

La programación lúdica como estrategia de articulación entre niveles

Marcela F. López, Paola del Olmo, Carina Reyes, Eduardo F. Fernández

Universidad Nacional de Salta - Facultad de Ciencias Exactas

mfflopez@unsa.edu.ar - pdelolmo@unsa.edu.ar - reyescarina@gmail.com - effer@cidia.unsa.edu.ar

Palabras claves: Articulación - Programación - Tutoría de Pares - Aprendizaje Social

Resumen

El presente trabajo describe el proyecto "Jugá a ser programador", perteneciente al programa de voluntariado universitario, aprobado por Resolución de la Secretaría de Políticas Universitarias de la Nación N° 2333.

Fundamentamos la elaboración del mismo en la preocupación permanente como grupo interdisciplinario de investigación comprometido en problemáticas de ingreso y permanencia a los estudios superiores. En este sentido reconocemos que en el ingreso a la carrera Licenciatura en Análisis de Sistemas de la Universidad Nacional de Salta (UNSa.), se registra una baja sistemática en el índice de ingreso. Por otra parte los ingresantes se encuentran desorientados, sin una comprensión clara en lo que se refiere al perfil de programador, como futuro campo de estudio y de ejercicio profesional.

Este proyecto busca acercar a los alumnos del último año de nivel secundario a una experiencia de programación de animaciones y juego en 3D, usando los software Alice y Diagramar.

Se plantea resignificar el impacto que esta experiencia pueda generar en la toma de decisiones que los estudiantes deben realizar al momento de su egreso del nivel medio, respecto a continuar sus estudios en carreras universitarias y particularmente en carreras del área de informática.

Introducción

Las Universidades Argentinas enfrentan hoy problemáticas preocupantes con respecto a la retención y la permanencia de los alumnos ingresantes a las carreras científico-tecnológicas en general y a las vinculadas con la disciplina Informática en particular. Esto ocurre en un contexto de cambio del nivel secundario, en el que la provincia presenta una adecuación de los contenidos curriculares correspondientes al área de Informática. Otro acontecimiento importante y simultáneo es el desarrollo del Programa Conectar Igualdad, cuya finalidad es fortalecer las capacidades tecnológicas de las instituciones educativas secundarias de la región.

Desde el Programa de voluntariado se procura ofrecer a los estudiantes poseedores de las netbooks, situaciones que permitan una finalidad de uso esencialmente académica en un entorno de trabajo lúdico y de alta interacción a través de los mecanismos masivos de comunicación que brinda Internet.

Sabemos que la deserción no sólo responde a cuestiones socioeconómicas. Entran en juego la diversidad cultural, problemáticas de orden personal, la carencia de conocimientos, la falta de estrategias cognitivas y metacognitivas de abordaje de los mismos, la falta de adecuación a los tiempos de estudio universitarios. También es necesario tener en cuenta los problemas generados por las diferencias respecto de aprendizajes y conocimientos previos. Por tanto, desde este Proyecto se intenta articular contenidos y estrategias cognitivas para el estudio de la Programación. Se espera que esto disminuya la brecha entre niveles educativos logrando la ansiada

articulación y facilite la permanencia y la trayectoria exitosa en el Primer Año.

Descripción del proyecto

Las ofertas educativas del Nivel Medio de la provincia vinculadas con la Informática, son escasas y en la mayoría de los casos, están más dirigidas al estudio del hardware que del software. La industria reconoce la insuficiencia de recursos humanos en TIC. Según datos del Observatorio Permanente de la Industria de Software y Servicios Informáticos de la Argentina (OPSSI - CESSI) los graduados cubren apenas un poco más de la mitad que los aproximadamente 7000 graduados que demanda la industria. Este fenómeno puede explicarse – en parte- a la modalidad de enseñanza de la Programación, basada en la práctica mecánica y casi memorística de aspectos sintácticos de un lenguaje de programación y su aplicación al desarrollo de programas ajenos a los intereses de los estudiantes. Para revertir esta situación, planteamos como objetivo del proyecto favorecer una mejor competencia para la Programación, aplicando como estrategia el potencial de conocimiento y creatividad a través del juego, en una red de pares y profesionales programadores.

