

Application of Geolocated Activities for Teaching the Fundamentals of Object-Oriented Programming and Software Modeling

Marco Aedo López, Mg.¹, Juan Juárez Bueno¹, Ing. Eveling Castro Gutiérrez, Mg.¹

¹Universidad Nacional de San Agustín de Arequipa, Perú, maedol@unsa.edu.pe, jjuarez@unsa.edu.pe, ecastro@unsa.edu.pe

Abstract– This article describes an experience in the application of geolocated activities to teach first-year students of an engineering university education program, the fundamentals of object-oriented programming and software modeling, generating a motivating and effective environment for their learning. Object-oriented programming is based on general concepts that are known and applied by human beings in general in their daily lives, but its teaching creates challenges that must be faced, mainly due to its abstract nature. Software modeling is a very useful tool for any professional who develops software and it is best to learn it from the beginning of their university studies. It is shown that a teaching based on the application of geolocated activities to teach the concepts of object-oriented programming and software modeling is motivating and effective for students belonging to the Centennial Generation.

Keywords-- Programming fundamentals, Object-oriented programming, Geolocated activities, Software modeling, Unified Modeling Language.

Digital Object Identifier: (only for full papers, inserted by LACCEI).

ISSN, ISBN: (to be inserted by LACCEI).

DO NOT REMOVE

Aplicación de Actividades Geolocalizadas para la Enseñanza de los Fundamentos de la Programación Orientada a Objetos y del Modelado de Software

Marco Aedo López, Mg.¹, Juan Juárez Bueno¹, Ing. Eveling Castro Gutiérrez, Mg.¹

¹Universidad Nacional de San Agustín de Arequipa, Perú, maedol@unsa.edu.pe, jjuarez@unsa.edu.pe, ecastro@unsa.edu.pe

Abstract– *Este artículo describe una experiencia en la aplicación de actividades geolocalizadas para enseñar a los estudiantes de primer año de un programa de educación universitaria de ingeniería, los fundamentos de la programación orientada a objetos y del modelado de software, generando un entorno motivador y efectivo para su aprendizaje. La programación orientada a objetos tiene sus fundamentos en conceptos generales que son conocidos y aplicados por los seres humanos en general en su vida diaria, pero que su enseñanza genera retos que deben afrontarse, debido fundamentalmente a su naturaleza abstracta. El modelado de software es una herramienta muy útil para cualquier profesional que desarrolla software y es mejor aprenderlo desde el inicio de sus estudios universitarios. Se demuestra que una enseñanza basada en la aplicación de actividades geolocalizadas para enseñar los conceptos de la programación orientada a objetos y del modelado de software es motivadora y efectiva para los estudiantes pertenecientes a la Generación Centennial.*

Keywords– *Fundamentos de programación, Programación orientada a objetos, Actividades geolocalizadas, Modelado de software, Lenguaje de Modelado Unificado.*

I. INTRODUCCIÓN

Está comprobado que los actuales estudiantes universitarios que pertenecen a la llamada Generación Z o Centennials, nacieron junto a la tecnología y la aplican en su vida diaria, tanto en su educación, entretenimiento, comunicación, etc. Los teléfonos celulares inteligentes constituyen su herramienta fundamental y los utilizan en todas sus actividades rutinarias que van desde su aprendizaje, relaciones interpersonales, diversión, etc. [1],[2].

En la Escuela Profesional de Ingeniería de Sistemas (EPIS) de la Universidad Nacional de San Agustín de Arequipa – Perú (UNSA), el curso básico de programación se denomina Fundamentos de Programación 1 (FP1), curso fundamental del plan de estudio vigente, utiliza el lenguaje de programación Java, se orienta a la enseñanza de los conceptos básicos de la programación de computadoras, tanto del paradigma procedimental como del paradigma orientado a objetos, busca desarrollar el pensamiento algorítmico en los estudiantes, brinda los fundamentos del modelado de software utilizando el Lenguaje de Modelado Unificado (UML) y pertenece al primer semestre de estudios [3].

Dentro del curso hay una unidad de aprendizaje que es muy importante y es la encargada de los conceptos

fundamentales de la programación orientada a objetos y del modelado de software, que por su naturaleza abstracta siempre generó dificultades en su proceso de enseñanza/aprendizaje.

