

# Approach toward creating recreational environments for the first CS1 session-An experience with digital natives

Marco Aedo López, Ing.<sup>1</sup>, Elizabeth Vidal Duarte, Mg.<sup>1</sup>, Eveling Castro Gutiérrez, Mg.<sup>1</sup>, Alfredo Paz Valderrama, Ing.<sup>1</sup>

<sup>1</sup> Universidad Nacional de San Agustín de Arequipa, Perú, [maedol@unsa.edu.pe](mailto:maedol@unsa.edu.pe), [eviald@unsa.edu.pe](mailto:eviald@unsa.edu.pe), [ecastro@unsa.edu.pe](mailto:ecastro@unsa.edu.pe), [apazv@unsa.edu.pe](mailto:apazv@unsa.edu.pe)

*Abstract– This study presents an approach toward creating recreational environments based on video games and 3D animations to provide students with a motivating environment for the first programming course. The course faculties of Peruvian universities believe that most students find computer programming completely new. Although most students already have basic knowledge about software, very few have experience with programming. This study proved that a visual approach based on video games and 3D animations to teaching programming concepts is more effective and motivating than traditional methods. This study describes our experiences with the Lightbot video game and Alice as a 3D environment in introducing programming concepts for teaching the CS1 course in the School of Systems Engineering at the National University of San Agustín de Arequipa, Peru, and considers the learning styles of digital natives.*

*Keywords–* CS1, teaching, motivation, recreational environments, pedagogical tools.

**Digital Object Identifier (DOI):**

<http://dx.doi.org/10.18687/LACCEI2017.1.1.249>

ISBN: 978-0-9993443-0-9

ISSN: 2414-6390

# Aproximación orientada a Entornos Lúdicos para la primera sesión de CS1 - Una experiencia con nativos digitales

Marco Aedo López, Ing.<sup>1</sup>, Elizabeth Vidal Duarte, Mg.<sup>1</sup>, Eveling Castro Gutiérrez, Mg.<sup>1</sup>, Alfredo Paz Valderrama, Ing.<sup>1</sup>

<sup>1</sup>Universidad Nacional de San Agustín de Arequipa, Perú, maedol@unsa.edu.pe, evidald@unsa.edu.pe, ecastro@unsa.edu.pe, apazv@unsa.edu.pe

**Abstract**– Este artículo describe una aproximación orientada a entornos lúdicos, basada en videojuegos y animaciones 3D, para brindar a los estudiantes un ambiente motivador para el primer curso de programación. Los educadores que enseñan tal curso en cualquier carrera profesional de una universidad peruana encuentran que para la mayoría de los estudiantes la programación de computadoras es un tema completamente nuevo; aunque casi todos los estudiantes tienen conocimiento de nivel de usuario de software, muy pocos han tenido alguna experiencia en la programación. Nosotros probamos que una aproximación visual basada en videojuegos y animaciones en 3D para enseñar conceptos de programación es más efectiva y motivadora. Aquí describimos nuestra experiencia usando un videojuego llamado Lightbot y Alice como entorno 3D, para introducir tales conceptos de programación. Esta experiencia ha sido implementada en la Universidad Nacional de San Agustín de Arequipa – Perú en su carrera profesional de Ingeniería de Sistemas, en el curso CS1 correspondiente y toma en consideración los estilos de aprendizaje de los nativos digitales.

**Keywords**— CS1; Enseñanza; Motivación; Entornos Lúdicos; Herramientas Pedagógicas

## I. INTRODUCCIÓN

Hay experiencias que han mostrado que la naturaleza visual y lúdica de los videojuegos son muy útiles para el aprendizaje de los estudiantes [1],[2],[3],[4]. Incluso hay una implementación de esta aproximación en Karel the robot [5], el cual usa clases predefinidas para introducir los conceptos fundamentales de programación.

El curso CS1, que en la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional de San Agustín [6] se denomina Fundamentos de Programación 1 – FP1, se empezó a enseñar como parte del plan de estudios 2013 y está centrado en el lenguaje de programación Java.