Para lograr este objetivo nos proponemos las siguientes metas:

- 1 - Que el alumno secundario reconozca la Programación como experiencia placentera y divertida, produciendo aplicaciones multimediales relacionadas a sus intereses, socializándolas en red.
- 2 - Que cada estudiante y docente universitario se comprometa a sostener su rol de tutor sobre el alumno del Nivel Medio, atendiendo los aspectos formativos y afectivos de esa función.
- 3 - Que el equipo contribuya a desmitificar las carreras informáticas como ámbitos de trabajo colaborativo y no solitario ni individualista.

Para el logro de este objetivo y metas convocamos a un grupo de estudiantes del último año de nivel secundario para realizar tareas de programación mediante el desarrollo de animaciones y juegos. Este desarrollo se sostendrá desde la interacción entre pares con el apoyo tutorial de alumnos avanzados y docentes-investigadores de la carrera Lic. en Análisis de Sistemas de la UNSa. Los procesos de desarrollo y las producciones de los estudiantes se socializan en una red, aprovechando la conectividad y accesibilidad de los estudiantes a la WEB 2.0.

Como herramienta de desarrollo utilizaremos la aplicación Alice, la cual permite la creación de animaciones y juegos de forma sencilla. Esta aplicación permite que el alumno se acerque a la Programación Orientada a Objetos e impulsada por eventos, de manera intuitiva. Sin embargo, para consolidar los saberes corresponde formalizarlos. Con este fin, presentaremos Diagramar para profundizar los aspectos relativos a la estructura de los métodos, aplicando conceptos básicos de la Diagramación y Programación.

La tutoría entre pares como pilar fundamental

Entendemos la tutoría como proceso de interacción encuadrada en una concepción de “relación de ayuda”. Entendemos por ayudar como el acto de promover en una persona un cambio constructivo en el comportamiento, descubrir nuevos aspectos y posibilidades en sí mismos.

Desde el Proyecto fundamentamos la Tutoría en la Teoría Socio-histórica de Vigostky ya que la misma está ligada a una asimetría entre sus miembros. Alguien que conoce y resuelve de manera autónoma, un experto y un novato que todavía no tiene la capacidad de resolver de manera autónoma. Ambos se vinculan mediante la acción de “**apuntalamiento**”.

Este apuntalamiento se visualiza como andamio, en el sentido estricto de la palabra en el marco de la Teoría.

El andamiaje es un proceso mediante el cual el experto proporciona un apoyo que le permite al novato avanzar más allá de lo que hubiera podido hacer por sí solo es decir de manera autónoma.

Esta relación de ayuda de apuntalamiento no puede prolongarse indefinidamente en el tiempo. Por ello debe poseer las siguientes características:

- Ajustable: De acuerdo con el nivel de competencia del sujeto menos experto y de los progresos que se produzcan.
- Temporal: No debe ser crónico, debe otorgar autonomía gradualmente.
- Audible y visible: Debe ser un dispositivo explícito.

Desde este andamiaje se genera un vínculo de acompañamiento, estimulando el diálogo, el intercambio, para que sea un encuentro de sentidos. El sentido principal de la función tutorial es ofrecer al estudiante herramientas apropiadas que estimulen su autonomía, orienten y faciliten su aprendizaje **significativamente**.

Desde el proyecto se trabajó con los miembros del voluntariado para que se posicionen teórica y prácticamente en las Tutorías, respetando el sentido que se le dio en el marco del Proyecto.

Siguiendo esta línea, se establecieron una serie de encuentros presenciales entre tutores pares y estudiantes secundarios con el fin de presentar los objetivos del proyecto y las aplicaciones para la producción de software, Alice y Diagramar. Durante estos encuentros se acordarán pautas para el desarrollo de las animaciones o juego que los estudiantes secundarios diseñen. Proponemos estos encuentros presenciales con la finalidad de favorecer y fortalecer la cohesión del grupo y sostener la participación del tutor como referente en el seno del grupo. Se espera que estos encuentros presenciales refuercen la participación de los estudiantes desde la

virtualidad ya que los mismos fueron formados en un sistema educativo presencial y cuentan con poca experiencia del uso de los recursos de la WEB 2.0 para fines académicos.

