

# Experience in the Use of a Virtual Classroom for Teaching Computer Programming: Development of Algorithmic Thinking and Learning a Programming Language

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

*Abstract– This article describes the use of a virtual classroom to provide students in the first semester of the Professional School of Ingeniería de Sistemas of the Universidad Nacional de San Agustín de Arequipa - Peru an organized, motivating, communicative and innovative environment for the basic course of Computer programming: Fundamentals of Programming 1. A problem that exists in many Latin American countries is that educators who teach the basic computer programming course find that for most students, computer programming is a completely new subject. As a result, it has been shown that the use of a virtual classroom as a support tool to carry out the teaching-learning process of computer programming concepts gives students a more effective and motivating experience. Here we will describe the experience using the Virtual Classroom implemented by the University Office of Information and Communication Technologies to teach the basic concepts of computer programming.*

*Keywords– Programming Fundamentals, Learn to Program, Motivation in Teaching, Virtual classroom, Pedagogical Tools.*

Digital Object Identifier (DOI):  
<http://dx.doi.org/10.18687/LACCEI2020.1.1.37>  
ISBN: 978-958-52071-4-1 ISSN: 2414-6390

# Experiencia en la Utilización de un Aula Virtual para la Enseñanza de la Programación de Computadoras: Desarrollo del Pensamiento Algorítmico y Aprendizaje de un Lenguaje de Programación

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

**Abstract**— Este artículo describe la utilización de un aula virtual para proporcionar a los estudiantes del primer semestre de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional de San Agustín de Arequipa - Perú un entorno organizado, motivador, comunicativo e innovador para el curso básico de programación de computadoras: Fundamentos de Programación 1. Un problema que existe en muchos países latinoamericanos es que los educadores que enseñan el curso básico de programación de computadoras encuentran que para la mayoría de los estudiantes, la programación de computadoras es una materia completamente nueva. Como resultado, se ha demostrado que la utilización de un aula virtual como herramienta de apoyo para llevar a cabo el proceso de enseñanza-aprendizaje de los conceptos de programación de computadoras brinda a los estudiantes una experiencia más efectiva y motivadora. Aquí describiremos la experiencia usando el Aula Virtual implementada por la Dirección Universitaria de Tecnologías de Información y Comunicación para enseñar los conceptos básicos de la programación de computadoras.

**Keywords**— Fundamentos de Programación; Aprender a Programar; Motivación en la Enseñanza; Aula Virtual; 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]. En base a dichas experiencias, a partir del 2015, se introdujeron estrategias de enseñanza de conceptos básicos de la programación de computadoras [5],[6],[7], estrategias que dieron buenos resultados y motivaron a los estudiantes.

Si bien el rendimiento académico de los estudiantes mostró una mejora importante, se decidió para el 2019 integrar en un curso desarrollado sobre un Aula Virtual todas las estrategias aplicadas anteriormente y además sumar nuevas estrategias.

El curso de introducción a la programación de computadoras, que en la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional de San Agustín de Arequipa – Perú [8] se denomina Fundamentos de Programación 1 (FP1), se empezó a enseñar como parte del Plan de Estudios 2013 y continúa conformando el nuevo Plan de Estudios 2017, busca enseñar los conceptos básicos de la

programación de computadoras desarrollando el pensamiento algorítmico y está centrado en el aprendizaje del lenguaje de programación Java [9].

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 7 años en los que se viene dictando el curso, a la pregunta realizada el primer día de clases de “¿cuántos de ustedes tienen alguna experiencia en programación de computadoras?”

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 la mayoría de los estudiantes del curso.

TABLA I  
RESPUESTAS A CUESTIONAMIENTO INICIAL

Año	Si	No
2013	5%	95%
2014	6%	94%
2015	5%	95%
2016	7%	93%
2017	5%	95%
2018	4%	96%
2019	7%	93%

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 números)
2013	40%	55%	3%	2%	0%
2014	45%	49%	4%	2%	0%
2015	35%	60%	2%	2%	1%
2016	40%	53%	3%	3%	1%
2017	39%	57%	3%	1%	0%
2018	40%	55%	3%	1%	1%
2019	39%	56%	4%	1%	0%

Digital Object Identifier (DOI):  
<http://dx.doi.org/10.18687/LACCEI2020.1.1.37>  
 ISBN: 978-958-52071-4-1 ISSN: 2414-6390

La heterogeneidad mostrada en el grupo de estudiantes es tratada de homogeneizar basada en los estilos de aprendizaje de los post-millennials o Generación Z, y usando herramientas que ellos conocen muy bien y utilizan cotidianamente: herramientas de software tales como videojuegos, entornos de animación en 3D, chat, redes sociales y otros entornos virtuales.

En este artículo presentamos nuestra experiencia utilizando como plataforma de comunicación y de enseñanza-aprendizaje al Aula Virtual implementada por la Dirección Universitaria de Tecnologías de Información y Comunicación que está basada en Moodle [10].

