

Corrección automática de situaciones algorítmica - Programación autoevaluable

Mariana P. Simieli; Mario I. Rinaldi; Paula A. Toselli

Dpto. de Ingeniería en Sistemas de Información - Universidad Tecnológica Nacional, Facultad Regional Villa María, Avenida Universidad 450, – XGB5900 – Villa María, Córdoba, Argentina.
simielip@frvm.utn.edu.ar - ptoselli@frvm.utn.edu.ar - rinaldimario@gmail.com

Resumen

La necesidad de utilizar un componente para la corrección automática de situaciones algorítmicas se manifiesta por la masividad de estudiantes que ingresan en el primer nivel de la carrera de Ingeniería en Sistemas de Información, en la cátedra algoritmos y estructura de datos, la cual pertenece al bloque de programación del diseño curricular.

Los/as estudiantes del primer nivel de ingeniería requieren de una enseñanza personalizada y sistemática, donde la mediación pedagógica, es decir, la metodología que van a implementar las/os docentes para el logro de aprendizaje se va componer de estrategias, desafíos y actividades prácticas, tanto individuales como grupales. Esto sin la ayuda de recursos mediados tecnológicamente hace que se dificulte el proceso de aprendizaje y se requiera con mayor énfasis una intervención autónoma de cada estudiante.

En este trabajo se presenta la implementación de un componente de licencia libre denominado VPL (Laboratorio Virtual de Programación) permitiendo una retroalimentación automática para los actores del proceso de aprendizaje, logrando una motivación por superar las diferentes instancias resolutivas), este componente puede ser instalado en la plataforma Moodle y utilizado por múltiples cátedras de ingeniería. Su funcionalidad en la resolución de problemas algorítmicos permite al estudiante observar información sobre los casos de pruebas en la cual fue analizada su algoritmia y finalmente su calificación final.

Como conclusión, se indica que la implementación del recurso VPL se diferencia en tres procesos que evolucionan optimizando el resultado esperado. El primer proceso es la intervención del docente en la configuración de los múltiples parámetros y casos de pruebas según la finalidad en la resolución de cada propuesta. El segundo es la asimilación del estudiante con el recurso, la cual presenta una cierta resistencia en la primera fase, que nos permite analizar diferentes habilidades de los/as estudiantes para continuar con la mediación tecnológica propuesta. Finalmente, el tercero comienza cuando el uso del recurso deja de ser una resistencia y el propósito se transforma en el pensamiento computacional óptimo para resolver el problema, obteniendo una retroalimentación inmediata y una autonomía de los/as estudiantes en su proceso de aprendizaje.

Palabras Clave: laboratorio virtual de programación (VPL), programación, retroalimentación inmediata, autonomía de los estudiantes.

Introducción

El aprendizaje de la programación en los primeros años de las carreras de ingeniería es compleja y dificultosa, la misma requiere un desarrollo intelectual y una capacidad de abstracción de los problemas, síntesis de la información y un pensamiento que permita a los/as estudiantes seleccionar decisiones y resolver inconvenientes de forma efectiva, lo cual es fundamental para aprender a desarrollar algoritmos.

Actualmente, la cantidad de estudiantes del primer nivel de ingeniería se ha incrementado exponencialmente y requiere de una enseñanza personalizada y sistemática, donde la mediación pedagógica, es decir, la metodología que van implementar las/os docentes para el logro del aprendizaje se va a componer de estrategias, desafíos y actividades prácticas, tanto individuales como grupales. Para ello, se introducen en las universidades nuevas metodologías, donde se modifica el uso del tiempo y del espacio en el aula y fuera de ella, pero también se altera el protagonismo de los actores que intervienen en el proceso.

En el contexto de la enseñanza en programación, la plataforma Moodle permite implementar un módulo denominado Virtual Programming Lab (VPL) que, entre otras funcionalidades, ofrece a los/as estudiantes codificar en diferentes lenguajes, utilizando un navegador, sin grandes requerimientos de *hardware* (<https://vpl.dis.ulpgc.es/index.php>).