Durante la primera sesión del curso, a lo largo de los años, se nota la existencia de un grupo de estudiantes bastante heterogéneo, tanto en conocimientos como en motivaciones. Así, la Tabla I muestra los resultados de los 4 años en los que se viene dictando el curso, a la pregunta realizada el primer día clases de “¿cuántos de ustedes tienen alguna experiencia en programación de computadoras?”

TABLA I. RESPUESTAS A CUESTIONAMIENTO INICIAL

Año	Si	No
2013	5%	95%
2014	6%	94%
2015	5%	95%
2016	7%	93%

En la Tabla II se hace hincapié en diferentes niveles de experiencia en programación y se obtiene resultados más puntuales, descubriendo que al menos algunos conceptos simples, sin llegar al nivel de la creación de programas, son poseídos por los estudiantes del curso.

TABLA II. RESPUESTAS A CUESTIONAMIENTO INICIAL DETALLADO

Año	Ninguna	Crean fórmula en hoja de cálculo	Programa simple (tipo suma de 2 números)	Programa simple (tipo mayor de 2 números)	Programa medio (tipo ordenar arreglo de datos)
2013	40%	55%	3%	2%	0%
2014	45%	49%	4%	2%	0%
2015	35%	60%	2%	2%	1%
2016	40%	53%	3%	3%	1%

Sin embargo, en la Tabla III, ante la pregunta “¿quiénes utilizan videojuegos?”, las respuestas son casi lo opuesto a las respuestas obtenidas para la primera pregunta.

TABLA III. RESPUESTAS A UTILIZACIÓN DE VIDEOJUEGOS

Año	Si	No
2013	90%	10%
2014	92%	8%
2015	95%	5%
2016	97%	3%

La heterogeneidad mostrada en el grupo de estudiantes es tratada de homogeneizar basada en los estilos de aprendizaje de los nativos digitales y usando herramientas que ellos conocen

muy bien y utilizan cotidianamente: videojuegos y otros entornos lúdicos, tal como de animaciones en 3D.

En este artículo presentamos nuestra experiencia en el uso de las herramientas lúdicas: videojuego Lightbot [7] y complementado con un entorno de animación en 3D llamado Alice [8].

El resto del artículo está organizado de la siguiente manera. En la sección II se describe el primer curso de programación de nuestra currícula. En la sección III se muestra los estilos de aprendizaje de los nativos digitales en nuestro entorno en particular. En la sección IV se describe brevemente las herramientas que utilizamos, Lightbot y Alice. En la sección V se muestra el diseño de la propuesta de la primera sesión del curso, basándonos en el estilo de aprendizaje de nuestros estudiantes y usando las herramientas antes descritas. En la sección VI se muestran los resultados obtenidos del rendimiento de los estudiantes, donde se compara su rendimiento cuando utilizamos un enfoque tradicional de enseñanza con un enfoque lúdico de enseñanza. Finalmente se presentan nuestras conclusiones.

## II. EL PRIMER CURSO DE PROGRAMACIÓN

### A. Generalidades

El primer curso de programación en nuestra currícula es denominado Fundamentos de Programación 1, es dictado en el primer semestre de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional de San Agustín de Arequipa - Perú. El curso tiene una duración de 17 semanas y tiene 8 horas semanales (2 horas teóricas, 2 horas prácticas y 4 horas de laboratorio). Como lenguaje de programación utiliza Java, pero desde el 2015 nos apoyamos con herramientas lúdicas para su enseñanza.

### B. Competencias

Nuestra currícula se realizó tomando en cuenta las competencias indicadas en ABET (Accreditation Board for Engineering and Technology) [9]. Donde se destaca la importancia de las habilidades profesionales y las habilidades de conciencia además del desarrollo de las habilidades técnicas para lograr excelencia en la formación de ingenieros.

En la Tabla IV y Tabla V se muestran las competencias generales y específicas del curso de Fundamentos de Programación 1.