El resto del artículo está organizado de la siguiente manera. En la sección II se describe el primer curso de programación de nuestro plan de estudios. En la sección III se muestran los estilos de aprendizaje de los estudiantes en nuestro entorno en particular. En la sección IV se muestra la experiencia del curso utilizando un Aula Virtual, basándonos en el estilo de aprendizaje de nuestros estudiantes y usando diferentes herramientas, recursos y actividades. En la sección V se muestran los resultados obtenidos del rendimiento de los estudiantes, donde se compara el rendimiento cuando utilizamos un enfoque tradicional de enseñanza, comparándolo con un enfoque lúdico de enseñanza y contrastándolo con un enfoque lúdico e integrado en un Aula Virtual. Finalmente se presentan nuestras conclusiones.

## II. EL PRIMER CURSO DE PROGRAMACIÓN

### A. Generalidades

El primer curso de programación de computadoras en los Planes de Estudio 2013 y 2017 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 y desde el 2019 se utiliza un Aula Virtual como apoyo al proceso enseñanza-aprendizaje.

### B. Competencias

Ambos planes de estudio se elaboraron tomando en cuenta los resultados del estudiante indicados por Accreditation Board for Engineering and Technology (ABET) [11]. 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 III y Tabla IV se muestran las competencias generales y específicas del curso de Fundamentos de Programación 1.

### C. Contenidos Conceptuales

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

TABLA III  
COMPETENCIAS GENERALES DEL CURSO FP1

<p>C.c. Diseña responsablemente sistemas, componentes o procesos para satisfacer necesidades dentro de restricciones realistas: económicas, medio ambientales, sociales, políticas, éticas, de salud, de seguridad, manufacturación y sostenibilidad.</p> <p>C.p. Aplica de forma flexible técnicas, métodos, principios, normas, estándares y herramientas de ingeniería necesarias para la construcción de software e implementación de sistemas de información</p>
---

TABLA IV  
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 de repetición apropiadas para la elaboración de programas.</li> <li>4. Introduce, analiza y utiliza el concepto de métodos reconociendo su importancia en la programación.</li> <li>5. Introduce, analiza y aplica los conceptos de arreglo estándar y ArrayList reconociendo su importancia en la solución de ciertos problemas.</li> <li>6. Analiza, conecta e integra los conceptos de clase, objeto, atributo y método reconociendo su importancia en la programación</li> <li>7. Concibe, analiza y utiliza las excepciones como una condición anormal de ejecución de un programa.</li> </ol>
--

TABLA V  
CONTENIDOS CONCEPTUALES DEL CURSO FP1

Capítulo 1. Generalidades
<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>Introducción</p> <ul style="list-style-type: none"> <li>• Conceptos generales. Una aproximación con Lightbot, Lightbot 2.0 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>Manipulación de Expresiones</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, asignaciones y operadores</li> </ul> <p>Iniciando con Java.</p> <ul style="list-style-type: none"> <li>• Entrada/Salida estándar</li> <li>• Primer programa en Java</li> </ul>

<ul style="list-style-type: none"> <li>• Componentes de un programa</li> <li>• Convenciones estándar para nombres</li> <li>• Plantilla de programas Java simples</li> </ul> <p>Declaración y Creación de Objetos de Clases Estándar</p> <ul style="list-style-type: none"> <li>• Conceptos básicos de la orientación a objetos: clase, objeto, atributo y método</li> <li>• Estado de la memoria de un programa</li> <li>• Envío de mensajes</li> <li>• Utilizando clases estándar de Java <ul style="list-style-type: none"> <li>○ JOptionPane</li> <li>○ String</li> <li>○ Date</li> <li>○ SimpleDateFormat</li> </ul> </li> </ul> <p>Tipos de Datos Numéricos</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>• Salida estándar/ Entrada estándar</li> <li>• Expresiones matemáticas, utilizando la clase Math</li> </ul>
<p>Capítulo 2. Sentencias de Selección y Repetición</p>
<p>Aplica las sentencias de selección y de repetición apropiadas para la elaboración de programas.</p> <p>Sentencias de Selección</p> <ul style="list-style-type: none"> <li>• Sentencia if</li> <li>• Expresiones Booleanas y variables</li> <li>• Sentencia if- anidado</li> <li>• Sentencia if- else if</li> <li>• Comparación entre objetos</li> <li>• Sentencia switch</li> </ul> <p>Sentencias de Repetición</p> <ul style="list-style-type: none"> <li>• Sentencia while</li> <li>• Errores comunes al escribir sentencias de repetición</li> <li>• Sentencia do-while</li> <li>• Bucles y Control de repetición</li> <li>• Dialogo de confirmación</li> <li>• Sentencia for</li> <li>• Sentencia for-anidado</li> <li>• Bucles anidados</li> <li>• Generación aleatoria de números</li> </ul> <p>Sentencias Compuestas</p> <ul style="list-style-type: none"> <li>• Usar conjuntamente sentencias de selección y de repetición</li> <li>• Recomendaciones y errores comunes</li> </ul>
<p>Capítulo 3. Métodos</p>
<p>Introduce el concepto de métodos y/o funciones reconociendo su importancia en la programación.</p> <p>Métodos y Construcción Modular de Programas</p> <ul style="list-style-type: none"> <li>• Construcción modular</li> <li>• Declaración de métodos propios</li> <li>• Invocación de métodos</li> <li>• Ventajas del uso de métodos en la Ingeniería de Software</li> <li>• Alcance de las variables en métodos</li> </ul>

