

Algunas dificultades en el aprendizaje del concepto de variable

Juan Saavedra y Adrián Silveira
 Facultad de Ingeniería, Universidad de la República
 Junio, 2011
 {juansaav,adrians}@fing.edu.uy

Resumen— El concepto de variable tanto en matemática como en informática es fundamental y su correcto aprendizaje por parte del estudiante es un problema de difícil abordaje para la didáctica. Se presenta en este artículo un estudio del arte sobre la enseñanza del concepto de *variable* tanto en cursos de programación como de matemática para estudiantes de diferentes edades.

El estudio consiste en un análisis de seis trabajos, algunos orientados al diagnóstico de las dificultades para su aprendizaje y otros que presentan potenciales herramientas para una mejor enseñanza del concepto.

Palabras Claves— Didáctica de la informática, Concepto de variable, Problemas de aprendizaje.

I. INTRODUCCIÓN

ESTE trabajo presenta un estudio sobre la dificultad del aprendizaje del concepto de variable tanto en programación como en matemática, resumiendo y detallando los conceptos relevantes a la hora de investigar las dificultades asociadas con la enseñanza de la informática en particular.

Es un trabajo del área de la didáctica de la informática, realizado como complemento al curso de "Una Introducción a la Didáctica de la Informática" dictado por la Dra. Sylvia da Rosa en la Facultad de Ingeniería en el año 2010. Se basa en el estudio de seis artículos y algunas referencias de los mismos, en donde se cuenta con una variedad de investigaciones con distintos objetivos donde todos abarcan en mayor o menor medida las dificultades que existen en el aprendizaje del concepto de variable. Un conjunto de estos trabajos, está basado sobre el concepto de *rol de una variable* que detallaremos más adelante pero que en resumen implica un uso típico de una variable en un programa diferenciándose en la forma en que toman valores.

Se trata de brindar la mayor agregación posible de los conceptos vertidos en los diferentes artículos de forma que exista una mayor claridad en la comparaciones entre ellos. Sin embargo, cada artículo contiene por si mismo algunas ideas o experiencias que merecen un tratamiento individual y por ende estas se encuentran enumeradas por separado. Todo esto se encuentra distribuido a lo largo del trabajo realizado y no se restringe a una única sección.

El trabajo está estructurado de la siguiente forma: en la siguiente sección se repasa el conjunto de artículos que son la base del estudio presentado, que nutren las siguientes secciones. Luego, se dedica una sección a un análisis según diferentes perspectivas sobre las dificultades del aprendizaje de las variables. Previo a la presentación de las conclusiones

en la última sección se dedican dos secciones a la presentación de los roles de las variables como una herramienta en la enseñanza y a diferentes experimentos realizados para evaluar su efectividad, así como otras hipótesis sobre las dificultades del aprendizaje de las variables.

II. TRABAJOS PREVIOS

SE introduce aquí el conjunto básico de artículos utilizados para el desarrollo de este trabajo, de forma de brindar el contexto sobre los mismos al lector. A lo largo de las siguientes secciones se brinda mayor información sobre cada uno y es conveniente referirse a los artículos para un mayor detalle y analizar la completitud de los mismos, ya que para algunos fueron dejados por fuera aquellos datos que no son relevantes para el análisis.

El trabajo de Pauli Byckling y Jorma Sajaniemi [1] presenta un estudio sobre las mejoras en los programas desarrollados por estudiantes que aprenden programación, a los cuales se le enseñaron roles de variables en diferentes formas. El objetivo es demostrar la eficacia de los roles como instrumentos en la didáctica de la informática para facilitar el entendimiento por parte del estudiante. Presentan un experimento con tres grupos de control y analizan los resultados obtenidos en cada grupo en los cursos iniciales de programación, así como las cualidades de un programa de prueba desarrollado por algunos estudiantes para el estudio realizado. Este artículo puede verse como una continuación del trabajo desarrollado por Sajaniemi [2], en donde el autor introduce el concepto de *rol de una variable* que se encuentra más adelante.

Wagner [3] presenta una discusión en torno a la dificultad de entender el concepto de variable para los estudiantes de matemática a niveles iniciales y una enumeración de los posibles obstáculos para su aprendizaje, presentando su visión sobre por qué los símbolos literales son un concepto fácil de utilizar pero difícil de entender. Además plantea algunas sugerencias para la enseñanza de las variables utilizando los conceptos con los que cuenta el estudiante previo al estudio de programación.

