unidades 1, 2 y 3 (Matemática Discreta, Recursión y TAD). Por otra parte, los contenidos desarrollados en la Unidad 7 (Técnicas de diseño de algoritmos), son a partir de los algoritmos vistos

en la Unidad 6 (Contenedores no lineales). Con esta propuesta², se refuerza el aprendizaje significativo.

5.2.2 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. Estas herramientas permiten 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 (1997) en el postulado perspectivista.

La clase práctica se desarrolla en laboratorio. En ella se pretende que el estudiante consolide la lógica y la abstracción 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 desarrollada por la Cátedra. Este recurso permite desarrollar soluciones a problemas computacionales en menos tiempo (aumentando así la productividad (Abramowsky, 2007); es extensible, y sirve como material de estudio.

Desde 2013, los estudiantes desarrollan un Trabajo Práctico Final. Con esto, se pretende que apliquen los conocimientos aprendidos a diferentes situaciones reales, específicas y novedosas (buscando así capturar la atención de ellos (Huertas, 1996) como así también, se fomenta el trabajo en equipo y que entre compañeros se ayuden a aprender (de acuerdo al postulado interaccional de Bruner (1997). Para ello, al comienzo de la segunda mitad del cuatrimestre, los docentes proponen un tema de complejidad media y realizan 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 (Huertas, 1996), mientras que la instrucción explícita resulta fundamental para promover la metacognición en el estudiante, según Mateos (2001). Luego, en horarios de consulta y a través de los foros disponibles en Moodle, guían y supervisan (es decir, se aplica la práctica guiada propuesta por Mateos (2001) el avance de los grupos a su cargo. Entre los temas tratados podemos mencionar: compresión de datos sin pérdida, evaluador de expresiones matemáticas, detección de bordes en imágenes digitales, criptografía, String Matching, Auto-suggest, Sudoku, revisor ortográfico, 8-Puzzle, Text to Sound (TTS) entre otros.

Desde 2014 se realiza un taller de JAVA, en horario extra-áulico y al inicio del cuatrimestre, para introducir al estudiante de LAS en dicho lenguaje y en el Paradigma Orientado a Objetos. Así, 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

²El nuevo programa se encuentra disponible en <http://bo.unsa.edu.ar/dex/R2015/R-DEX-2015-604.PDF>

desarrollarlos, dadas las ventajas que ofrecen los mismos en la actualidad (Dale y otros, 2016).y la conveniencia de integrarlos a los contenidos desarrollados en la materia.

A partir de 2012, se 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. De esta manera, se dispone de otro medio de comunicación. En este caso, virtual.

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

Una acción que ha sido implementada en 2017 está relacionada con la práctica guiada. Siguiendo el método propuesto por Schoenfeld (Mateos, 2001), desde la revisión de trabajos finales (en horarios de consulta) se trabajó de manera cooperativa con los integrantes de cada grupo a través del plugin Saros (2018). De esta manera, se supervisaba y ayudaba a mejorar la solución computacional propuesta por cada grupo de manera virtual. El uso de Saros fue aceptado de manera masiva por los estudiantes quienes en general estaban acostumbrados a trabajar en grupos pero no de manera colaborativa. Con esta herramienta, la producción de software aumenta de manera significativa.

5.3 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 (Celman, 1998). En ese sentido, se evalúa al estudiante en dos momentos puntuales durante el cursado lo que nos permite tomar decisiones consecuentes: al finalizar la unidad 2 (Recursividad) y al finalizar la unidad 7 (Técnicas de diseño de algoritmos). En el primero, 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 del programa analítico vigente.

La evaluación de exámenes parciales y trabajos finales es procesual (tal como sugiere el parámetro Evaluación del modelo Target (Huertas, 1996). En los parciales, se evalúan los temas incluidos, con ejercicios fragmentados y evitando solapamiento de contenidos. 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, similares a los propuestos en los trabajos prácticos en cuanto a consigna y complejidad. 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 aplica la *evaluación dinámica* (Celman, 1998); esto es, circular en ciertos momentos por los pasillos, observar el desarrollo de los exámenes y brindar pistas que sean disparadores para que 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 incluyen anotaciones que indican si lo desarrollado es correcto o en caso contrario, lo 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.