<ul style="list-style-type: none"> <li>• Promoción de los argumentos</li> <li>• Métodos de la clase Math</li> <li>• Recursión</li> <li>• Sobrecarga de métodos</li> </ul>
<p>Capítulo 4. Arreglos Estándar y ArrayList</p>
<p>Introduce el concepto de arreglo estándar (arrays) y ArrayList reconociendo su importancia en la programación.</p> <p>Arreglos estándar y ArrayList</p> <ul style="list-style-type: none"> <li>• Fundamentos de arreglos</li> <li>• Declaración de arreglos</li> <li>• Creación de arreglos</li> <li>• Inicialización de elementos</li> <li>• Valores por defecto en arreglos</li> <li>• Propiedad de longitud del arreglo</li> <li>• Arreglos parcialmente llenados</li> <li>• Copiando un arreglo</li> <li>• Histogramas</li> <li>• Búsquedas en arreglos</li> <li>• Ordenamiento en arreglos</li> <li>• Arreglos Bidimensionales</li> <li>• Arreglos de Objetos</li> </ul> <p>La Clase ArrayList</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 del ArrayList</li> <li>• Impresión o concatenación de un ArrayList</li> <li>• Primitivas de ordenamientos en un ArrayList</li> <li>• Objetos ArrayList versus arreglos estándar</li> </ul>
<p>Capítulo 5. Programación Orientada a Objetos y Manejo de Excepciones</p>
<p>Profundiza los fundamentos de la programación orientada a objetos y la creación de clases propias, reconociendo su importancia en la programación.</p> <p>Analiza el concepto de herencia y polimorfismo interpretando el paradigma orientado a objetos.</p> <p>Concibe el concepto de excepciones como una condición anormal de ejecución de un programa.</p> <p>Fundamentos de la Programación Orientada a Objetos</p> <ul style="list-style-type: none"> <li>• Clase, objeto, atributo, método y mensaje</li> <li>• Creación de clases propias</li> <li>• Métodos constructores, accesores y mutadores</li> <li>• Métodos booleanos</li> <li>• Método toString</li> </ul> <p>Introducción a la Herencia y Polimorfismo</p> <ul style="list-style-type: none"> <li>• Jerarquía de clases en Java</li> <li>• List, LinkedList y ArrayList</li> </ul> <p>Manejo de Excepciones</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 - más detalles</li> <li>• Utilizando documentación API para el manejo de excepciones</li> <li>• Cuando un bloque try-catch lanza diferentes tipos de excepciones</li> <li>• La clase Exception y el método getMessage</li> <li>• Múltiples bloques catch</li> <li>• Comprendiendo los mensajes de una excepción</li> </ul>

#### D. Cursos de programación en los Planes de Estudio 2013 y 2017

En la Fig. 1 se muestra la línea de programación en ambos planes de estudio, donde se muestra lo importante que es el curso Fundamentos de Programación 1 en toda la carrera profesional. Observar que el curso es prerrequisito de Fundamentos de Programación 2 (FP2) y de Programación Web 1 (PW1), los que a su vez abren ramas importantes de cursos fundamentales del plan de estudios.

Se muestra la cantidad de créditos, cantidad de horas teóricas, prácticas y de laboratorio respectivamente.

#### E. Forma de Evaluación del Curso

La evaluación del curso tiene dos componentes:

1) *Examen*: netamente práctico donde los estudiantes solucionan problemas aplicando los conocimientos teóricos enseñados. Rinden 3 exámenes durante el desarrollo del curso.

2) *Evaluación Continua*: constituida principalmente por la evaluación de laboratorio, prácticas en clase individuales y grupales, tareas para la casa, participación en la solución de problemas en clase, trabajos de investigación, pequeños proyectos de desarrollo de software y lectura de artículos. Son 3 calificaciones a lo largo del curso.

El promedio final se calcula de la siguiente manera:

$$PF = EC1 * 0.05 + EX1 * 0.05 + EC2 * 0.05 + EX2 * 0.2 + EC3 * 0.3 + EX3 * 0.35$$

PF = Promedio Final

EC = Evaluación Continua

EX = Examen

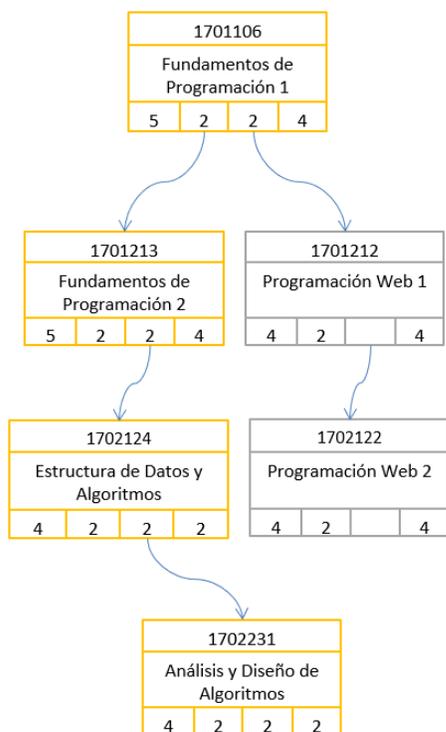


Fig. 1 Cursos de la línea de Programación en los Planes de Estudio 2013 y 2017

#### III. ESTILO DE APRENDIZAJE DE LOS ESTUDIANTES EN NUESTRA REALIDAD

Actualmente, los estudiantes que cursan estudios universitarios son diferentes a los que lo hacían hace una década o más, la pregunta es: “¿está nuestro sistema educativo actual diseñado para este tipo de estudiantes?”