Durante el proceso de desarrollo de sus producciones, los estudiantes contarán con la asistencia de los tutores pares desde la distancia, utilizando para ello medios de comunicación previstos desde un aula virtual, como son los foros y la mensajería interna y otros disponibles en la WEB 2.0. El objetivo es sostener comunidades virtuales donde prime el espíritu colaborativo y solidario y se oriente hacia el aprendizaje significativo para intercambiar el uso de herramientas de programación. En este sentido, las tutorías virtuales sostienen y enfatizan el aprendizaje autónomo de los estudiantes quienes, utilizando Alice como herramienta de desarrollo y desde un contexto de aprendizaje social y significativo, diseñarán y desarrollarán libremente sus aplicaciones. Estas producciones participarán en exposiciones que realizan anualmente en los colegios.

En este contexto, el tutor par, deberá motivar y sostener el aprendizaje de los alumnos, minimizando de esta forma la distancia transaccional, la cual, en términos de Moore se define por *"la distancia existente en las relaciones educativas, determinada por la cantidad y calidad del diálogo que tiene lugar entre el estudiante y el profesor y la estructuración que existe en el diseño del curso"*(Segura, J. 2000). El tutor virtual es concebido como un acompañante cognitivo quien debe fomentar la participación y el espíritu colaborativo, desde el cual cada estudiante se compromete no sólo en su propia formación, sino también en la formación de sus compañeros. Este proceso lleva a que el estudiante adquiera gradualmente mayor autonomía en sus procesos de aprendizaje, que devengan en procesos reflexivos críticos. El tutor par resulta una pieza importante en la configuración del proyecto ya que éste se constituye en referente del estudiante secundario y entre sus funciones está la de

presentar a los mismos la cultura universitaria, los hábitos y estrategias de estudio necesarios en ese nivel de aprendizaje.

Alice como aplicación para el desarrollo creativo

Alice es un lenguaje de programación educativo de acceso libre y abierto con un innovador entorno de desarrollo integrado (IDE) de programación 3D. Esta herramienta fue desarrollada en la universidad de Carnegie Mellon como parte de un proyecto de realidad virtual, con apoyo de la Fundación Nacional de Ciencia americana, DARPA e Intel entre otros colaboradores.

Alice está diseñado únicamente para enseñar a programar. Intenta atraer usuarios que no se enfrentan normalmente a problemas de programación, tales como estudiantes de secundaria. Alice pretende facilitar el primer contacto de los estudiantes con la programación. Su IDE basado en arrastrar y soltar, presenta objetos 3D como: plantas, personajes, vehículos, entre otros en un mundo virtual. Además, el estado de estos objetos que puede cambiar gracias a métodos como “mover hacia adelante un metro” o “girar a la izquierda un cuarto de vuelta” siendo fácil de entender por cualquier estudiante.

Alice se basa en el paradigma de Programación Orientado a Objetos e impulsado por eventos. Permite a los estudiantes aprender conceptos fundamentales de programación dentro de un contexto de creación de animaciones ó videojuegos interactivos.

Para utilizar Alice los estudiantes no necesitan recordar ninguna sintaxis especial, sólo deben de arrastrar y soltar bloques gráficos. De esta forma inicialmente pueden no conocer todas las reglas de un lenguaje de programación para realizar programas, ya que los elementos arrastrados al editor son siempre válidos permitiendo a los mismos ir comprendiendo el funcionamiento y el comportamiento de los

diferentes elementos disponibles para cada objeto que haya integrado a su mundo virtual. Así, los estudiantes irán desarrollando una intuición para la sintaxis.

Los estudiantes al usar Alice colocarán objetos en el mundo virtual y luego los harán interactuar según un guión en texto. Al ejecutar el código visualizarán cómo se animan los objetos. Así, conceptos abstractos como el incremento de una variable podrán convertirse en conceptos concretos al observar que un conejo se mueve hacia adelante en lugar de hacia atrás. Mediante la intuición y motivación por lograr realizar el guión y bajo este concepto intuitivo y visual que nos proporciona esta herramienta los estudiantes ganarán experiencia con todo tipo de estructuras típicas de un curso introductorio de programación.