5.4 Análisis de Porcentajes de Regularidad

La Fig. 2 se muestran los porcentajes de regularidad por Cohorte. La gráfica de negro muestra porcentajes de regularidad sobre el total de estudiantes inscriptos, mientras que la de gris, sobre el total de estudiantes regulares y no-regulares. Si se analizan los porcentajes de regularidad de los últimos dos años (2017 y 2018) se observan mejoras por sobre los anteriores. El porcentaje de regularidad, sobre los que regularizaron y los que quedaron libres, fue de 72.58% en 2017 y 82.22% en 2018. En tanto que si analizamos los porcentajes de regularidad, sobre el número de inscriptos, en 2017 se obtuvo 52.94% en 2017 y 46.84% en esta última cursada. Respecto a estudios realizados sobre cátedras con contenidos similares, nuestros resultados son alentadores. La Red Martínez y otros (2015) en la UTN-Resistencia informan porcentajes de regularidad cercanos al 20%, en tanto que Ferreira Szpiniak y Rojo (2006) en la UNRC, 30.5%.

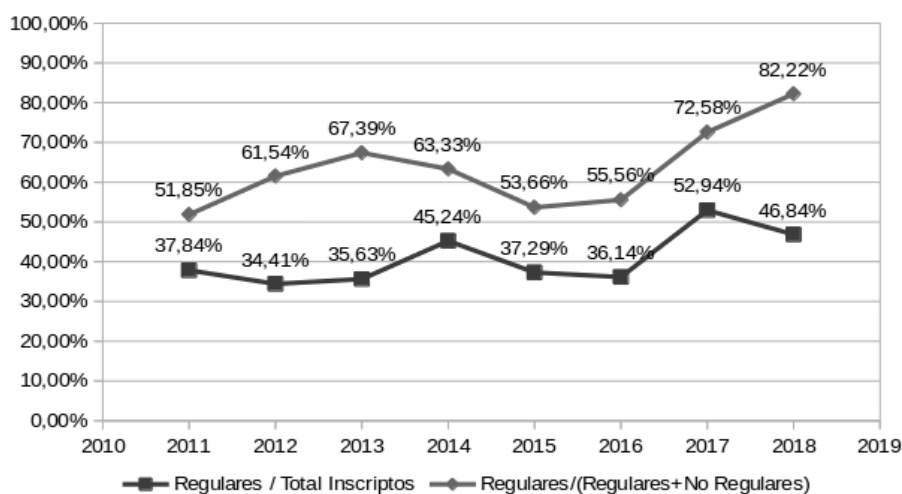


Figura 2: Resultados por Cohorte

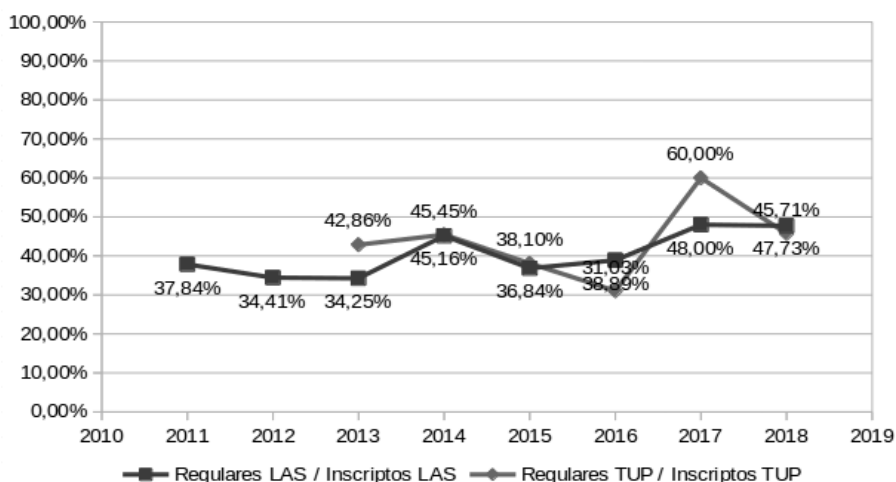


Figura 3: Resultados por Cohorte y Carrera

En la Fig. 3 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 de color negro) y decreciente (gráfico color gris) de los estudiantes de TUP, pero con un importante repunte durante 2017. Los porcentajes de regularidad de LAS y los de TUP en 2018 fueron de 47.73% y 45.71% respectivamente; estos valores evidencian que se está alcanzando una estabilidad en los porcentajes de regularidad de ambas carreras.