TABLA IV. COMPETENCIAS GENERALES DEL CURSO FP1

- |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>c. Habilidad para diseñar un sistema, componente o proceso que satisfaga necesidades dentro de restricciones realistas tales como economía, medio ambiente, sociales, políticas, éticas, salud y de seguridad, manufacturación y sostenibilidad.</p> <p>k. La capacidad de utilizar las técnicas, habilidades y herramientas modernas de ingeniería y computación necesarias para la práctica de la ingeniería del software.</p> <p>m. Habilidad para aplicar apropiadamente matemáticas discretas, probabilidad y estadísticas, y tópicos</p> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

<p>relevantes en computación y disciplinas de apoyo a sistemas de software complejo</p>
-----------------------------------------------------------------------------------------

TABLA V. COMPETENCIAS ESPECÍFICAS DEL CURSO FP1

- |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> <li>1. Identifica, establece e integra los diferentes conceptos de programación reconociendo los componentes y características de un algoritmo.</li> <li>2. Elabora, crea y codifica algoritmos para la solución de problemas reales en un lenguaje de programación</li> <li>3. Aplica, codifica y ejecuta sentencias de selección y control apropiadas para la elaboración de programas.</li> <li>4. Introduce, analiza, y utiliza el concepto de funciones reconociendo su importancia en la programación</li> <li>5. Sintetiza, obtiene y elabora métodos que acepten arrays como parámetros reconociendo su uso.</li> <li>6. Analiza, conecta e integra el concepto de clases, agregación, composición, herencia y polimorfismo interpretando el paradigma orientado a objetos</li> <li>7. Concibe, analiza y utiliza las excepciones en una condición anormal de ejecución de un programa.</li> </ol> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### C. Contenidos

Los contenidos conceptuales del curso se resumen en la Tabla VI.

TABLA VI. CONTENIDOS CONCEPTUALES DEL CURSO FP1

Contenidos Conceptuales
<i>Unidad I. Conceptos Generales</i>
<p>Identifica los diferentes conceptos de programación reconociendo los componentes y características de un algoritmo.</p> <p>Elabora algoritmos para la solución de problemas reales.</p> <p><b>Introducción</b></p> <ul style="list-style-type: none"> <li>• Conceptos generales. Una aproximación con Lightbot y Alice</li> <li>• Algoritmo</li> <li>• Paradigmas de Programación</li> <li>• Lenguajes de Programación</li> <li>• Componentes de un Algoritmo</li> <li>• Característica de un algoritmo</li> <li>• Compiladores vs Intérpretes</li> <li>• IDEs</li> <li>• Pasos a seguir para crear programas</li> <li>• Pseudocódigo y diagramas de flujo</li> </ul> <p><b>Manipulación de Expresiones</b></p> <ul style="list-style-type: none"> <li>• Definición de Programa</li> <li>• Valores</li> <li>• Variables</li> <li>• Introducción a Tipos de Datos</li> <li>• Expresiones</li> <li>• Asignaciones</li> <li>• Operadores</li> </ul> <p><b>Iniciando con Java.</b></p> <ul style="list-style-type: none"> <li>• Entrada/Salida estándar</li> <li>• Primer programa en Java</li> <li>• Componentes de un Programa</li> <li>• Convenciones estándar para nombres</li> </ul>