Cuando nos centramos en la programación, estamos considerando que la verdadera práctica se manifiesta en el pensamiento computacional, es decir, en la abstracción para la resolución de problemas, que es lo central y primordial en toda situación de algoritmia. Por lo general, el/la estudiante interpreta que debe codificar en un lenguaje de programación como primera instancia. El uso de un laboratorio virtual permitió que el/la estudiante disponga de una herramienta automatizada para codificar y autoevaluarse, logrando, como retroalimentación, que saber utilizar únicamente la sintaxis de un lenguaje de programación no le asegura la solución a un problema algorítmico.

Objetivos planteados al momento de utilizar un laboratorio virtual como experiencia en la enseñanza de la programación:

- Visualizar la configuración de los múltiples parámetros y casos de pruebas en la herramienta analizada (VPL) en la resolución de cada propuesta.
- Disponer de una herramienta automatizada para lograr una retroalimentación inmediata hacia el/la estudiante y hacia el/la docente.
- Lograr una autonomía de los/as estudiantes en su proceso de aprendizaje en programación.
- Concluir en las nuevas capacidades a desarrollar en los/as estudiantes cuando su aprendizaje está mediado tecnológicamente. Como, por ejemplo, el trabajo colaborativo, el pensamiento crítico de la información a la que se accede, curación de contenidos, entre otras.

Aspectos positivos y dificultades encontradas

Análisis y desarrollo de etapas

El presente trabajo tiene un alcance descriptivo, ya que pretende medir o recoger información de manera independiente o conjunta sobre el rendimiento académico y el aprendizaje del estudiante en el área de programación.

El alcance de esta investigación es de tipo descriptivo porque se observan diferentes prácticas en la cátedra Algoritmos y Estructura de Datos del 1er Nivel de la carrera Ingeniería en Sistemas de Información durante los años 2020, 2021 y 2022.

Primera etapa: configuración de los múltiples parámetros y casos de pruebas según la finalidad en la resolución de cada propuesta.

Una vez instalado el módulo de VPL (por el administrador de la plataforma), el mismo se visualiza como un recurso disponible en la plataforma Moodle. En la figura 1 se muestra el icono representativo del módulo a implementar.