Diagramar como herramienta de formalización

Diagramar es una aplicación desarrollada desde un Proyecto de Investigación "Una estrategia metodológica: el uso de las NTICs en el ingreso masivo universitario", acreditado y subvencionado por el Consejo de Investigación de la Universidad Nacional de Salta (UNSa.). Esta herramienta permite crear, editar y ejecutar diagramas de bloques, bajo la metodología de una programación estructurada y es utilizada desde el año 2009 por la cátedra de Elementos de Programación, materia de primer año de las carreras de Licenciatura en Análisis de Sistemas y de Tecnicatura Universitaria en Programación que se dictan en la UNSa.

Presenta una interfaz intuitiva y fácil de utilizar, para ello se ha enriquecido la interfaz con iconos de aspecto profesional, que permiten deducir el funcionamiento de cada uno de ellos. La interfaz se encuentra dividida en paneles, bien diferenciados, donde cada uno de ellos tiene un papel importante en el diseño y ejecución de los diagramas de bloques.

La herramienta presenta barras con iconos que se deben seleccionar y arrastrar al panel de edición.

La primer barra de iconos presenta las estructuras de control simples (Entrada, Salida, Asignación, Alternativa, Ciclo de repetición condicionada y Ciclo de repetición incondicional) que permiten diseñar un algoritmo representándolo a través de un Diagrama de Bloques.

También propone trabajar el diseño de un algoritmo, utilizando el concepto de componentes, entendiéndose como componente a la agrupación estructuras de control simples, que resuelven un proceso puntual. La característica principal del componente es que se puede generalizar, para reutilizarse en otras soluciones algorítmicas.

La herramienta propone dos barras más, en donde se encuentran los componentes para manipular vectores y los componentes para manipular Tablas.

El programa permite guardar un diagrama de bloques en un archivo con extensión deb (Diagrama de bloques), también presenta la posibilidad de exportar el diagrama a un formato gráfico (bmp o jpg) que permite compartirlo con otros usuarios.

Metodología Alice + Diagramar - Diagramar + Alice

Como se indicó anteriormente, los estudiantes de nivel secundario utilizarán Alice para el desarrollo de sus aplicaciones, por lo que la programación que realizarán se basa en el paradigma de Programación Orientado a Objetos e impulsado por eventos. Esta programación se realiza de una manera intuitiva a través de la interfaz que presenta Alice.

La metodología de trabajo consistirá en presentar Alice mediante un pequeño juego como ejemplo, pero que contenga los conceptos básicos y necesarios que debe

conocer el estudiante para desarrollar luego su aplicación, entre los cuales se destaca el concepto de objetos reconocidos como actores del mundo que se está creando, la programación métodos para agregar funcionalidad a los actores, el uso de estructuras de control, variables y parámetros y el manejo de eventos. Para este cometido destacamos la conveniencia de crear un guión que permita planificar la secuencia de las escenas y las actividades de los actores.

Durante la programación del juego, los estudiantes utilizarán los recursos que provee Alice de forma intuitiva, sin embargo, el proceso de construcción será guiado y monitoreado por el tutor.

Para lograr que el alumno reconozca las estructuras formales que subyacen en el programa desarrollado de una manera reflexiva nos apoyaremos en la aplicación Diagramar, a partir de la cual se analizará el diagrama resultante.

El siguiente ejemplo ilustra un juego cuya historia o guión se narra a continuación:

"Un conejo debe recorrer un camino recto. En este camino se encuentran cuatro zanahorias y tres piedras. Las zanahorias tienen asociadas un tiempo de permanencia en el camino registrado mediante un número al azar. El propósito del juego consiste en que el conejo salte sobre cada zanahoria y la coma antes que finalice el tiempo asociado a cada una de ellas.