<ul style="list-style-type: none"> <li>Plantilla de programas Java simples</li> </ul> <p><b>Creación y Declaración de Objetos</b></p> <ul style="list-style-type: none"> <li>Conceptos básicos de la Orientación a objetos</li> <li>Estado de la memoria de un programa</li> <li>Envío de Mensajes</li> <li>Utilizando Clases Standard <ul style="list-style-type: none"> <li>JOptionPane</li> <li>String</li> <li>Date</li> <li>SimpleDateFormat</li> </ul> </li> </ul> <p><b>Tipos de Datos Numéricos</b></p> <ul style="list-style-type: none"> <li>Datos numéricos.</li> <li>Expresiones aritméticas en Java.</li> <li>Evaluación de expresiones aritméticas utilizando reglas de precedencia.</li> <li>Casting implícito y explícito</li> <li>Constantes</li> <li>Asignación de memoria para los objetos y tipos de datos primitivos.</li> <li>Convertir valores String en datos numéricos.</li> <li>Standard Output/ Standard Input</li> <li>Expresiones matemáticas, utilizando la clase Math</li> </ul>
<b>Unidad II. Sentencias de Selección y Repetición</b>
<p>Aplica el uso de sentencias de selección y control apropiadas para la elaboración de programas. Introduce el concepto de funciones reconociendo su importancia en la programación</p> <p><b>Sentencias de Selección</b></p> <ul style="list-style-type: none"> <li>Sentencia if</li> <li>Expresiones Booleanas y variables</li> <li>Sentencia <b>if- anidado</b></li> <li>Sentencia <b>if- else if</b></li> <li>Comparación entre objetos</li> <li>Sentencia <b>switch</b></li> </ul> <p><b>Sentencias de Repetición</b></p> <ul style="list-style-type: none"> <li>Sentencia <b>while</b></li> <li>Errores comunes al escribir sentencias de repetición</li> <li>Sentencia <b>do-while</b></li> <li>Bucles y Control de repetición</li> <li>Dialogo de Confirmación</li> <li>Sentencia <b>for</b></li> <li>Sentencia <b>for-anidado</b></li> <li>Formato de salida</li> <li>Generación aleatoria de numeros</li> </ul>
<b>Unidad III. Métodos en Java</b>
<p>Introduce el concepto de métodos y/o funciones reconociendo su importancia en la programación</p> <p><b>Métodos y Construcción Modular de Programas</b></p> <ul style="list-style-type: none"> <li>Construcción Modular</li> <li>Declaración de Métodos Propios</li> <li>Invocación de Métodos</li> <li>Ventajas uso de métodos en la Ingeniería de Software</li> <li>Alcance de las variables en métodos</li> <li>Promoción de los argumentos</li> <li>Métodos de la clase Math</li> <li>Recursión</li> <li>Sobrecarga de métodos</li> </ul>

<b>Unidad IV. Arrays y ArrayList</b>
<p>Introduce el concepto de Arrays y ArrayList reconociendo su importancia en la programación</p> <p><b>Arrays y ArrayList</b></p> <ul style="list-style-type: none"> <li>Fundamentos de Arrays</li> <li>Declaración de Arrays</li> <li>Creación de Arrays</li> <li>Inicialización de Elementos</li> <li>Valores por defecto en Array</li> <li>Propiedad de largo del Array</li> <li>Arreglos parcialmente llenados</li> <li>Copiando un Array</li> <li>Histogramas</li> <li>Búsquedas en Arrays</li> <li>Ordenamiento en Arrays</li> <li>Ordenamiento por Selección</li> <li>Arrays Bi-Dimensionales</li> <li>Arrays de Objetos</li> </ul> <p><b>La Clase ArrayList</b></p> <ul style="list-style-type: none"> <li>Creación de objetos ArrayList</li> <li>Agregando elementos</li> <li>Accediendo a elementos en un ArrayList</li> <li>Actualizando un Objeto ArrayList</li> <li>Métodos adicionales ArrayList</li> <li>Impresión o Concatenación de un ArrayList</li> <li>Primitivas de Ordenamientos en un ArrayList</li> <li>Objetos ArrayList Objects Versus Arrays Standard</li> </ul>
<b>Unidad V. Programación Orientada a Objetos y Manejo de Excepciones</b>
<p>Analiza el concepto de clases, herencia y polimorfismo interpretando el paradigma orientado a objetos Concibe el concepto de excepciones como una condición anormal de ejecución de un programa</p> <p><b>POO, Herencia y Polimorfismo</b></p> <ul style="list-style-type: none"> <li>Jerarquía de clases en Java</li> <li>List, Linked List y ArrayList</li> </ul> <p><b>Manejo de Excepciones</b></p> <ul style="list-style-type: none"> <li>Utilizando Bloques Try-Catch para manejar llamada a métodos</li> <li>NumberFormatException</li> <li>Bloque Try-Catch - Mas detalles</li> <li>Dos tipos de Excepciones - Checked y Unchecked</li> <li>Excepciones Unchecked y Checked</li> <li>Utilizando Documentación API para el manejo de excepciones</li> <li>Cuando un Bloque try lanza diferente tipos de Excepciones</li> <li>La Clase Exception y el método getMessage</li> <li>Múltiples bloques catch</li> <li>Comprendiendo los Mensajes de Excepción</li> </ul>

#### D. Cursos de programación en la Currícula

En la Figura 1 se muestra la línea de programación en nuestra actual currícula, donde se muestra lo importante que es el curso Fundamentos de Programación 1 en toda la carrera. Se muestra la cantidad de créditos, cantidad de horas teóricas, prácticas y de laboratorio respectivamente.