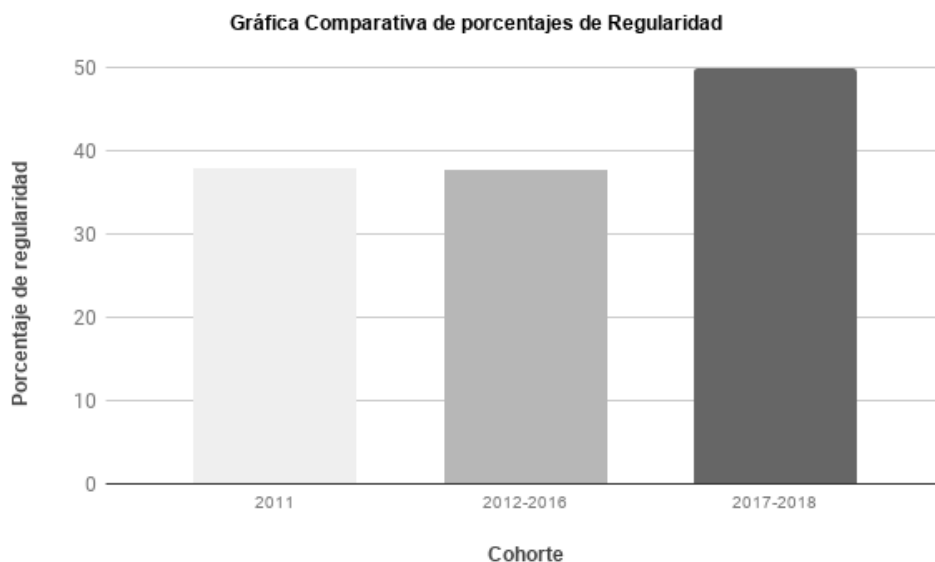


Figura 4: Comparación de Porcentajes de Regularidad

En la Fig. 4 se presenta un gráfico comparativo de los porcentajes de regularidad, según etapas de la asignatura. En los inicios, se observa un porcentaje de regularidad de 37.84%, porcentaje similar al período 2012-2016. En el período 2017-2018 el porcentaje promedio de regularidad es del 49.89%. Este aumento del porcentaje de regularidad alcanzado confirma las mejoras realizadas.