Para el avance del conejo se programaron la tecla derecha (RIGHT) que permite que el conejo realice saltos lineales de una unidad de distancia y la tecla arriba (UP) la que permite sortear las piedras realizando un salto de dos unidades. Cuando el conejo se encuentra frente a una piedra se cambia la funcionalidad de la tecla RIGHT impidiendo el avance, provocando sólo el saltos en el lugar, mientras que, cuando el conejo se encuentra lejos de las piedras, se cambia la funcionalidad de la tecla UP, impidiendo que el conejo salte dos unidades de medida.

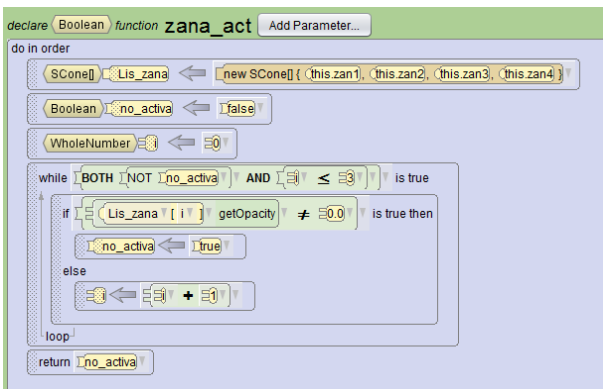
El juego finaliza cuando las cuatro zanahorias desaparecen ya sea porque el conejo las comió

o porque se agotó el tiempo de permanencia en el camino."

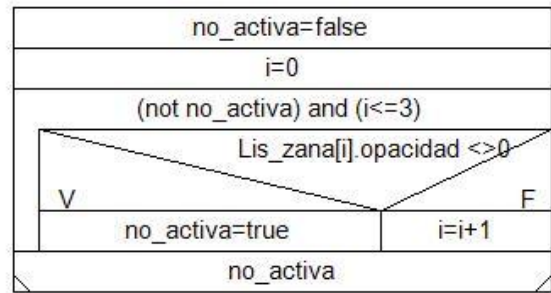
A partir de esta narrativa los estudiantes desarrollarán el juego, planteando en primer lugar el escenario y los actores y luego programando los métodos necesarios y los eventos para las teclas RIGHT y UP.

Con el propósito de reflexionar sobre las estructuras de control, se analizará el método **zana_act**. Este método pertenece al escenario y tiene como propósito indagar si alguna de las zanahorias continúan activas o todas están inactivas, retornando ese informe. Cuando una zanahoria está inactiva su opacidad es cero. El método es utilizado para controlar el fin del juego.

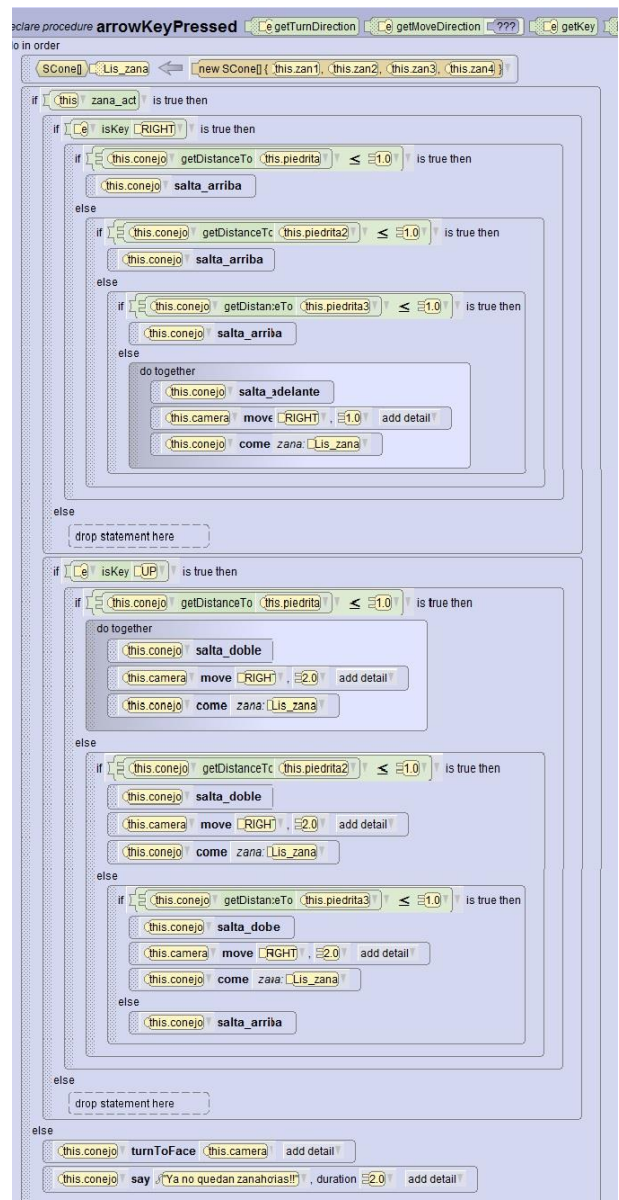
La imagen a continuación muestra la programación del método **zana_act**, utilizando Alice.