Renan Samurçay [4] presenta un estudio empírico para entender el concepto de variable con el que cuenta un estudiante al iniciar sus estudios en programación, de forma de conocer el punto de partida para la construcción de situaciones para la enseñanza de la misma. El trabajo se da en el marco de una investigación mayor con el objetivo de enumerar las características intrínsecas de la programación para mejorar su

enseñanza. Parte de ese trabajo está dedicado a identificar y describir los conceptos de programación desde un punto de vista epistemológico tomando en cuenta su complejidad relativa (tanto por cantidad de información como por la dificultad conceptual). Apunta a construir una serie de problemas teóricos y prácticos con el objetivo de que el estudiante vaya incorporando los conceptos que solucionan dichos problemas. Lina Morales Peral et al. [5] presentan un estudio de algunas dificultades que tienen los estudiantes universitarios para alcanzar un manejo adecuado del concepto de variable. Proporcionan un marco teórico para su trabajo y realizan una encuesta de modo de detectar los usos del concepto de variable.

Juha Sorva et al. [6] abordan el uso de roles de variables en la enseñanza y cómo se pueden beneficiar los instructores al incorporarlo al material. Además, presentan al rol de variable como formador de esquemas y mediante dos experiencias en cursos universitarios evalúan su impacto.

Uolevi Nikula et al. [7] discuten la aplicabilidad de los roles de variable en Python y presentan variantes a la clasificación de Sajaniemi para adaptarse a las características del lenguaje. Además, exponen tres experiencias de distintas universidades donde los cursos de iniciación a la programación adoptaron Python como lenguaje y roles de variable para ayudar a la comprensión del comportamiento de los programas.

III. PROBLEMA PARA EL APRENDIZAJE

PARA poder crear mejores herramientas para la enseñanza es preciso realizar un diagnóstico de las dificultades que existen en torno a la comprensión. Del estudio realizado, es posible obtener varias visiones sobre las dificultades que existen para una correcta comprensión del concepto de variable, tanto en la matemática como en la programación.

A. Dificultades en la matemática

En el trabajo de Lina Morales et al. [5] se realiza una primera encuesta a estudiantes y docentes de diferente nivel educativo que consiste en definir solamente el concepto de variable, con el fin de detectar el uso de la variable por los estudiantes. A partir de las respuestas, se diseñó un segundo cuestionario con el fin de evaluar el conocimiento de las distintas caracterizaciones de la variable (como una incógnita específica, como un número general y como una relación funcional) por parte de los estudiantes. Del análisis cualitativo de las respuestas se destaca:

- No es clara la diferencia entre variable y constante
- No se tiene la habilidad para interpretar y representar mediante una expresión los problemas
- Es dificultosa la generalización de resultados en un proceso, así como para representar una situación como relación funcional

Como explicación, los autores señalan que hay una dificultad intrínseca en la comprensión del concepto de variable y es su propia definición. El carácter multifacético del concepto se evidencia a lo largo de la literatura matemática, donde se utilizan los mismos símbolos para denotar diferentes caracterizaciones y diferentes símbolos son empleados para denotar la misma

caracterización. Küchemann [8], mediante un experimento con estudiantes de entre 13 y 15 años, identificó seis maneras diferentes de interpretar los símbolos literales que representan variables:

- Letra evaluada, se le asigna un valor numérico.
- Letra no utilizada, es ignorada o su existencia es reconocida pero carece de significado.
- Letra como objeto, es una abreviación del nombre de un objeto o un objeto en sí.
- Letra como incógnita específica, representa un número particular pero desconocido y los estudiantes son capaces de operar directamente sobre ella.
- Letra como número generalizado, representa o es capaz de asumir distintos valores.
- Letra como variable, representa un rango de valores no especificado y que existe una relación sistemática entre dos conjuntos de valores de este tipo.

De allí, se destaca que la fuente más frecuente de respuestas erróneas al resolver un problema con variables es la incorrecta interpretación de las mismas. Además, Küchemann sugiere que la enumeración anterior de las distintas interpretaciones está en un orden creciente de dificultad de comprensión, lo cual significa que un estudiante habrá comprendido el uso de los símbolos literales del álgebra cuando logre trabajar con la "*letra como variable*".

Por otro lado, Wagner[3][5] plantea que la interpretación de las variables por parte de los estudiantes, se debe a la combinación de tres componentes (símbolo, referente y contexto) y del papel semántico, sintáctico y matemático que desarrollan. Según el autor, las confusiones surgen en base a varios factores que pueden categorizarse en:

- Similitudes entre variables expresadas como literales (literales de ahora en más) y numerales.
- Similitudes entre literales y palabras.

Basado en esta categorización reconoce las similitudes y diferencias entre cada uno de los conceptos por categoría para luego pasar a explicar las dificultades que puede traer la concepción para el estudiante y las facilidades que trae para el docente.

Plantea que es sencillo explicarle a un estudiante literales mediante la asociación con numerales y palabras, puesto que son sistemas de símbolos ya conocidos y le pueden servir de referencia para comprender algunos conceptos sobre las variables (un abordaje similar se da en el trabajo de Samurçay[4]). Por ejemplo, una similitud entre literales y numerales es que existen literales que de hecho son números (por ejemplo, π). Sin embargo, un literal identifica usualmente elementos aleatorios y variables de un conjunto, mientras que los numerales representan un elemento fijo.