Algunos los llaman “Post-Millennials” y otros incluso “Generación Z”, individuos que nacieron junto al internet (desde fines de los 90s hasta la primera década de los 2000s) y cuya filosofía de vida y forma de aprender difieren de generaciones anteriores, viéndose ampliamente influenciados por la tecnología.

Un error común que cometen los docentes es no considerar esta situación y continuar pensando que los estudiantes actuales son iguales a los estudiantes que tenían hace años, y aún peor, no adaptarse a ello y seguir aplicando lo que en otras generaciones funcionaba pero que, con la actual generación, en muchos casos, parece no hacerlo.

Así, Prensky [12], observa que los estudiantes actuales son todos “nativos hablantes” del lenguaje digital de las computadoras, videojuegos y del internet.

Se puede decir que dicha generación construye sus conceptos en base a objetos digitales; absorben rápidamente la información de imágenes multimedia y videos; adquieren información en simultáneo de diversas fuentes, incluidas las redes sociales y esperan respuestas instantáneas.

Los estilos de aprendizaje son un conjunto de características cognitivas y pedagógicas que clasifican al estudiante según su forma de procesar y adquirir información [13]. 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 docentes del curso de Fundamentos de Programación 1, primer curso de programación, se interesan anualmente en descubrir el estilo de aprendizaje de los actuales estudiantes.

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 122 estudiantes en el 2019, ver Fig. 2, confirmando el resultado de los estudios de [12], los cuales determinan que nuestros actuales estudiantes de primer año han desarrollado una capacidad de aprendizaje primordialmente visual y kinestésico (situación que se acentúa año tras año).

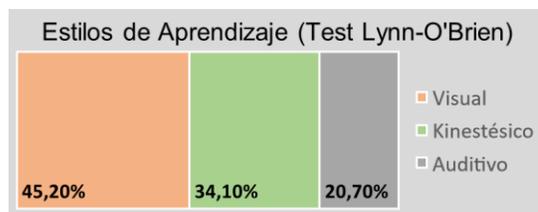


Fig. 2 Estilos de Aprendizaje – Test-Lynn O'Brien, aplicado a 122 estudiantes del curso de Fundamentos de Programación 1 en el año 2019

Los docentes del curso han utilizado este conocimiento para el rediseño del material del curso y así lograr captar la atención, romper barreras en los estudiantes de esta generación y enseñar los conceptos fundamentales de programación de computadoras de forma efectiva.

Asimismo, el rediseño del material del curso se basa en las sugerencias de [14], el cual recomienda técnicas para los estilos de aprendizaje y enseñanza en la educación de ingenieros. Se han considerado las siguientes sugerencias:

1) *Estilo Deductivo/Activo*: que para utilizar este estilo se deben utilizar actividades que sean asistidas por el computador.

2) *Aprendizaje Colaborativo*: que se les dé a los estudiantes la oportunidad de trabajar de forma colaborativa, ese aprendizaje activo mejora cuando interactúan con sus pares.

3) *Proceso Inductivo/Global*: que se debe motivar el aprendizaje, utilizando material que mejore la experiencia personal de los estudiantes.

#### IV. LA EXPERIENCIA

Consideramos que la motivación es un pilar fundamental en el aprendizaje de la programación de computadoras y que mejor que incorporar el video de “Todo el Mundo debería saber programar”, donde grandes personalidades mundiales como Mark Zuckerberg, el creador de Facebook, exponen la importancia de la programación de computadoras para todos los ciudadanos de la nueva sociedad de la información y el conocimiento. En la Fig. 3 se muestra el video referenciado en la página principal del curso en el Aula Virtual, junto a la descripción y competencias del mismo.



Fig. 3 Video motivador en la página principal del Aula Virtual del curso

Tratando de buscar un mejor orden de los contenidos del curso es que se colocaron al inicio los recursos “URL” con los enlaces a los entornos lúdicos utilizados, videos de otros cursos internacionales relacionados y a un entorno de programación en vivo. Además, se colocaron las diapositivas de los temas a desarrollar en clase divididos por capítulos según el sílabo correspondiente, para esto se utilizó el recurso “Archivo” del Aula Virtual. En la Fig. 4 se muestra el Aula Virtual creada con los URL, sílabo y contenidos teóricos.

En la Fig. 5 se muestran cómo las experiencias prácticas de laboratorio también son colocadas en el Aula Virtual en forma de actividades de tipo “Tarea” con un plazo de entrega adecuado y con retroalimentación constante. Hay que considerar que varios de los problemas a solucionar en dichas prácticas de laboratorio tienen un componente lúdico, tal como la utilización y la implementación de videojuegos.