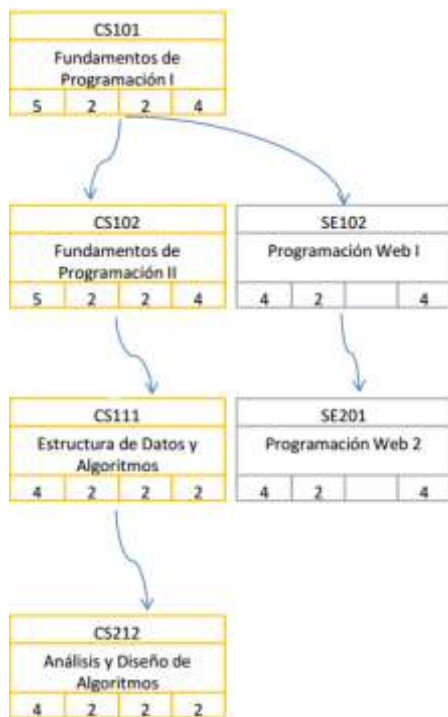


Fig. 1. Cursos de línea de Programación en la Currícula

#### E. Forma de Evaluación del curso

El curso se evalúa de la siguiente manera:

- Examen netamente práctico donde solucionan problemas aplicando los conocimientos teóricos enseñados. Rinden 2 exámenes a lo largo del curso.
- Evaluación Constante: constituida por prácticas en clase individuales o grupales, tareas para la casa, participación en la solución de problemas en clase, trabajos de investigación y lectura de artículos.
- Evaluación de laboratorio: prácticas, tareas y proyecto final.

El promedio final se calcula de la siguiente manera:

$$PF = (EC * 0.20 + IE * 0.30 + IIE * 0.50) * 0.70 + Laboratorio * 0.30$$

$PF =$  Promedio Final

$EC =$  Evaluación Constante

$IE =$  I Examen

$IIE =$  II Examen

### III. ESTILO DE APRENDIZAJE DE LOS NATIVOS DIGITALES

Los estudiantes de hoy en día han cambiado radicalmente, no son más las personas para las cuales el sistema educativo actual fue diseñado.

#### A. ¿Qué son los nativos digitales y cómo aprenden?

Prensky [10], creador del término “Nativo Digital”, observa que actualmente los estudiantes son todos “nativos hablantes” del lenguaje digital de las computadoras, videojuegos y del internet.

Se puede decir que el nativo digital construye sus conceptos en base a objetos digitales; absorben rápidamente la información de imágenes multimedia y videos; adquieren información en simultáneo de diversas fuentes, esperan respuestas instantáneas [10].

Su estilo de aprendizaje ha cambiado con el tiempo.

#### B. ¿Qué son los estilos de aprendizaje?

Los estilos de aprendizaje se definen como: “Rasgos cognitivos, afectivos, fisiológicos de preferencia por el uso de los sentidos, ambiente, cultura, psicología, comodidad, desarrollo y personalidad, que sirven como indicadores realmente estables de cómo las personas perciben, interrelacionan y responden a sus ambientes de aprendizaje y a sus propios métodos o estrategias en su forma de aprender” [11].

Es un conjunto de características cognitivas y pedagógicas que clasifican al estudiante según su forma de procesar y adquirir información [11]. De acuerdo a esa clasificación, se establece que los estudiantes pueden tener diferentes estilos de aprendizaje: visual, auditivo, kinestésico, etc.

En la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional de San Agustín de Arequipa - Perú, los profesores del curso de Fundamentos de Programación 1, primer curso de programación, se han interesado por descubrir el estilo de aprendizaje de los estudiantes denominados “nativos digitales” en nuestro entorno en particular.

El objetivo es aplicar métodos de enseñanza-aprendizaje de acuerdo a su proceso cognitivo y pedagógico. Se ha aplicado el Test desarrollado por Lynn O'Brien, a un universo de 92 estudiantes, ver Figura 2, confirmando el resultado de los estudios de [10], los cuales determinan que los nativos digitales han desarrollado una capacidad de aprendizaje muy visual.