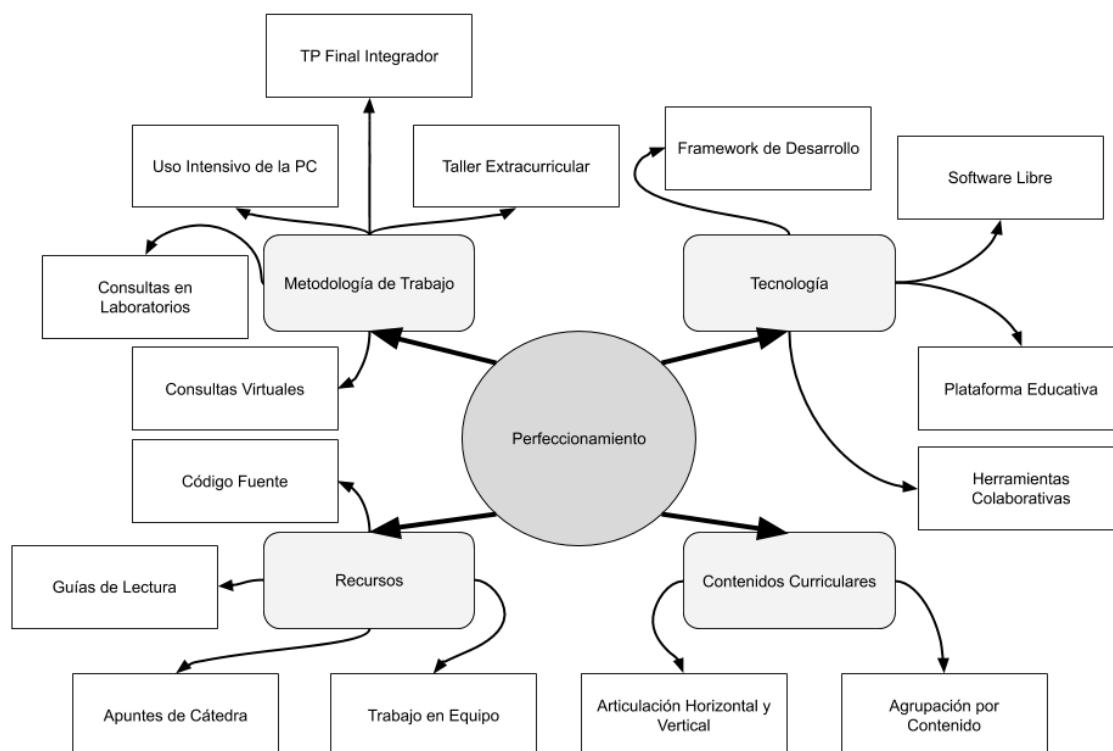


Figura 5: Esquema de la metodología implementada.

En la Fig. 5 se muestra un esquema sobre los aspectos más relevantes de la metodología de trabajo llevada adelante por la Cátedra.

5.5 Mejoras a Implementar en la Asignatura

Más allá de las mejoras alcanzadas (como se muestra en la figura anterior), es necesario llevar adelante nuevas acciones, las cuales han sido puestas en estudio.

La manera de transferir conocimientos y experiencias, depende en parte de la formación académica y experiencia laboral del Docente. Una mejor formación contribuirá a motivar mejor a sus estudiantes, motivándose primero él mismo (evitando el nopodernimiento, indicado por Abramowsky (2007)). 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 y un Jefe de Trabajos Prácticos tienen título de Doctor en Ciencias de la Computación.

Otra mejora tiene que ver con incrementar el material didáctico disponible para los estudiantes. Para ello, se desarrollarán *guías de lectura* para las unidades 6 y 7 del programa. Consideramos esta acción una medida positiva por varias razones. En primer lugar, acerca al estudiante a la lectura de un libro. En segundo lugar, las guías propuestas hacen referencia a algoritmos tratados en clase, lo cual favorece al aprendizaje significativo. Por último, las guías acompañan al estudiante en su aprendizaje autónomo.

Otra acción a implementar partir del próximo año es el trabajo colaborativo en clases prácticas. Siguiendo el método propuesto por Schoenfeld (Mateos, 2001), los Jefes de Trabajos Prácticos en laboratorio, pedirán a sus estudiantes que participen de

manera colectiva en la resolución de un problema computacional, de manera similar a lo indicado por Gallego y Gortázar(2009) en cuanto al uso de Eclipse-Gavab. En nuestro caso, en clases prácticas se utilizará SAROS.

Finalmente, en 2017 se inició la tutoría a estudiantes que no regularizaron la asignatura y quieren rendir el examen final libre, invitando a los estudiantes que obtuvieron entre 40 y 59 puntos en el segundo examen parcial o recuperatorio, a participar de la misma. Durante 3 semanas, se asiste en las unidades del programa donde su rendimiento fue inferior al esperado de manera que pueda superar esas deficiencias. En el segundo turno del mes de julio de 2017 y 2018, los estudiantes tutorizados (que se presentaron a rendir) por la Cátedra, aprobaron sendos exámenes libres, algo que anteriormente ocurrió una única vez en 6 años.

6. Conclusiones y trabajo futuro

Se ha analizado la práctica docente de la asignatura Algoritmos y Estructuras de Datos. Asimismo, y como parte del análisis de diferentes fuentes de datos (entrevistas, encuestas, registros de clase, documentación, entre otros) se han iniciado nuevas acciones desde 2017 a la fecha, en pos de mejorar el trabajo realizado por la Cátedra.