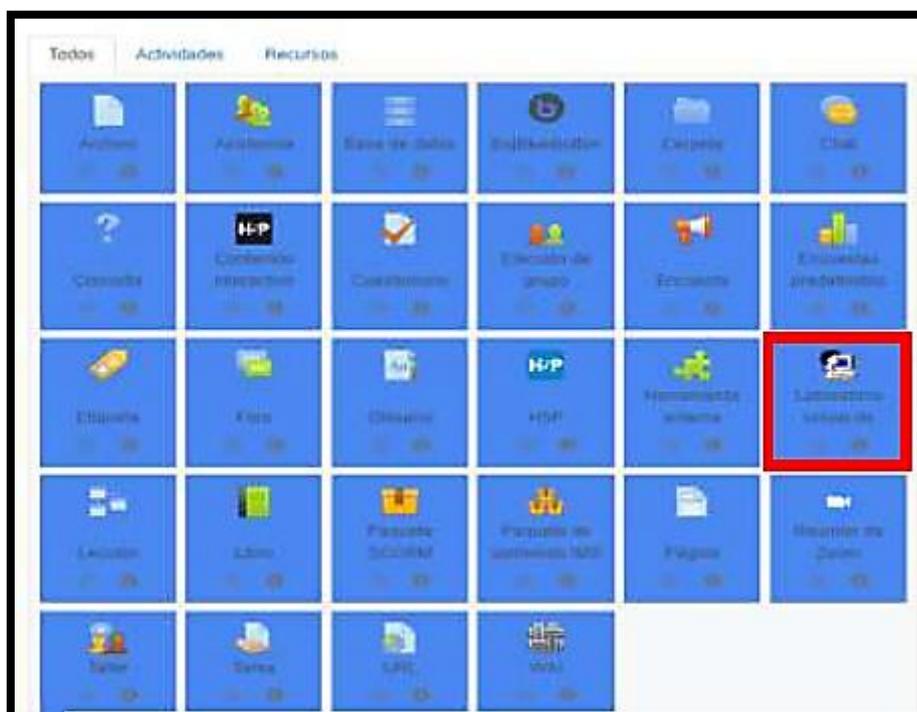


Figura 1 - Icono representativo del módulo VPL

Aquí comienza el trabajo del docente, que consiste en configurar los diferentes parámetros a establecer, como nombre y descripción del recurso, fecha de apertura y cierre, cantidad de intentos, penalidades, tipo de trabajo (individual o grupal), cantidad de archivos posibles a ejecutar; establecer la calificación mediante escala o puntuación, según evaluemos por objetivos o por puntos y, en una escala de 10 a 100, se permite especificar la puntuación mínima para aprobar, reducir puntos por número de evaluaciones automáticas (pruebas del

alumno) y número de evaluaciones permitidas, sin reducción de puntuación. Son todos los parámetros básicos muy similares a cualquier recurso que se utiliza en Moodle como una página, una tarea o un foro. (Figura 2)

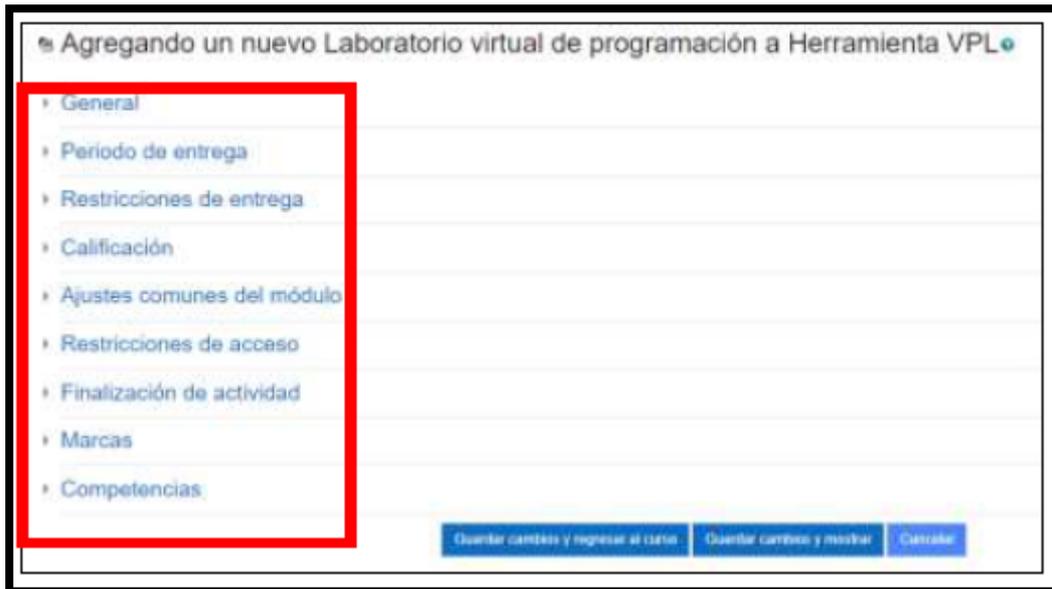


Figura 2 - Opciones de configuración VPL

A continuación, el recurso VPL nos presenta una cierta cantidad de opciones más específica, como se visualiza en la Figura 3, según el objetivo que se persigue con la actividad virtual. Una de esas opciones son los casos de prueba, su carga bajo la sintaxis propuesta es fundamental, porque esto le permite al estudiante que, cuando ejecute su código, pueda visualizar información sobre en qué casos de prueba su lógica de resolución funciona o no, o si contiene alguna dificultad de análisis.