La clasificación que plantea O'Brien corresponde a los siguientes estilos de aprendizaje: i) los visuales: los individuos en quienes predomina este sistema representacional captan el mundo con los ojos, se fijan mucho en los detalles visuales, recuerdan muy especialmente aquello que ven y hablan con predicados vinculados a este sentido, ii) los auditivos: experimentan el mundo a través del oído, se fijan mucho en los detalles auditivos, recuerdan lo que dice la gente y su lenguaje está muy influido por términos y expresiones vinculadas a la audición y iii) los kinestésicos: en esta tercera y última categoría se incluyen las personas en quienes predomina el tacto, el olfato y el gusto. Se trata de individuos que registran sus experiencias con el mundo exterior a través de alguno o varios de estos tres sentidos y, por supuesto, se expresan con predicados verbales acorde a ello.



Fig. 2. Estilos de Aprendizaje – Test-Lynn O'Brien, aplicado a 92 alumnos del curso de Fundamentos de Programación 1, en la Escuela Profesional de Ingeniería de Sistemas

Los profesores del curso han rediseñado el material de la primera sesión del curso de Fundamentos de Programación 1, para lograr captar la atención y romper barreras en los estudiantes de ésta generación.

Asimismo, el rediseño del material se basa en las sugerencias de [12], el cual determina técnicas para los estilos de aprendizaje y enseñanza en la educación de ingenieros. Se ha considerado los siguientes: i) sugiere que para utilizar un estilo de aprendizaje (deductivo/activo), se deben utilizar actividades que sean asistidas por el computador, ii) sugiere que se les dé a los estudiantes la oportunidad de trabajar de forma colaborativa, ese aprendizaje activo mejora cuando interactúan con sus homólogos, iii) motivar el aprendizaje, utilizando material que mejore la experiencia personal de los estudiantes, proceso inductivo/global.

#### IV. LIGHTBOT Y ALICE

##### A. Lightbot

Es un videojuego tipo rompecabezas (puzzle game). Sus propiedades nos llevan directamente a establecer relaciones con conceptos de programación. La meta del juego es guiar al robot a encender los cuadros azules. El jugador le debe indicar cómo lograrlo, usando un conjunto de instrucciones mostradas en la Figura 3. Estas instrucciones son `forward()`, `light()`, `turnLeft()`, `turnRight()`, `jump()`, `proc1()` y `proc2()`.



Fig. 3. Instrucciones usadas en Lightbot

Al comienzo sólo aparecen las dos primeras instrucciones y el resto van apareciendo a medida que vamos alcanzando un nuevo nivel y se constituyen en herramientas para la solución de problemas más complejos.

##### B. Alice

Es un entorno de programación diseñado específicamente como una herramienta de enseñanza-aprendizaje que permite a los programadores novatos poder crear animaciones y juegos con mundos 3D [13].

Fue desarrollado por la Universidad de Carnegie Mellon en EE.UU. como apoyo para la enseñanza de la Programación Orientada a Objetos.

Con este programa resulta más fácil comprender la programación de la orientación a objetos, por su naturaleza visual y el atractivo de las animaciones en 3D.

Alice va de la mano con Java, ya que utiliza parte de este lenguaje para las funciones de animación aplicables a objetos, tales como mover, saltar, girar, etc.

#### V. PROPUESTA DE LA PRIMERA SESIÓN DE CS1

La primera sesión se convierte en un hito temprano para el resto del curso. Después de la presentación del docente y del curso en general se realiza la encuesta cuyos resultados se muestran en las tablas I, II y III. Posteriormente se realizan las siguientes actividades:

##### A. Introducción al videojuego Lightbot

Se les explica que aprenderán muchos de los conceptos básicos de la programación “jugando”. Inicialmente muestran sorpresa por dicha afirmación, aunque queda debidamente justificada por los resultados de la encuesta, y se despierta expectativa, capturando así su atención.

En la Figura 4 se les muestra el entorno del juego, la forma de interactuar con él, indicando el objetivo y las instrucciones que pueden utilizar para alcanzarlo.