Las acciones 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 POO), en tecnología (por ejemplo, la migración de Pascal a Java) y TICS, en evaluación de saberes y aplicación de los mismos (Trabajo Práctico Final). 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.

Sin embargo, aún queda mucho por hacer más allá de lo iniciado en 2017. Es necesario realizar un mejor seguimiento de los estudiantes que no aprobaron el primer parcial y no finalizaron Programación, pues tienen alta probabilidad de abandonar el cursado. Asimismo, debemos continuar estudiando el fenómeno del abandono, si queremos retener más estudiantes y aumentar el número de graduados en nuestro Departamento.

Como trabajo futuro vamos a estudiar diferentes configuraciones de capas y cantidad de nodos de la arquitectura MLP con el objetivo de mejorar la performance final de las predicciones de la situación al final del cursado de los estudiantes de AyED. También vamos a implementar otros algoritmos de aprendizaje automático para resolver este problema de clasificación, por ejemplo el uso de SVM (Huang y Fang, 2013).

Referencias

- Abramowsky, A. (2007). Variaciones del nopodernimiento escolar: perezas y desmotivaciones en Interés, motivación y deseo. Noveduc.
- Aggarwal, C. (2018). Neural Networks and Deep Learning: A textbook. Springer.
- Arellano, N. y otros (2014). Una experiencia en la enseñanza de la programación para la permanencia de los alumnos de Ingeniería Electrónica. Enseñanza y Aprendizaje de Ingeniería de Computadores, 4 (20), 69-80.

- Bruner, J. (1997). La educación, puerta de la cultura. Visor.
- Celman, S. (1998). ¿Es posible mejorar la evaluación y transformarla en conocimiento? en La evaluación de los aprendizajes en el debate didáctico contemporáneo. Paidós.
- Dale, N. y otros (2016). Object-Oriented Data Structures using Java. Jones & Bartlett Learning.
- Ferreira Szpiniak, A. y Rojo, G. (2006). Enseñanza de la Programación. Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología, 49 (2), 1-8.
- Gallego, M. y Gortázar, F. (2009). Eclipse-Gavab, un entorno de desarrollo para la docencia online de la programación. Trabajos de las XV JENUI, Barcelona.
- Huang, S. y Fang, N (2013). Predicting student academic performance in an engineering dynamics course: A comparison of four types of predictive mathematical models. Computers & Education, 61, 133-145.
- Huertas, J. y otros (1996). Principios para la intervención motivacional en el aula en Motivación. Querer aprender. Aique.
- Keras. (2015). Librería Keras. Disponible en <https://github.com/fchollet/keras>.
- La Red Martínez, D. y otros (2015). Determinación de Perfiles de Rendimiento Académico en la UTN-FRRe. Trabajos del III CONAIISI, UTN-Buenos Aires, Buenos Aires.
- Mateos, M. (2001). Metacognición y educación. Aique.
- Moreno, E. y otros (2013). El trabajo colaborativo como estrategia para mejorar el proceso de enseñanza-aprendizaje aplicado a la enseñanza inicial de Programación en el ambiente universitario. Trabajos del I CONAIISI, UTN-Córdoba, Córdoba.
- Pozo, J. (2006). Teorías cognitivas del aprendizaje. Morata.
- Rokach, L. y otros (2014). Data Mining with decision trees: theory and applications. World Scientific Publishing.
- Salazar, L. (2015). Nuevas estrategias en la enseñanza de la programación mediada por TIC para alumnos de la asignatura Informática de la UNNE-FACENA (tesis de maestría), Universidad Nacional de Córdoba, Argentina.
- Saros. (2018). Saros plugin para Eclipse. Disponible en <http://www.saros-project.org/>.
- Soler Pellicer, Y. y otros (2018). Ambiente integrado de visualización de Estructuras de Datos para la enseñanza-aprendizaje de la Programación de Computadoras. Opuntia Brava, 10(2), 10-24.
- Zhao, Y. (2013). R and Data Mining: examples and case studies. Academic Press.