Con las palabras, una similitud notable planteada por Wagner es que las palabras y los literales pueden utilizarse como *placeholders* o delimitadores. En la frase "*Él es mecánico*" la palabra *él* se comportaría igual que el literal x en $2x + 5 = 8$ ya que es posible reemplazar ambos por valores posibles (personas para el primer caso) y verificar si el resultado es verdadero o falso. Otra similitud entre las palabras y los literales es que ambos pueden referirse a diferentes cosas en

diferentes contextos, ya que el significado de un literal depende de su rol, su dominio y los valores que lo verifican (en una proposición).

Una diferencia entre palabras y literales es que la misma palabra puede referirse a diferentes cosas en el mismo contexto, mientras que un literal no. Continuando con el ejemplo anterior, x se refiere siempre a lo mismo en un contexto dado, mientras que una misma palabra puede referirse a cosas diferentes en una misma oración inclusive (ejemplo en inglés: "His leg became numb during the last leg of the flight.", "leg" refiere a conceptos diferentes en sus dos apariciones).

Las consecuencias de dichas confusiones implican que el concepto no esté correctamente interiorizado y esto se manifiesta en el campo de la acción, por ejemplo, en la incapacidad de realizar generalizaciones de forma adecuada (un ejemplo de esto puede encontrarse en [3]). Este anclaje perdura en la etapa universitaria y los estudiantes no son capaces de reconocer el rol de las variables, bloqueando el aprendizaje de la matemática.

B. Dificultades en la programación

Uno de los trabajos que se dedica en exclusividad a las variables en programación es el de Samurçay [4], que trata de estudiar las dificultades sobre el aprendizaje y cuál es el entendimiento de los estudiantes sobre el concepto.

El argumento común a la hora de categorizar los conceptos según su dificultad de menor a mayor, es la base con la que cuenta el estudiante en el momento que aprende los mismos. Por ejemplo, se asume que conociendo las variables entendidas según la matemática el estudiante cuenta con un esquema que le permita acceder a algunos conceptos sobre las variables en programación, pero que dicha construcción es insuficiente ya que el concepto de variable en esta última disciplina es diferente.

Primero, reconoce que el identificar una variable como una dirección en memoria es un acercamiento insuficiente para que el estudiante analice su significado funcional. Un ejemplo de esto es que el valor puede cambiar, pero su relación con y su función en el programa permanecen inalteradas, algo que según el autor es más complicado de apreciar entendiendo las variables como direcciones de memoria.

Luego, hipotetiza sobre las dificultades respecto a ciertos tipos de variables y ciertas operaciones entre las mismas. Divide las variables en primera instancia en internas o externas, siendo las internas aquellas que pertenecen al alcance del programa que realiza la solución y no son controladas por el "usuario" mientras que las externas son los parámetros que envía el usuario de la función. Establece que las variables internas son un concepto más difícil de aprender y a su vez, que las tareas de inicialización de dichas variables son las más complejas de las tres (inicialización, actualización y prueba) ya que es una tarea no presente en la "solución manual".

Además plantea como una de las diferencias con las variables matemáticas, la operación de asignación que se realiza sobre las variables en programación, dividiéndolas en cuatro categorías:

- Asignación de un valor constante $x := 1$

- Asignación de un cálculo (que no involucra la propia variable) $x := 2 * k$
- Duplicación $x := k$
- Acumulación (cálculo que involucra la propia variable) $x := x + 1$

Señala a la acumulación como la operación de mayor complejidad conceptual para el estudiante, ya que implica agregar una noción de tiempo a la concepción de variable y que es la forma de mayor complejidad al trabajar con bucles y variables en programación.

Veremos más adelante la experiencia realizada en [4] para evaluar estas hipótesis.

IV. USO DE ROL DE VARIABLE EN EL APRENDIZAJE

Sajaniemi introdujo en el año 2002[2] el concepto de *Roles de variables* como estereotipos del uso de variables en programas de computadora, diferenciados entre sí por la sucesión de valores que toman y su alcance en el programa, semejantes hasta cierto punto a los patrones de diseño [1]. Se definen además como un concepto cognitivo y no técnico. Sajaniemi define un conjunto de roles de variables dividido en dos grupos:

- Roles Generales:
 - *Fixed Value*: el valor de la variable no cambia luego de inicializada.
 - *Stepper*: toma valores en un orden sistemático y predecible.
 - *Most-Recent Holder*: almacena el último valor en una secuencia de valores impredecibles.
 - *Most-Wanted Holder*: almacena el "mejor" valor encontrado en una secuencia de valores.
 - *Gatherer*: usada de alguna manera para combinar valores que son encontrados en una secuencia de valores y el valor de la variable representa este valor acumulado.
 - *Follower*: almacena el más reciente valor previo de otra variable. Si el valor de la variable seguida cambia, su valor también cambia.
 - *One-Way Flag*: solo tiene dos valores posibles y el cambio del valor es permanente.
 - *Temporary*: si el valor de la variable es utilizado por un período de tiempo determinado.
 - *Other*: cualquier otra variable que no esté en las categorías dadas.
- Roles Relacionados a Estructuras de Datos:
 - *Organizer*: almacena una colección de elementos con el proposito de tener dicha colección re-ordenada.
 - *Container*: almacena una colección de elementos en la cual más elementos pueden ser agregados.
 - *Walker*: atraviesa la estructura de datos, moviéndose de una localización en la estructura a otra.