Entre los videojuegos utilizados para enriquecer el proceso enseñanza-aprendizaje cabe destacar LightBot [15] y LightBot 2.0 [16] que son videojuegos tipo rompecabezas (puzzle game) que permiten desarrollar el pensamiento algorítmico de forma lúdica y muy motivadora para los estudiantes. Estos videojuegos se utilizan principalmente en las primeras sesiones del curso para realizar una introducción lúdica y amigable a los conceptos básicos de programación.

Alice 3 [17] 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 y permite entender de manera muy amigable los conceptos de la programación orientada a objetos.

La aplicación educativa de estos entornos lúdicos es ampliamente descrita en [5],[6]. Ver Fig. 6 y Fig. 7.

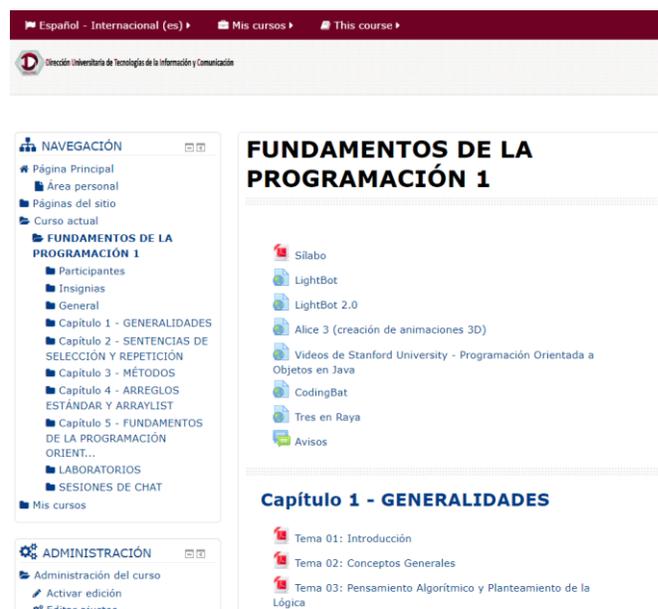


Fig. 4 Sílabo, URLs y contenidos teóricos divididos por capítulos en el Aula Virtual

1. LightBot
2. LightBot 2
3. Alice 3
4. Instalación de Eclipse y Primer Programa en Java
5. Entrada y Salida Estándar
6. Creación de Objetos Básicos
7. Expresiones Aritméticas
8. Sentencias Condicionales Simples
9. Sentencias Condicionales Avanzadas
10. Sentencias Repetitivas Simples
11. Sentencias Repetitivas Avanzadas
12. Sentencias Compuestas
13. Métodos - CodingBat
14. Métodos - Mente Maestra La Entrevista de Google
15. Arreglos Unidimensionales
16. Arreglos Unidimensionales - Juego del Ahorcado
17. Arreglos Bidimensionales o Matrices
18. Aplicación de Arreglos Bidimensionales - Tres en Raya
19. ArrayList
20. Programación OO - Creación de Clases Propias
21. Programación OO - Métodos y Constructores
22. Manejo de Excepciones
23. Proyecto - Tres en Raya v.1.0.
24. Proyecto - Tres en Raya v.2.0.
25. Proyecto - Tres en Raya v.3.0.

Fig. 5 Tareas prácticas de laboratorio del Aula Virtual del curso

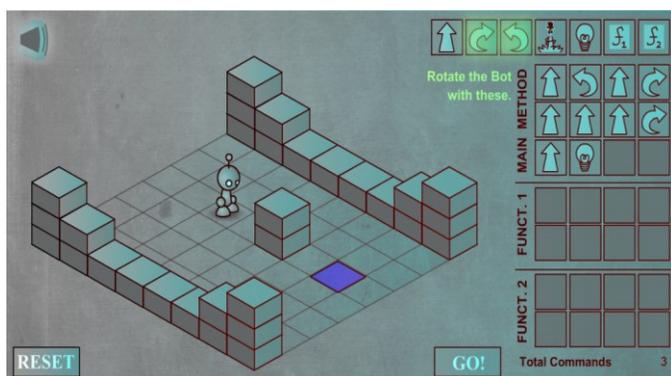


Fig. 6 Entorno del videojuego Lightbot

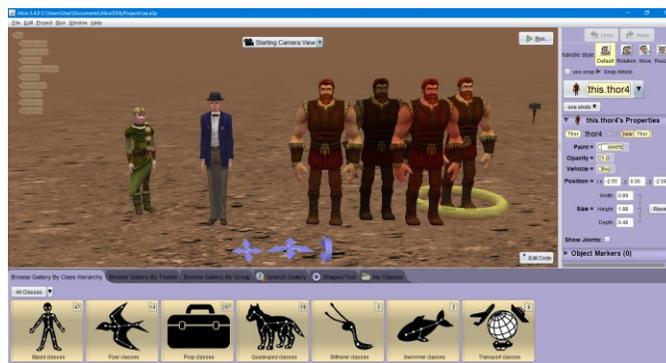


Fig. 7 Entorno de la herramienta de animaciones 3D Alice 3

También se utilizó la actividad “Juego Ahorcado” que brinda el Aula Virtual, complementado con la actividad “Glosario” compuesto por términos relacionados a la computación. Dicho juego se utilizó como fundamento y motivación para la implementación personalizada del juego por parte de cada estudiante en las prácticas de laboratorio del Tema 15 sobre arreglos estándar. En la Fig. 8 y la Fig. 9 se puede observar su utilización.