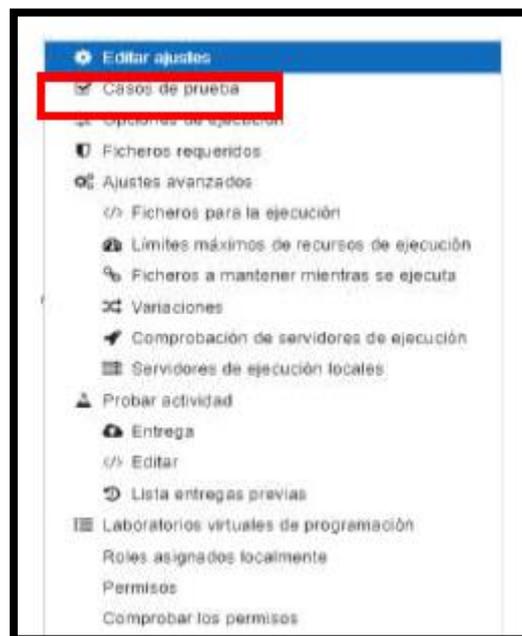


Figura 3 - Opciones de configuración VPL

Los casos de prueba permiten configurar los parámetros de entrada y salida a los que se verá condicionada una solución para poder obtener un informe de evaluación satisfactorio. Todo caso de prueba debe incluir una breve descripción, unos parámetros de entrada y salida esperada, y un grado de reducción en caso de fallo.

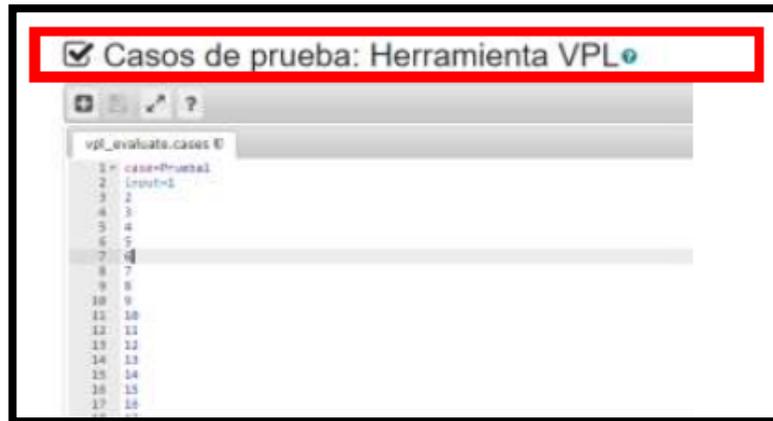


Figura 4 – Casos de Prueba VPL

Para definir los casos de prueba se deben tener en cuenta las siguientes consideraciones:

Todo caso de prueba tiene una cabecera que determina el inicio del caso de evaluación. Se pueden definir varios casos de pruebas dentro de un mismo archivo de evaluación, cada uno separado por su cabecera: **Case = Name Case** Los parámetros de entrada en un caso de evaluación se definen mediante la estructura **Input = Text|Number**. Las entradas del caso de evaluación pueden comprender varias líneas y culmina cuando se define otra sentencia. La salida que debe arrojar la solución a la actividad se puede definir en el caso de prueba con la estructura **Output = Text|Number|Regex**. Se pueden definir múltiples salidas y terminan cuando se define otra sentencia.

Segunda etapa: asimilación del estudiante con el recurso VPL

Cuando el/la estudiante comienza a interactuar con el recurso presenta dificultades, porque desconoce su funcionalidad y la interfaz del editor es rígida y debe respetarse con exactitud. Pero, principalmente, el grado de dificultad se observa en la lógica computacional del estudiante al resolver la situación problemática.

En referencia a la enseñanza de la programación, el dilema de hoy parece estar en torno al *software* que se escribe versus la forma en que se piensa para resolver problemas. Raja (2014) indica que, si se empieza por enseñar el pensamiento computacional en vez de la programación, van a tener saberes y habilidades relacionadas con la forma de pensar y la resolución de problemas que los harán más competentes en el desarrollo de sus profesiones, cualesquiera que estas sean.