Es apreciable que los roles dados no son muchos y según Sajaniemi es posible abarcar el 99% de los usos de las variables de los cursos de programación iniciales con 10 roles únicamente. En las siguientes subsecciones presentaremos algunas ventajas presentadas por el uso de roles en la enseñanza del concepto de variable.

A. Como herramienta para comprender

Byckling y Sajaniemi[1] continúan el trabajo en la investigación sobre la enseñanza de programación mediante roles y plantean un estudio empírico para registrar el impacto de la enseñanza de roles variables a tres grupos de estudiantes en cursos iniciales de programación. Los tres grupos aprendieron variables en formas diferentes: en dos utilizaban roles de variables y en uno de estos, se utilizó además un programa especial que brindaba imágenes y animaciones para caracterizar y representar el funcionamiento de los roles respectivamente.

Sobre esta experiencia, Sajaniemi y Kuittinen en el año 2005 demostraron mediante el análisis de los exámenes finales que aquellos estudiantes que participaron en el grupo formado con roles explicados mediante animaciones lograron desarrollar un mejor marco para la comprensión del tema. El 35% de los estudiantes de dicho grupo, utilizaron en el examen final roles de variables en sus respuestas a pesar de que esto no les fue solicitado ni eran problemas similares a los del curso en los que se había introducido el concepto de rol de variable. El uso de roles en las respuestas mejoró el vocabulario y reflejó que dichos estudiantes desarrollaron un mejor marco conceptual que les permitió procesar la información sobre el programa en una forma similar a otros programadores más experimentados. En base a esto último entienden que dichos estudiantes son capaces de crear estructuras de programas más complejas, reflejo de una mejor comprensión de la temática. Por esto, Byckling y Sajaniemi proponen en su trabajo comparar la calidad de los programas realizados por estudiantes con y sin formación con roles de variables.

El fundamento detrás de la investigación realizada es que la utilización de roles implica un conocimiento estratégico de su uso en la programación. Para un programador inicial, escribir un programa desde el principio puede ser una tarea muy difícil y analizar la transformación de los datos en términos de entrada/salida mediante roles permite tener una noción inicial de como hacerlo.

B. Como formador de esquemas

La teoría de esquemas [6] argumenta que las personas empaquetan experiencias similares en estructuras cognitivas llamadas esquemas. Los esquemas son abstracciones de casos concretos experimentados y juegan un papel fundamental en la forma en que se almacena, organiza y entiende la información. Un esquema es disparado y ejecutado concientemente o inconcientemente para resolver nuevos problemas similares. Entonces, es importante ayudar a los estudiantes a construir estos esquemas. Un procedimiento canónico es un procedimiento para resolver un problema que es disparado más o menos de forma automática por el problema. Estos están relacionados con los esquemas, por lo que fomentar la creación de ellos ayudará al estudiante a desarrollar estos procedimientos. La carga cognitiva está relacionada con la idea de que nuestra memoria para trabajar es limitada respecto a la cantidad de información que puede almacenar y la cantidad de operaciones que puede hacer

respecto de esta información. Una carga cognitiva excesiva dificulta la formación de esquemas. Entonces, los dos objetivos de formar esquemas y manejar la carga cognitiva están relacionados.

Muchos trabajos en Educación en Ciencia de la Computación pueden ser vistos como apoyo a la formación de esquemas haciendo explícitos los patrones que ocurren en los programas. Uno de estos enfoques son los roles de variables que ayudan a la construcción de programas convirtiéndose en una buena herramienta para promover la creación de esquemas en el estudiante.