Como proyecto de curso se realizó la implementación de un videojuego de tres en raya, en donde se realizaron varias iteraciones para completar el juego. Cada iteración era incremental, permitiendo al estudiante aplicar cada concepto desarrollado en el curso. El Aula Virtual permitió la presentación de los entregables de este proyecto como actividades de tipo “Tarea”. Ver Fig. 10.

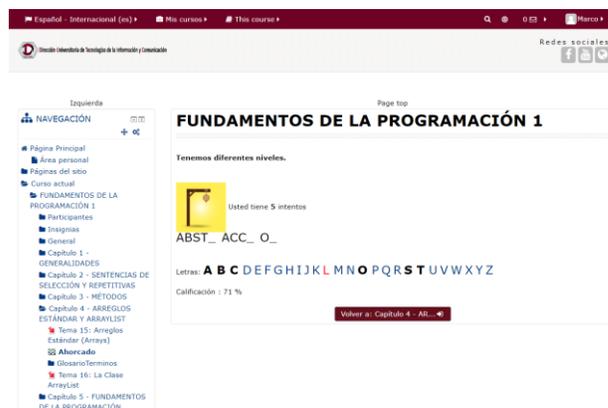


Fig. 8 Actividad – Juego Ahorcado del Aula Virtual del curso



Fig. 9 Juego del Ahorcado y su Glosario en el Aula Virtual del curso

Además, se creó una actividad “Chat” con un horario semanal para absolución de dudas de los temas desarrollados en clase y de las tareas propuestas. Ver Fig. 10.

En la Fig. 11 se muestra un recurso “URL” muy importante, que es la incorporación de cursos internacionales completos relacionados a Fundamentos de Programación 1. Cursos de alta calidad de universidades prestigiosas como Stanford University [18] o del Massachusetts Institute of Technology (MIT) [19]. Recurso que motivó bastante a los estudiantes, ya que valoraron que los contenidos que ellos desarrollaron en el curso son muy similares a los que universidades de clase mundial imparten a sus estudiantes.

En la Fig. 12 se muestra el entorno web CodingBat [20], que es un sitio web de libre acceso, que permite solucionar problemas de programación en vivo y desarrollar las habilidades de programación en Java y en otros lenguajes de programación. Es una excelente herramienta de aprendizaje de programación de computadoras y se implementó usando el recurso “URL” del Aula Virtual.

En la Fig. 13 se puede ver la forma de utilización de este recurso, primero se presenta el problema y el estudiante debe dar una solución, ejecutar el programa realizado y el CodingBat corrige utilizando varios casos de prueba y retroalimentando al estudiante como problema correctamente resuelto o no (situación en que algún caso de prueba falle).



Fig. 10 Proyecto videojuego Tres en Raya (3 iteraciones) y sesiones de chat del Aula Virtual del curso

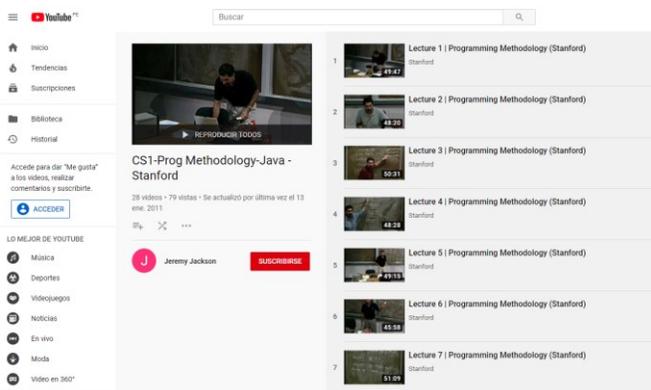


Fig. 11 Enlace a videos de un curso completo de programación en universidad de clase mundial en el Aula Virtual del curso