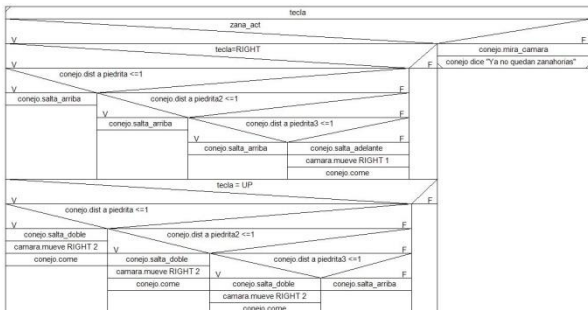
Con el fin de destacar las estructuras de control intervinientes en este método, se modelizará el algoritmo con la aplicación Diagramar. Cabe aclarar que siendo un método del escenario, no requiere parametrizar los objetos que representan a las zanahorias, a partir de los cuales se crea la variable Lis_zana como una lista de zanahorias, por esta razón en el diagrama de bloques se utiliza la variable Lis_zana sin mostrar su inicialización.



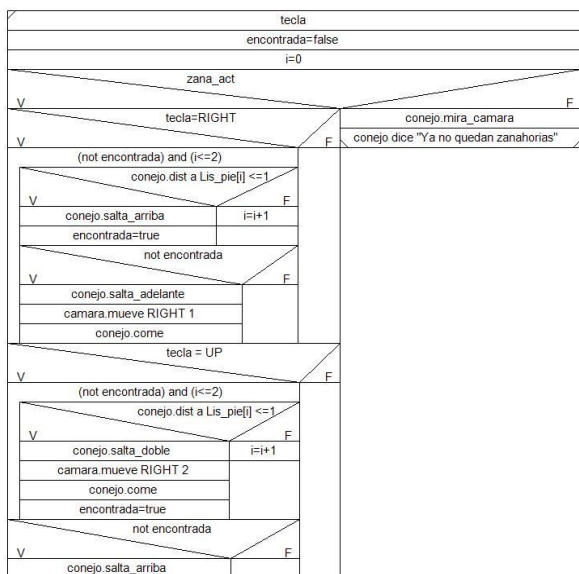
A continuación el tutor propondrá a los estudiantes el análisis de la programación del evento que controla las teclas de movimiento, el cual hace uso del método **zana_act** para controlar el fin del juego.



El tutor junto a los estudiantes utilizarán Diagramar para formalizar la lógica de este evento, resultando un diagrama similar al que se presenta a continuación.



A partir del diagrama y trasladando el razonamiento que se presentara y analizara previamente desde el método **zana_act**, se pretende reformular el diagrama del evento construyendo el siguiente diagrama.



Como puede observarse, en este diagrama se aplica como estrategia el uso del vector **Lis_pie** para contener los objetos piedras, tal como en el método **zana_act** se utilizó el vector **Lis_zana** para contener las zanahorias. A partir de la utilización de los vectores se puede simplificar las estructuras de control evitando el anidamiento de las alternativas e incorporando el uso de ciclos condicionados.