C. Python y el uso de roles de variables

En varias universidades se ha adoptado Python [7] como primer lenguaje de programación para enfrentar el abandono de los cursos y el decreciente interés que se presenta frente a las ciencias de la computación. Las principales características que hacen resaltar al lenguaje son que resulta divertido para el programador, tiene una sintaxis simple, la longitud de los programas es más corta en comparación con los lenguajes tradicionales y desarrollar un programa, en promedio, lleva la mitad de tiempo que llevaría construirlo en otros lenguajes. Un argumento en contra es la diferencia de sintaxis con lenguajes como C++ y Java, sin embargo, en muchos casos dicha sintaxis está diseñada para reducir errores. Por otro lado, se critica el uso de espacios para representar bloques de código, sin embargo si se indenta correctamente es igual a otro lenguaje, además se ha demostrado que ayuda a mejorar la comprensión del programa. Un punto altamente discutido es la falta de un estricto control de tipos, sin embargo, a pesar de ello Python tiene varias características para ser usado como primer lenguaje de programación. Por ello, un punto interesante introducido por Nikula et al. [7] es potenciar el aprendizaje de la programación con el lenguaje Python utilizando roles de variables. En su artículo describen una adaptación de los roles definido por Sajaniemi al lenguaje debido a sus características. Los roles *most-recent-holder*, *most-wanted-holder*, *gatherer*, *follower*, *one-way-flag*, *temporary* y *walker* son exactamente los mismos. Sin embargo, los siguientes roles presentan diferencias:

- En lo referente al rol *fixed-value*, se debe agregar un nuevo rol llamado *constant* para variables que son asignadas con un valor literal y utilizar el primero para variables cuyo valor depende de una entrada pero su valor no cambia luego de que fue inicializada.
- En lo referente al rol *stepper* se debe tener en cuenta que una variable es siempre un *stepper* si es la variable de un bucle for y a veces aparece en un bucle while en circunstancias particulares. Esto significa que variables que pueden clasificarse como *most-recent-holders* en otros lenguajes, en Python son *steppers*.
- No resulta claro si hay una diferencia marcada entre los roles *containers* y *organisers* para realizar la distinción, aunque los autores sugieren solamente utilizar el rol *container*.

V. RESUMEN DE EXPERIENCIAS REALIZADAS

A. Conocimiento de las variables en cursos iniciales

Samurçay[4] realiza en su artículo la presentación de los resultados obtenidos en un estudio realizado con el objetivo de comprender cuál es el concepto que tiene un estudiante inicial de programación sobre las variables.

Para esto, se presentó a 26 estudiantes de bachillerato que se encontraban culminando su primer curso de programación en Pascal, programas con líneas faltantes así como información de la tarea que debe realizar el programa. Las líneas quitadas fueron elegidas con el objetivo de observar si existe una transferencia desde los conocimientos familiares de resolución de problemas hacia la programación.

Uno de los principales problemas observados con el manejo de variables, confirma una de las hipótesis del autor, al faltar casi todas las inicializaciones de las variables. Esto se debería a la falta de inicialización que existe en la resolución "manual" del problema, algo que se apoya en el resultado obtenido de que los valores que deben ser leídos se omiten menos que aquellos que deben ser asignados y que es consecuente con la explicación de la transferencia del terreno de los conocimientos de resolución de problemas al de la programación.

Algunas de las más importantes conclusiones sobre la experiencia de Samurçay respecto a las variables pueden resumirse en:

- Existe un mejor entendimiento para aquellos aspectos de las variables en los cuales el estudiante posee una noción fuera del campo de la programación.
- La inicialización en variables es un concepto cognitivo más complejo que los otros.

Los resultados de su trabajo de experimentación tiene una fuerte conexión con los ejercicios propuestos a los estudiantes, siendo muy recomendable acceder al trabajo original para tener una noción más amplia de los resultados planteados.

B. Uso de rol de variable en la Helsinki University of Technology

Sorva et al. [6] desarrollaron un experimento en dos cursos universitarios (Helsinki University of Technology) describiendo la forma en que los roles de variables son utilizados en la enseñanza y el método que utilizaron para analizar la reacción de los estudiantes:

- El objetivo del curso "CS1" es introducir a los estudiantes en la programación orientada a objetos utilizando Java. En este curso se presentan los roles como un aspecto de las variables a ser considerado como lo es el tipo y el alcance. La introducción a los roles no se hizo de acuerdo al sugerido por Kuittinen y Sajaniemi, sino que se presentan de acuerdo a como está estructurado el material. Siguiendo la sugerencia de Sajaniemi cada vez que un nuevo rol es encontrado, se lo define a los estudiantes informalmente con énfasis en las diferencias con el rol encontrado anteriormente. En las tareas de programación, se les enseña que realicen el pasaje de la idea al código ayudándose de los roles, discutiendo los mismos luego que se comienza con el pseudo-código de la solución.

- El objetivo del curso "CS2" es enseñar estructuras de datos y algoritmos independientemente del lenguaje de programación. En este curso los roles de variables no son un objetivo de enseñanza, sino que son introducidos brevemente en tanto aparecen en los ejemplos sin enfatizar en ellos. Además, se les brinda a los estudiantes material opcional que pueden consultar si lo desean.