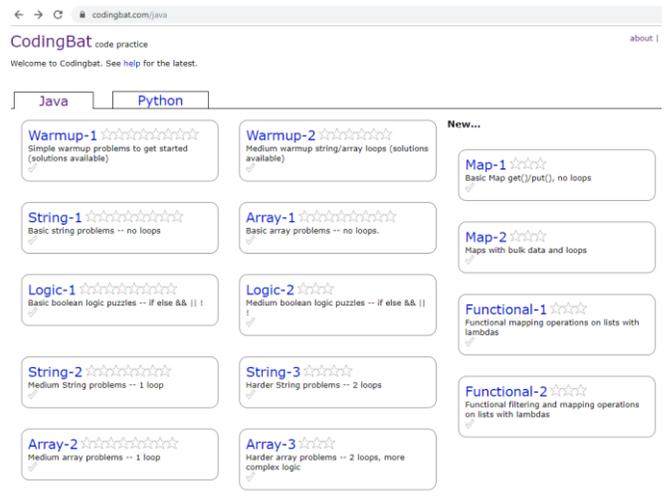


Fig. 12 Entorno del sitio web de programación en vivo CodingBat

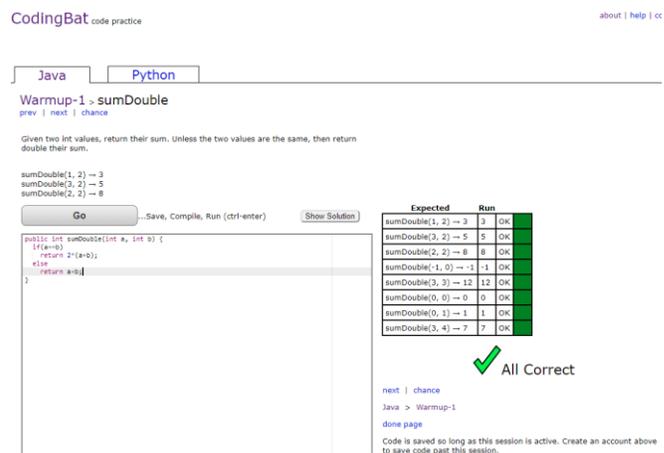


Fig. 13 Ejecución correcta de una solución a un problema planteado en CodingBat

## V. ANÁLISIS DE RESULTADOS

La retroalimentación que recibimos por parte de los estudiantes es muy positiva ya que realmente disfrutaron de un entorno dinámico, lúdico, motivador y organizado.

El uso de los diferentes recursos y actividades que proporciona el Aula Virtual se constituyó en un pilar fundamental en el desarrollo dinámico y práctico del curso.

Los resultados se pueden evidenciar en la Tabla VI, donde se comparan los resultados cuantitativos de los promedios finales del curso, comparando los resultados desde el 2013 (año en el que se empezó a dictar el curso) hasta el 2019 (año en que se empezó a utilizar el Aula Virtual). Se puede observar claramente la evolución de los resultados y su considerable mejora en el año 2019.

En la Fig. 14 y Fig. 15 se pueden observar las tres fases de evolución del curso, en 2013 y 2014 se utilizó un enfoque tradicional de enseñanza-aprendizaje, desde el 2015 al 2018 se aplicó un enfoque lúdico y el 2019 se aplicó aún más dicho

enfoque lúdico pero complementado e integrado con recursos y actividades del Aula Virtual.

En la Fig. 14 se observa la cantidad de aprobados y desaprobados (eje Y principal) y los promedios finales del curso y la desviación estándar (eje Y secundario), demostrando la evolución a lo largo de las tres fases.

En la Fig. 15 se observa claramente la evolución en el nivel de aprobados y desaprobados a lo largo de los años, quedando clara la evolución positiva en el tiempo.

La utilización del Aula Virtual ha sido un factor fundamental para lograr estos resultados, permitiendo impartir el conocimiento con una mejor organización, además de permitir una comunicación docente-estudiante más efectiva y activa, motivando a los estudiantes en la entrega de sus tareas, el uso del chat para absolver dudas, la difusión de videos de cursos de universidades de clase mundial y la aplicación de videojuegos y del CodingBat para brindar un entorno dinámico, lúdico y motivador.

Hay que considerar que en el curso siempre tuvimos un problema en el rendimiento de los estudiantes, así el nivel de aprobación y de los promedios finales siempre fueron bajos los dos primeros años en que se dictó el curso de forma tradicional. Después se mejoró con el uso de los entornos lúdicos; pero en el último año, con la utilización del Aula Virtual, se vieron favorecidos los resultados, tales como la cantidad de aprobados, promedio general, incluso las notas máxima y mínima aumentaron, y la desviación estándar disminuyó mostrando una mayor homogeneidad (calificación vigesimal). Tabla VI.

TABLA VI  
RESULTADOS DEL PROMEDIO FINAL

	2013	2014	2015	2016	2017	2018	2019
Cantidad estudiantes	115	120	111	102	120	130	122
Aprobados	46,34%	43,25%	53,80%	52,15%	56,10%	53,20%	64,75%
Desaprobados	53,66%	56,75%	46,20%	47,85%	43,90%	46,80%	35,25%
Promedio	9,57	9,25	10,8	11,48	11,63	11,52	12,15
Desviación Estándar	4,01	4,15	3,85	3,43	3,47	3,52	3,19
Máxima	16	16	19	18	18	18	19
Mínima	2	1	1	3	1	2	4

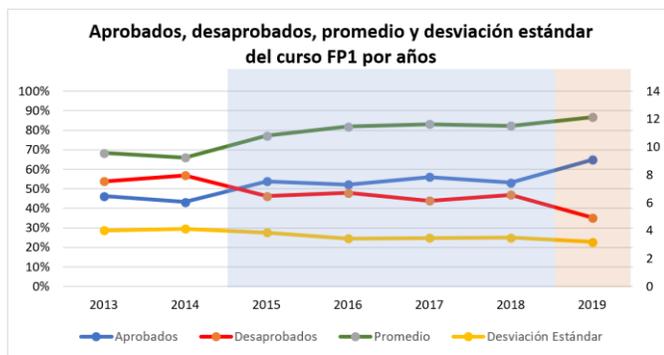


Fig. 14 Resultados del nivel de aprobados, desaprobados, promedio y desviación estándar del curso de FP1 por años y fases