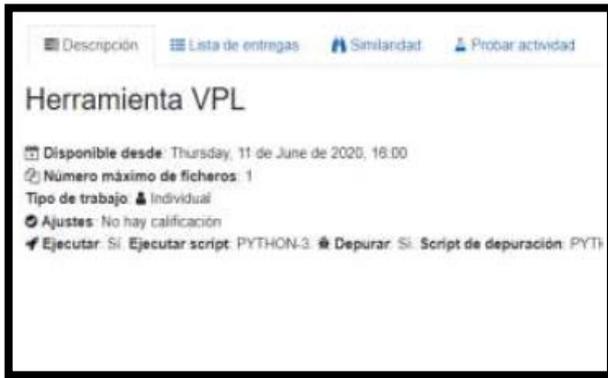


Figura 5 – Visualización del docente



Figura 6 - Visualización del estudiante

Tercera etapa: el propósito se transforma en el pensamiento computacional óptimo para resolver el problema

Cuando el/la estudiante asimila el conocimiento sobre la forma en que se deben enfrentar las etapas de análisis e implementación, su nivel de aprendizaje es adecuado en cuanto a pensar y resolver situaciones problemáticas, es decir, aplica pensamiento computacional, la metodología propuesta es eficiente y se logra un aprendizaje significativo en el/la estudiante.

En el rol de estudiante, se puede visualizar su código, los casos de pruebas y su calificación con la retroalimentación que el/la docente puede escribir en diferentes momentos de la actividad.

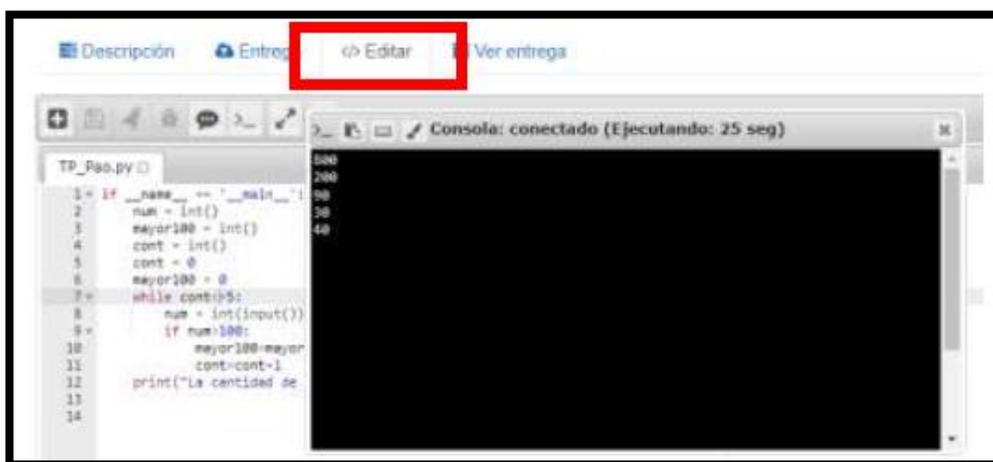


Figura 7 – Vista del estudiante - VPL

En el rol del docente, se observa un resumen de entregas y evaluaciones de todos los laboratorios cerrados, calificados, lo cual le concede al docente un seguimiento de cada estudiante, según las prácticas implementadas. VPL creados en el aula virtual, con distintas

opciones de visualización según su estado:



Figura 8 – Vista del profesor/a – Lista de entrega VPL

El rol docente tiene la funcionalidad de encontrar similitudes entre las distintas soluciones entregadas y determinar la cantidad de salidas que se quiere exportar para la búsqueda que mostrará parejas relacionadas por el porcentaje de similitud o similitud que existe entre ellas, y el/la profesor/a podrá tener un criterio para evaluar si dichas soluciones realmente pueden ser consideradas como plagio o no.



Figura 9 – Vista del profesor/a – Lista de entrega VPL

El/la profesor/a también puede adjuntar a las fuentes de búsqueda de similitud, las entregas de actividades del mismo tipo creadas con anterioridad en este u otro de sus cursos.