A continuación se exponen los resultados de cada experiencia:

Resultados en curso "CS1": La tarea de los instructores de adaptar todo el material del curso para incluir el concepto de rol de variable resultó fácil. Les permitió saber las similitudes entre los distintos ejemplos y cómo los patrones de programación pueden ser aplicados en tareas similares utilizando la misma combinación de roles. Además, les permitió documentar cómo distintas partes del código que hacen uso de la misma variable trabajan juntas para lograr la tarea. Todo esto ayuda al estudiante en la formación de esquemas y procedimientos canónicos. La utilización de roles también se aplicó convenientemente a los programas orientados a objetos, clarificando el uso de las variables miembros. La introducción a nuevos roles en algunas lecturas no se hizo gradual sino directamente, por ejemplo, en un mismo ejemplo se presentaron dos nuevos roles. Posiblemente, esto causó una alta carga cognitiva en los estudiantes, lo cual se puede evitar si se re-estructuran los ejemplos para que se presenten de forma paulatina. En la experiencia se reportó que los estudiantes perciben de forma distinta a las variables miembros y locales por más que se trate del mismo rol. Los autores no están decididos si se debe enfatizar dicha similitud a los programadores novatos porque puede resultar potencialmente confuso. Otra importante desventaja es que el aprendizaje de los roles implica un mayor manejo de vocabulario por lo que puede aumentar la carga cognitiva del estudiante. Sin embargo, en el curso se utilizaron los roles para enseñar el uso de las variables, lo cual se hubiera hecho de todas formas sin ellos y le permite al instructor contar con mayor vocabulario para la discusión sobre el tópico. De todas formas, los autores creen que cualquier carga cognitiva que se pudo ganar en ello, se compensa con el hecho de que los roles clarifican el comportamiento de los programas. Las tres cuartas partes de los estudiantes encontraron útiles la inclusión de los roles de variables ayudándolos a entender los códigos dados como ejemplos. Los resultados sugieren que los roles de variables pueden ayudar a los estudiantes a ganar mayor conocimiento al programar. Estos sienten que los roles son más útiles para comprender un programa que para construirlo, lo cual está de acuerdo con lo demostrado por Byckling y Sajaniemi respecto a que introducir roles sin hacer uso de simulaciones de programas basados en ellos tiene un impacto limitado en el desarrollo de habilidades para construir programas.

Resultados en curso "CS2": Desde el punto de vista del instructor, la inclusión de roles de variables en el curso ayudó a seguir desarrollando los materiales y el proceso de enseñanza al crear nueva documentación al respecto y re-examinar los códigos de ejemplos ya escritos. En el caso de los nombres de variables, ayudó a discutir la forma en que son utilizadas las variables y como su uso se refleja en su nombre. También permitió detectar problemas en los códigos donde

algunas variables tenían un comportamiento ambigüo. Una misma variable era usada para distintos propósitos y cambios pequeños en el código permitieron clarificar estos ejemplos. En la perspectiva de los estudiantes, solo la tercera parte de ellos consideró los roles de variables útiles en el curso y unos pocos estudiantes utilizaron nombres de las variables según su rol en las respuestas a las evaluaciones. En este curso de introducción casual no se observó un uso generalizado del concepto lo cual se debe, principalmente, al poco tiempo y énfasis que tuvieron de enseñanza del mismo. Una explicación alternativa, es que los algoritmos utilizados tenían un uso fuerte de nombres de variables relacionados con el algoritmo en sí (por ejemplo, *pyvite*).

C. Uso de rol de variable en tres instituciones universitarias

Nikula et al. [7] presentan las siguientes experiencias en distintas universidades en cursos de iniciación a la programación. Las tres experiencias utilizan como lenguaje de programación Python y el uso de roles de variables.

Curso en Joensuu: El curso de introducción a la programación de la Universidad de Joensuu tiene como objetivo enseñar los conceptos básicos de la programación imperativa y que todos disfruten del aprendizaje. Se utiliza Python y un programa animado basado en roles que muestra la ejecución de un programa (*PlanAni*). Para lograr que los estudiantes disfruten de la tarea, se proponen por ejemplo tareas de programación totalmente abiertas donde solamente se les indica la primer línea de código lo que permite que utilicen su creatividad y comparen las distintas ideas presentadas por otros. Al finalizar el curso, la mayoría de los estudiantes se mostraron satisfechos con lo aprendido y aprobaron la prueba final.

Los roles fueron introducidos de manera gradual durante el curso y los ejemplos de programas fueron diseñados para ello. De los seis roles introducidos en el curso, los estudiantes reportaron que los roles *fixed-value*, *most-recent-holder* y *temporary* les resultaron más fáciles que *stepper*, *gatherer* y *most-wanted-holder*. Esto puede estar relacionado al tiempo que tuvieron para trabajar con los mismos, dado que los últimos fueron presentados durante la última mitad del curso, mientras que los primeros al principio.

Los docentes percibieron que gracias a las características de Python los estudiantes pudieron construir programas fácilmente con mínimas nociones de sintaxis en contraste con Java, utilizado en años anteriores. Por otro lado, el uso de *range()* en bucles *for* resultó difícil de entender.