Fig. 4. Entorno de Lightbot y primer problema a resolver

La motivación es inmediata, despertando su interés por solucionar el problema. Se estimulan de forma visual, auditiva y por la interacción. Figura 5.



Fig. 5. Solución al primer problema de Lightbot



En los siguientes niveles se les muestra que la complejidad del problema puede aumentar, pero que se les brinda nuevas instrucciones para solucionarlo. Ver Figura 6.



Fig. 6. Niveles más complejos de problemas a resolver

Desde un primer momento se les hace hincapié sobre algunos conceptos básicos de la programación, tales como instrucción, abstracción, reúso, complejidad, algoritmo e incluso programa.

La complejidad continúa aumentando de problema en problema y la motivación aumenta paralelamente. Retroalimentamos preguntando sobre los conceptos que van aprendiendo y comprobando que son comprendidos por casi todos los estudiantes. Esta experiencia después es referenciada a lo largo del curso, cuando se les enseña algún concepto de manera formal se les hace la referencia a cierta situación de esta sesión de aprendizaje. Todo lo que se aprende en esta sesión se constituye posteriormente en un “saber previo” para el resto del curso.

Se plantea un conflicto cognitivo con el siguiente ejercicio, ver Figura 7. Se les da una restricción de 12 posibles instrucciones que tiene el programa principal o main, concepto

que también será importante a la hora de programar en cualquier lenguaje de programación, pero se les hace notar que aparece un segundo espacio para ubicar las instrucciones, el área PROC1. Ahí se les enseña a crear su propio procedimiento, llamado también función o método, mostrando el mecanismo de invocación del programa principal a dicho procedimiento.



Fig. 7. Introducción del concepto de procedimiento o método

Se continúa con otros problemas, introduciendo el concepto de eficiencia y mostrando que pueden haber varias soluciones para un mismo problema, pero unas soluciones más eficientes que otras.

Los estudiantes en este punto están totalmente inmersos en la sesión y se muestra su interés por seguir solucionando problemas. Se les hace hincapié en la metáfora que la programación va a ser un proceso muy similar a lo que están realizando con el juego y que la experticia que se logra en el juego es por medio de la práctica, de la misma manera se logrará la experticia para programar. Queda como actividad completar el juego en otros niveles, incluso de loops, que si bien no era el objetivo, ellos ya se encuentran con la motivación y el conocimiento para resolver los problemas por sí mismos.

Después de retroalimentar la experiencia, ingresamos a la segunda parte de la sesión.

*B. Introducción al entorno de animación 3D Alice*

Aquí se debe aprovechar el momentum obtenido previamente y se introduce Alice, explicándoles que ahora se verá otro entorno, la forma de interactuar con él y la forma básica de crear animaciones en 3D. Ver Figura 8.



Fig. 8. Entorno de la herramienta de animaciones 3D Alice

Nuestro curso, al tener como lenguaje de programación a Java, necesita ciertos conceptos de orientación a objetos que de forma convencional siempre resultaron algo complejos de enseñar, especialmente al inicio del curso.

Con Alice conceptos como clase, objeto, atributo, método y mensaje quedan bastante claros. Ver Figura 9.



Fig. 9. Ejemplos de animaciones 3D en Alice

Debemos aclarar que si bien podríamos entrar ya a la programación por medio de Alice, no es nuestro objetivo principal. Lo que deseamos es que se entiendan ciertos

conceptos y se sientan motivados para la enseñanza de un lenguaje de programación orientado a objetos como Java.

Finalmente se les motiva a que creen animaciones en 3D usando su imaginación y siendo totalmente libres para realizar la actividad.

## VI. ANÁLISIS DE RESULTADOS

La retroalimentación que recibimos por parte de los estudiantes es muy positiva ya que realmente disfrutaron las dos herramientas lúdicas que se les presentó. Incluso para la segunda sesión, al empezar se les pregunta sobre la actividad que se les dejó a desarrollar y la gran mayoría indicó que avanzaron en el videojuego de Lightbot, además unos cuantos hasta completaron todos los niveles. Sobre la actividad de Alice, todos sin excepción crearon algún mundo en 3D, unos más elaborados que otros, pero todos cumplieron.