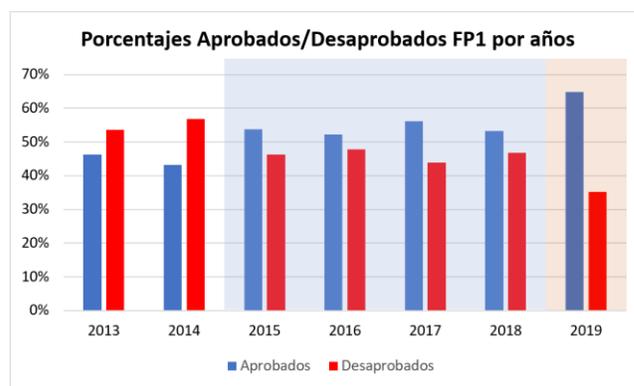


Fig. 15 Resultados comparativos del nivel de aprobados y desaprobados del curso de FP1 por años y fases

## CONCLUSIONES

En este artículo se ha compartido la experiencia de la utilización de un Aula Virtual para el curso de Fundamentos de Programación 1 para lograr un proceso enseñanza-aprendizaje más efectivo y práctico.

Se han presentado las herramientas lúdicas, herramientas de programación en vivo, recursos y actividades del Aula Virtual 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, reutilización, complejidad, algoritmo, programa, función o método, arreglo estándar, ArrayList, eficiencia, clase, objeto, atributo, método, excepción, mensaje y argumento; permitiendo suavizar la curva de aprendizaje de los contenidos del curso, aunque sean teóricos y abstractos.

Se ha demostrado que una aproximación de la enseñanza basada en entornos lúdicos y enmarcada en un Aula Virtual es efectiva para una generación de estudiantes que son denominados post-millennials o Generación Z, que entre otras características presentan desinterés por aquello que no los motiva en forma visual, que no es dinámico y que no les llama la atención.

La utilización del Aula Virtual constituye una ayuda tanto para el docente, ya que su utilización le permite realizar una labor más ordenada y comunicativa, y también para el estudiante al brindarle un entorno organizado, motivador e integrador de todos los recursos que lo apoyan en la enseñanza-aprendizaje del curso.

Desarrolla la habilidad blanda de puntualidad y responsabilidad al presentar los trabajos prácticos de laboratorio, ya que la actividad "Tarea" del Aula Virtual permite poner límite de tiempo para su entrega.

Muchas de las experiencias vividas durante el curso, constituyen parte de los ejemplos al enseñar los conceptos avanzados de la programación orientada a objetos en el curso de Fundamentos de Programación 2, facilitando de esta manera su posterior aprendizaje.

Consideramos que esta experiencia puede ser replicada en otras escuelas profesionales, no sólo de ingeniería, y también se puede replicar a nivel escolar primario y secundario.

## 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), pp. 99-100, 2002.
- [5] M. Aedo, E. Vidal, E. Castro & A. Paz, "Aproximación orientada a Entornos Lúdicos para la primera sesión de CS1 - Una experiencia con nativos digitales", 15th Latin American and Caribbean Conference for Engineering and Technology (LACCEI) - International Multi-Conference for Engineering, Education, and Technology, United States, 2017.
- [6] M. Aedo, E. Vidal, E. Castro & A. Paz, "Teaching Based on Ludic Environments for the First Session of Computer Programming - Experience with Digital Natives", Revista Iberoamericana de Tecnologías del Aprendizaje Volume 14, Issue 2, Article number 8736307, pp. 34-42, 2019.
- [7] M. Aedo, E. Vidal & E. Castro, "Experiencia en la enseñanza de Fundamentos de Programación Orientada a Objetos a través de la implementación de un Videojuego de Estrategia", 17th Latin American and Caribbean Conference for Engineering and Technology (LACCEI) - International Multi-Conference for Engineering, Education, and Technology, Jamaica, 2019.
- [8] Escuela Profesional de Ingeniería de Sistemas. <http://fips.unsa.edu.pe/ingenieriadesistemas/>
- [9] Java. <https://www.oracle.com/technetwork/es/java/index.html>
- [10] Moodle. <https://moodle.org/>
- [11] 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>
- [12] M. Prensky, "Digital Natives, Digital Immigrants," Horiz., vol. 9, no. 5, pp. 1–6, 2001.
- [13] J. García Cué, "Estilos de Aprendizaje y Tecnologías de la Información y la Comunicación en la Formación de Profesorado". Tesis Doctoral. España-UNED, 2006
- [14] R. Felder, L. Silverman, "Learning and teaching styles in engineering education," Eng. Educ., vol. 78, no. June, pp. 674–681, 1988.
- [15] Lightbot 1.0. <https://armorgames.com/play/2205/light-bot>
- [16] Lightbot 2.0. <https://armorgames.com/play/6061/light-bot-20>
- [17] Alice. <http://www.alice.org/>
- [18] Stanford University. CS1 - Programming Methodology - Java. <https://www.youtube.com/playlist?list=PLBB793100944713F8>
- [19] Massachusetts Institute of Technology. MIT Open Courseware. Introduction to Programming in Java. <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-092-introduction-to-programming-in-java-january-iap-2010/>
- [20] CodingBat. <https://codingbat.com/java>