En el 2022 se decidió tomar un enfoque diferente para la enseñanza de dicha unidad de aprendizaje, un enfoque basado en la realización de actividades geolocalizadas para la enseñanza de sus contenidos y que considere los estilos de aprendizaje de los estudiantes universitarios actuales.

En este artículo se presenta nuestra experiencia en la enseñanza de los fundamentos de la programación orientada a objetos y del modelado de software, basada en la realización de actividades geolocalizadas que guie a los estudiantes hacia un aprendizaje más efectivo y motivador.

El resto del artículo está organizado de la siguiente manera. En la sección II se describe el curso de Fundamentos de Programación 1 de nuestro plan de estudios y la unidad de aprendizaje relacionada a los fundamentos de la programación orientada a objetos y del modelado de software. En la sección III se describen las características de los estudiantes de la Generación Centennial, específicamente su estilo de aprendizaje. En la sección IV se muestra la aplicación de actividades geolocalizadas para el aprendizaje de los conceptos fundamentales del paradigma de programación orientada a objetos y del modelado de software, describiendo las diferentes misiones a cumplir para alcanzar un aprendizaje efectivo. En la sección V se muestran los resultados del rendimiento de los estudiantes, comparando el rendimiento en la unidad de aprendizaje utilizando un grupo experimental y un grupo de control. Al finalizar se presentan las conclusiones.

II. CURSO DE FUNDAMENTOS DE PROGRAMACIÓN 1 - UNIDAD DE APRENDIZAJE DE FUNDAMENTOS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS Y DEL MODELADO DE SOFTWARE

Nuestro plan de estudios vigente, se elaboró tomando en cuenta los resultados del estudiante indicados en Accreditation Board for Engineering and Technology (ABET) [3]. 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 el curso de Fundamentos de Programación 1 se busca alcanzar las competencias específicas de la Tabla I.

Digital Object Identifier: (only for full papers, inserted by LACCEI).

ISSN, ISBN: (to be inserted by LACCEI).

DO NOT REMOVE

TABLA I
CURSO FUNDAMENTOS DE PROGRAMACIÓN 1: COMPETENCIAS ESPECÍFICAS

1. Identifica, establece e integra los diferentes conceptos de programación reconociendo los componentes y características de un algoritmo.
2. Elabora, crea y codifica algoritmos para la solución de problemas reales en un lenguaje de programación.
3. Aplica, codifica y ejecuta sentencias de selección y repetición apropiadas para la elaboración de programas.
4. Introduce, analiza, y utiliza el concepto de métodos reconociendo su importancia en la programación.
5. Introduce, analiza, y utiliza el concepto de arreglos y ArrayList reconociendo su importancia en la programación.
6. Analiza, conecta e integra los conceptos fundamentales de clases, objetos, atributos, métodos, mensajes, abstracción y los fundamentos de agregación/composición, herencia y polimorfismo, interpretando el paradigma orientado a objetos.
7. Elabora diagramas básicos de clases de Unified Modelling Language (UML) reconociéndolos como fundamento del modelado de software

El curso pertenece al primer semestre de estudios de la EPIS, tiene 5 créditos, utiliza como lenguaje de programación a Java, tiene una duración de 17 semanas compuesta por 8 horas semanales (2 horas teóricas, 2 horas prácticas y 4 horas de laboratorio) y constituye el curso básico de la línea de cursos de programación de computadoras del plan de estudios. Ver Fig. 1.

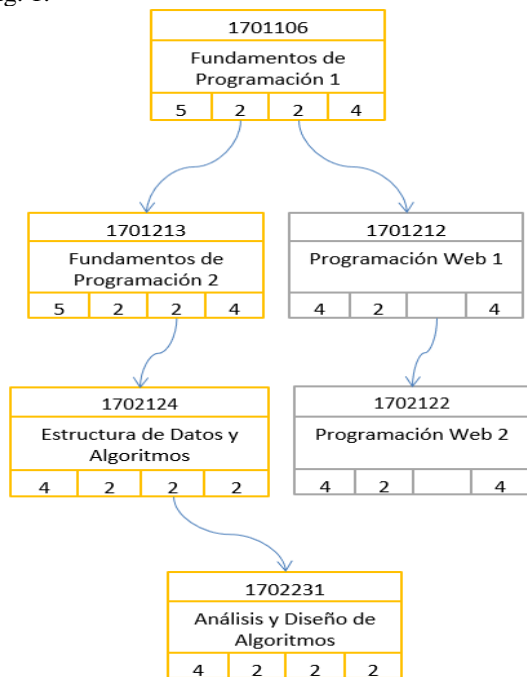


Fig. 1 Línea de cursos de programación del Plan de Estudios de la EPIS. Créditos, Horas Teóricas, Horas Prácticas, Horas Laboratorio

Considerando las competencias específicas 6 y 7, los contenidos conceptuales de la unidad de aprendizaje relacionada a la enseñanza de los fundamentos de la programación orientada a objetos y del modelado de software se resume en la Tabla II.