Es destacado el hecho que al preguntar al azar sobre alguno de los conceptos de programación vistos en la primera sesión, todos respondían correctamente.

Hay que considerar que en el curso siempre tuvimos un gran problema en el rendimiento del primer examen, el cual se realiza al acabar la segunda unidad, en él las notas siempre fueron bajas los dos años anteriores. Después de esta experiencia la cantidad de aprobados, promedio general, incluso las notas máxima y mínima aumentaron (calificación vigesimal). Tabla VII.

TABLA VII. RESULTADOS EVALUACIÓN DEL PRIMER EXAMEN

	2013	2014	2015	2016
Cantidad estudiantes	115	120	111	90
Aprobados	41.84%	44.12%	52.81%	55.02%
Desaprobados	58.16%	55.88%	47.19%	44.98%
Promedio	8.50	9.25	12.05	12.95
Desviación Estándar	4.12	4.28	3.78	3.52
Máxima	16	17	19.5	19
Mínima	1	2	4.5	7

## CONCLUSIONES

En este artículo se ha compartido la experiencia de la primera sesión del curso CS1, denominado en nuestra currícula como Fundamentos de Programación 1.

Se ha presentado las herramientas lúdicas que utilizamos para estimular la motivación de los estudiantes y enseñar de manera práctica y efectiva conceptos fundamentales de la programación, tales como: instrucción, abstracción, reúso, complejidad, algoritmo, programa, procedimiento o método, eficiencia, clase, objeto, atributo, métodos y mensaje.

Se ha demostrado que una aproximación a entornos lúdicos es efectiva para una generación de estudiantes que son denominados nativos digitales, que entre otras características presentan desinterés por lo que no les motiva visualmente ni les llama la atención.

La relación con estas herramientas no acaba ahí, siempre constituyen parte de los ejemplos al enseñar los conceptos de forma convencional a lo largo del curso, suavizando de esta forma su posterior aprendizaje.



## REFERENCIAS

- [1] E. Sweedyk, M. deLaet, M. C. Slattery, and J. Kuffner, "Computer games and CS education: why and how," in Proceedings of the 36<sup>th</sup> SIGCSE technical symposium on Computer science education , St. Louis, Missouri, USA , 2005, pp. 256-257.
- [2] U. Wolz, T. Barnes, I. Parberry, and M. Wick, "Digital gaming as a vehicle for learning," ACM SIGCSE Bulletin, vol. 38, no. 1, pp. 394-395, 2006.
- [3] J. D. Bayliss and S. Strout, "Games as a "flavor" of CS1," ACM SIGCSE Bulletin, vol. 38, no. 1, pp. 500-504, Mar. 2006.
- [4] T. Lorenzen, W. Heilman , "CS1 and CS2: Write Computer Games in Java!" SIGCSE Bull., 34(4), 99-100, 2002.
- [5] B. W. Becker. "Teaching CS1 with karel the robot in Java." ACM SIGCSE Bulletin. Vol. 33. No. 1. ACM, 2001.
- [6] Escuela Profesional de Ingeniería de Sistemas. <http://www.episunsa.edu.pe>
- [7] Lightbot. <https://lightbot.com/hocflash.html>
- [8] Alice. <http://www.alice.org/>
- [9] ABET. Criteria for Accrediting Engineering Programs, 2015 – 2016. <http://www.abet.org/wp-content/uploads/2015/05/E001-15-16-EAC-Criteria-03-10-15.pdf#outcomes>. Último acceso Enero 2017.
- [10] M. Prensky, "Digital Natives, Digital Immigrants," Horiz., vol. 9, no. 5, pp. 1–6, 2001.
- [11] J. García Cué, "Estilos de Aprendizaje y Tecnologías de la Información y la Comunicación en la Formación de Profesorado". Tesis Doctoral. España-UNED, 2006
- [12] R. Felder, L. Silverman, "Learning and teaching styles in engineering education," Eng. Educ., vol. 78, no. June, pp. 674–681, 1988.
- [13] W. Dann, "Mediated Transfer: Alice 3 to Java", 2012