Curso en Lappeenranta: En el curso introductorio a la programación de la Universidad Tecnológica de Lappeenranta, para enfrentar el bajo desempeño de los estudiantes, se decidió cambiar el lenguaje de programación desde C a Python y además incluir el concepto de rol de variable. También se utilizó la herramienta *PlanAni* para introducir el concepto.

Respecto al cambio de lenguaje, la mayoría de los estudiantes reportaron que lo encontraron más fácil, claro e interesante. Esto alentó a que más estudiantes se inscribieran en el curso. Los roles de variables fueron explicados por separado a medida que aparecían en los ejemplos de programas. Luego

de esto, se les aconsejaba a los estudiantes que siguieran su estudio por su cuenta y consultaran. El resultado de las evaluaciones fueron considerablemente mejores que el año anterior y además se reportó una baja significativa de abandono por parte de los estudiantes. En el lenguaje C los estudiantes reportaban como temas difíciles la entrada y salida, sin embargo, este año fueron los temas que resultaron más fáciles. En general, los docentes y estudiantes vieron de forma positiva la inclusión del concepto de rol destacando que ayudó al aprendizaje, sin embargo, al ser la primer experiencia no se pueden sacar conclusiones contundentes.

Curso en Blandford: El curso introductorio a la programación en la "Royal School of Signals" en Blandford enseña programación procedural y orientada a objetos con Python.

En cursos anteriores se utilizaba Delphi y los estudiantes vieron de forma positiva el cambio encontrando el material más sencillo. También, se trabajaron los roles como una ayuda para entender los diferentes uso de las variables. Como consecuencia, los estudiantes hacían uso de los nombres de los roles al hablar de los programas lo cual los ayudaba a describir la estructura del mismo. Sin embargo, la habilidad de utilizar roles de forma correcta en la construcción de programas resultó bastante limitada. Esto se puede deber a que durante el curso no se utilizó ninguna herramienta de animación para presentar los roles.

D. Uso de roles y las mejoras en las cualidades del programador

Byckling y Sajaniemi [1] presentan en su artículo un experimento con el objetivo de validar la hipótesis de que utilizando roles de variables en la enseñanza, los estudiantes aprenden más fácilmente programación.

Para esto en un curso introductorio a nivel universitario dictado en Pascal, se dividió la masa estudiantil en tres grupos que recibieron formaciones diferentes: unos fueron instruídos de la forma tradicional para dicho curso mediante clases y ejercicios tradicionales de *debugging* en Turbo Pascal (el grupo "*tradicional*"), otro grupo que fue formado con clases magistrales utilizando roles en las lecciones de forma sistemática (el grupo "*roles*") y por último el grupo que recibió las mismas clases, pero en donde se utilizaron también animaciones para las lecciones (el grupo "*animación*"). Sajaniemi y Kuitten analizaron los resultados de las pruebas finales para dicho curso según los grupos, cuyos resultados fueron comentados anteriormente, en la sección IV-A. Al final del curso, los autores realizaron un estudio para lo cual utilizaron voluntarios de cada uno de los grupos, a cambio de una pequeña compensación.

El estudio consistió en la elaboración en parejas de un pequeño programa de prueba consistente en tres sub-tareas, mientras eran monitoreados por cámaras. El programa consistía en la conversión de un tiempo leído en horas y minutos a minutos únicamente, con algunas restricciones adicionales.

Según los autores, la solución óptima a dicho problema debiese de utilizar 5 variables con roles definidos (2 *most-recent holders*, 1 *transformation*, 1 *most-wanted holder* y 1 *one-way flag*). Esto no implica que una solución correcta deba utilizar obligatoriamente este conjunto de roles.

La evaluación abarca varios aspectos, dos de ellos son la correctitud de la implementación realizada y la claridad del programa. Otro aspecto evaluado y discutible, es la cobertura de los roles de variables provista por los diferentes grupos. Esto es, cuáles fueron los grupos que utilizaron todos los roles necesarios en la solución óptima. Es opinión de los autores de este trabajo que a *prima facie* dicha evaluación no tiene mayor sentido sobre aquellos estudiantes que nunca fueron formados en roles (grupo "*tradicional*"). Quizás sí es más importante la comparación entre estudiantes de "*animación*" y "*roles*". Los resultados de las otras dos evaluaciones sobre los programas realizados (correctitud semántica del programa y el orden en el que fue escrito) avalan parcialmente la hipótesis de que estudiantes aprenden más fácilmente programación mediante roles con animación. Las únicas soluciones que funcionaban correctamente salieron todas de dichos grupos y fueron las soluciones más claras, con mayor correctitud semántica. A pesar de que los autores reconocen que no es posible hacer una inferencia estadística dada la poca cantidad de estudiantes en el curso, también señalan las notables diferencias de los resultados entre los estudiantes de los diferentes grupos, siendo los más destacados aquellos estudiantes en el grupo de "*animación*".