Con esto, el código en Alice resulta como sigue:

```

declare procedure arrowKeyPressed
do in order
  SConejo ← Lis_zana ← new SConejo ( this zan1, this zan2, this zan3, this zan4 )
  Boulder1 ← Lis_pie ← new Boulder1 ( this piedra1, this piedra2, this piedra3 )
  Boolean encontrada ← false
  WholeNumber i ← 0
if this zana_act is true then
  if isKey RIGHT is true then
    while BOTH NOT encontrada AND i <= 2 is true
      if this conejo getDistanceTo Lis_pie [ i ] <= 1.0 is true then
        this conejo salta_arriba
        encontrada ← true
      else
        i ← i + 1
    loop
    if NOT encontrada is true then
      do together
        this conejo salta_adelante
        this camera move RIGHT : 1.0 add detail
        this conejo come zana Lis_zana
      else
        drop statement here
    else
      drop statement here
  if isKey UP is true then
    while BOTH NOT encontrada AND i <= 2 is true
      if this conejo getDistanceTo Lis_pie [ i ] <= 1.0 is true then
        do together
          this conejo salta_doble
          this camera move RIGHT : 2.0 add detail
          this conejo come zana Lis_zana
        encontrada ← true
      else
        i ← i + 1
    loop
    if NOT encontrada is true then
      this conejo salta_arriba
    else
      drop statement here
  else
    this conejo turnToFace this camera add detail
    this conejo say "Ya no quedan zanahorias!" , duration 2.0 add detail
  
```

A partir de esta experiencia esperamos que los alumnos se acerquen a la programación con intención lúdica pero a la vez reflexiva respecto a los procesos cognitivos involucrados y sus expectativas relacionadas al campo profesional que esta disciplina involucra. Sostenemos que estas instancias de juego se presentan como situaciones potencialmente generadoras de estrategias que involucran procesos cognitivos por parte de los estudiantes, y tal vez desde un análisis más profundo se establezcan procesos de comprensión de los mismos, es decir lo metacognitivo.

Lo que se espera lograr

Desde este proyecto esperamos poder llegar a los estudiantes con una propuesta lúdica respecto a la programación pero a la vez realista de los que implica ser programador, avanzando desde lo intuitivo e informal hacia lo formal y preciso, donde se ponen en juego no sólo la experiencia del programador sino también la capacidad de análisis y reflexión que el profesional debe realizar, lo cual resulta conveniente desde la formalización y esquematización logrado a partir de herramientas como los diagramas de bloques.

Nos proponemos **conocer** el impacto que esta experiencia **puede** generar en la toma de decisiones que los estudiantes deben realizar al momento de su egreso del nivel medio, respecto a continuar sus estudios en carreras universitarias y medir la proporción de alumnos de nivel secundario participantes de la experiencia que se inscriben en carreras del área de informática.

Colateralmente, analizaremos también el desempeño de los alumnos universitarios en el rol de tutor par, su incidencia en la formación de alumnos secundarios, el impacto de esta experiencia en su concepción como futuros docentes y los requerimientos que surjan referidos a la formación y desempeño en tutoría.

Bibliografía

- Baquero, R. (1996) "Vigostky y el aprendizaje escolar" Ed. Aique. Buenos Aires.
- Elichiry, N. (2009) "Escuela y aprendizajes" Trabajos de Psicología Educativa. Ed. Manantial. Buenos Aires.
- Ferrari, C. (1994) "¿Qué es el juego? Alrededor del fútbol" Revista Universidad de Antioquía LXIII. Citado en Duarte D., Jakeline. (2003) "Ambiente de Aprendizaje. Una aproximación

conceptual". Revista Estudios Pedagógicos (online) N° 29.

- Gallino, Mónica y otros. (2007) "Análisis de los procesos de interacción en una propuesta virtual de formación docente". En www.utn.edu.ar/aprobedutec07/docs/16.doc.
- Moll, L. (1993) "Vigostky y la Educación" Connotaciones y aplicaciones de la Psicología Sociohistórica en la Educación. Ed. Aique. Buenos Aires.
- Segura, Jordi Adell y otros. (2000) "Enseñanza on line: elementos para la definición del rol del profesor" en libro "Las nuevas tecnologías para la mejora educativa" Comp. Cabero, Julio. Ed. Sevilla Kronos. Sevilla.