Se mostrará una matriz de relación entre cada solución entregada en la presente actividad y las demás soluciones, así como el porcentaje de semejanza que hay entre cada una de ellas. De esta manera se puede analizar con más profundidad las soluciones y el porcentaje de coincidencia que existe entre ellas.

Conclusiones

Este módulo de VPL implementado en el aula virtual permite al estudiante dedicar el tiempo necesario a cada una de las principales etapas que se integran en la solución de problemas (análisis de problemas, diseño e implementación de programas), sin verse sometido a las limitaciones propias del tiempo establecido para una clase presencial.

La estrategia, la cual consiste en la confección de múltiples casos prácticos de diferentes complejidades con sus casos de pruebas, permite al estudiante avanzar a su propio ritmo, comunicándose con el profesor cuando le sea necesario, ya sea en el tiempo de clase presencial o fuera de los límites del aula, utilizando recurso de comunicación que ofrece la plataforma Moodle.

La ventaja del uso del VPL depende de todos los componentes que se integran en la práctica de laboratorio. El trabajo colaborativo de los estudiantes es fundamental, lo cual puede analizarse como una de las capacidades fundamentales, como así también la autonomía en su proceso aprendizaje. Cuando el estudiante asimila el conocimiento sobre la forma en que se deben pensar y resolver situaciones problemáticas, es decir, aplica pensamiento computacional, se logra el aprendizaje significativo por parte del estudiante. Obteniendo una retroalimentación inmediata y autonomía en su proceso de aprendizaje.

El compilador integrado a la plataforma cumple con su propósito en forma eficiente, permitiendo la edición, ejecución, y prueba de programas a través de una interfaz de fácil manejo a la que se ingresa por la web.

Esta característica permite que los/as estudiantes puedan iniciar la implementación de sus programas en clase presencial y continuar posteriormente desde un lugar externo a la Universidad sin necesidad de tener instalado un lenguaje de programación específico.

Una característica destacable del módulo VPL está en las posibilidades de aprovechamiento del historial de entregas de cada estudiante, para hacer un seguimiento a la forma en que se avanza en la implementación de la solución al problema.

Bibliografía

- ALZATE-ORTIZ, Faber Andrés, CASTAÑEDA-PATIÑO, Juan Carlos. (2020). Mediación pedagógica: Clave de una educación humanizante y transformadora. Una mirada desde la estética y la comunicación. *Revista Electrónica Educare*, 24(1), 411-424. Epub January 30, 2020. <https://dx.doi.org/10.15359/ree.24-1.21>
- BORDIGNON, Fernando (2020) Introducción al pensamiento computacional. - 1a edición para el profesor - Ciudad Autónoma de Buenos Aires: UNIPE; Editorial Universitaria; EDUCAR S.E., Libro digital, PDF Recuperado en: <https://libros.unlp.edu.ar/index.php/unlp/catalog/view/1300/1283/4210-1>
- CABERO, Julio y LLORENTE, M^a del Carmen. (2008). Del e-learning al Blended Learning: nuevas acciones educativas. *Quaderns Digitals*, (51)
- FERREIRA Szpiniak Ariel, ROJO Guillermo. (2006) Enseñanza de la programación. En: *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología. Volumen 1*, Número 1
- SAMPERIO MONROY, T., HERNÁNDEZ MENDOZA, S., LICONA OLMOS, J., & PÉREZ SILVA, D. (2016). Plataformas LMS, una alternativa TI de éxito en los sistemas educativos de nivel superior. *Ciencia Huasteca boletín científico de la escuela superior de Huejutla*, 4(7). doi:10.29057/esh.v4i7.1137
- VPL– Virtual Programming Lab. What is VPL? Disponible (9 de noviembre de 2023) en: <https://vpl.dis.ulpgc.es/index.php/about>
- ZAPATA-ROS, M. (2015). Pensamiento computacional: una nueva alfabetización digital. *RED, Revista de Educación a Distancia*, 46(4).