VI. CONCLUSIONES Y TRABAJO FUTURO

EN este trabajo se ha presentado un estudio de la problemática en el aprendizaje del uso de la variable que presentan los estudiantes, utilizando artículos de diferentes autores. Se ha demostrado que es un concepto difícil de interiorizar debido principalmente a las distintas caracterizaciones con las que cuenta el concepto de variable, según el problema en el que está inmerso.

Es importante destacar que existe un problema potencial para la enseñanza de las variables en la programación si se utiliza como base la concepción de variable con la que cuenta el estudiante de su formación en matemática y la misma es incorrecta. Esto es de importancia en contextos como el de Facultad de Ingeniería (UdelaR) en donde los estudiantes actualmente entran con una formación deficiente en ciencias básicas en un número no despreciable. Resultaría relevante que existiera una formación adicional sobre las variables en matemática previo a la formación en programación en caso de utilizarse como base las variables literales del álgebra.

La utilización de roles de variables en el aprendizaje se presenta como una herramienta potencialmente valiosa para introducir en los cursos de introducción a la programación con el objetivo de ayudar en la correcta interiorización del concepto. A través de las experiencias descritas, se ha mostrado cómo ha ayudado a crear y comprender programas. Se debe tener en cuenta que los roles resultan más provechosos para los estudiantes si son introducidos de forma uniforme en todos los aspectos del curso: materiales, discusiones y especialmente con el uso de una herramienta de animación basada en roles.

Además, se describió cómo los roles de variables son una herramienta que permiten a los instructores ayudarse

de ella para fomentar la formación de múltiples modelos mentales y patrones de solución de problemas generales en los programadores novatos. Sin embargo, es importante considerar la carga cognitiva que el estudiante genera al adoptar estos nuevos conceptos aunque se cree que cualquier costo inicial relacionado con ello se supera con la formación de esquemas a largo plazo.

Desde el punto de vista docente, se mostró cómo la inclusión del uso de roles, ayudó a seguir desarrollando los materiales y detectar errores referente a comportamientos ambiguos que las variables presentaban en el material ofrecido a los estudiantes.

Por otro lado, resulta alentador los resultados expuestos en relación a utilizar Python como primer lenguaje de programación combinado con los roles. Además de que los estudiantes obtuvieron mejores resultados, se incrementó su satisfacción con lo aprendido lo cual disminuyó la cantidad de abandono de los cursos.

Dado que la problemática que tienen los estudiantes de Facultad de Ingeniería (UdelaR) es similar a la aquí expuesta y que las experiencias reportaron beneficios tanto para el estudiante como para el docente, resulta prometedor elaborar un diagnóstico de dicha situación en función de aplicar roles de variables en el primer curso de programación. Además, resultaría beneficioso considerar el costo de migrar el lenguaje de programación de Pascal a Python favoreciendo el aprendizaje y la motivación de los estudiantes.

REFERENCIAS

- [1] P. Byckling and J. Sajaniemi, "Roles of variables and programming skills improvement," in *SIGCSE*, D. Baldwin, P. T. Tymann, S. M. Haller, and I. Russell, Eds. ACM, 2006, pp. 413–417. [Online]. Available: <http://doi.acm.org/10.1145/1121341.1121470>
- [2] J. Sajaniemi, "An empirical analysis of roles of variables in novice-level procedural programs," in *Proceedings of IEEE 2002 Symposium on Human Centric Computing Languages and Environments (HCC'02)*, Arlington, VA, Sep. 2002, pp. 37–39.
- [3] S. Wagner, "What are these things called variables?" *Mathematics Teacher*, pp. 474–479, October 1983.
- [4] R. Samurçay, "The concept of variable in programming: Its meaning and use in problem-solving by novice programmers," in *Studying the Novice Programmer*, E. Soloway and J. C. Spohrer, Eds. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers, 1989, ch. 8.
- [5] L. M. P. y José Luis Díaz Gómez, "Concepto de variable: Dificultades de su uso a nivel universitario," *Mosaicos Matemáticos*, vol. 11, 2003.
- [6] J. Sorva, V. Karavirta, and A. Korhonen, "Roles of variables in teaching," *JITE*, vol. 6, pp. 407–423, 2007. [Online]. Available: <http://www.jite.org/documents/Vol6/JITEv6p407-423Sorva280.pdf>
- [7] U. Nikula, J. Sajaniemi, M. Tedre, and S. Wray, "Python and roles of variables in introductory programming: Experiences from three educational institutions," *JITE*, vol. 6, pp. 199–214, 2007. [Online]. Available: <http://www.jite.org/documents/Vol6/JITEv6p199-214Nikula269.pdf>
- [8] D. Kchemann, "Children's understanding of numerical variables," *Mathematics in School*, vol. 7, no. 4, pp. 23–26, September 1978. [Online]. Available: <http://www.jstor.org/stable/30213397>