TABLA II
CONTENIDOS CONCEPTUALES DE LA UNIDAD DE APRENDIZAJE: FUNDAMENTOS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS Y DEL MODELADO DE SOFTWARE

1. Clases y objetos
2. Diagramas de Clases de UML
3. Atributos y métodos
4. Mensajes
5. Abstracción y algoritmos
6. Introducción a la agregación/composición
7. Introducción a la herencia y polimorfismo

El curso es evaluado en 3 fases, cada fase está constituida por un examen y por la evaluación continua (actividades en clase, ejercicios y otros). Los pesos de cada componente de evaluación están resumidos en la Tabla III. La unidad de aprendizaje analizada constituye una actividad de la evaluación continua Fase 3.

TABLA III
PESOS DE CADA COMPONENTE DE EVALUACIÓN DEL CURSO

Tipo Evaluación	Examen	Evaluación Continua	Ponderación Porcentual por fase
Fase 1	5%	5%	10%
Fase 2	15%	15%	30%
Fase 3	20%	40%	60%
Total	40%	60%	100%

III. CARACTERÍSTICAS DE LOS ESTUDIANTES DE LA GENERACIÓN CENTENNIAL

La Generación Centennial, también llamada Generación Z, Postmillennial o simplemente Nativos Digitales, está constituida por personas nacidas junto al internet. No existe un consenso sobre el rango de años en los cuales nacieron los individuos de esta generación, pero se puede indicar que abarca desde el último lustro del milenio pasado hasta la primera década del nuevo milenio, aproximadamente desde 1995 hasta 2010 [4].

Se les califica como una segunda cohorte de nativos digitales, educados y formados en un contexto dominado por el internet y sus extensiones móviles e inteligentes. Son personas muy integradas a la sociedad de la información y del conocimiento, son hiperconectados, utilizan exhaustivamente las redes sociales, tienen capacidad de respuesta rápida, deseo de inmediatez e interacción continua [5],[6].

En [7] se hace un estudio del estilo de aprendizaje de los estudiantes en nuestra universidad y se confirma que tienen un estilo de aprendizaje más visual y kinestésico que auditivo. Valoran sentirse involucrados en su propio aprendizaje y en interactuar para alcanzarlo. Construyen sus conceptos en base a objetos digitales, absorben rápidamente la información de imágenes, videos y otros objetos multimedia, adquieren información en simultáneo de diversas fuentes, esperan respuestas instantáneas y utilizan los smartphones y las redes sociales como herramientas de aprendizaje.

Aplicar estrategias de enseñanza del siglo pasado a estudiantes de esta generación no da buenos resultados, a ellos se les complica mantener la atención en algo por mucho tiempo, así desarrollar cualquier curso sólo realizando clases magistrales se contrapone con su estilo de aprendizaje [8].

Los educadores actuales deben adaptarse a estos nuevos tiempos, deben considerar dichos aspectos y no cometer el gran error de considerar a estos estudiantes iguales a los de generaciones anteriores.

IV. PROPUESTA DE LAS ACTIVIDADES GEOLOCALIZADAS A REALIZAR

La aplicación del Google Maps [9] es masiva en diferentes aspectos de nuestra vida diaria, tales como los videojuegos, los negocios, el transporte, la seguridad, etc.

Las actividades geolocalizadas son muy populares en la actualidad debido a su alta interactividad, resaltando los recorridos, trayectorias y exploraciones de lugares físicos [10].

Además, se deben destacar los beneficios obtenidos por la actividad física que brindan a los estudiantes de todas las edades [11], beneficios que se resaltan aún más en esta era pospandémica.

La UNSA brinda a su comunidad de docentes y estudiantes un aula virtual [12] basado en Moodle [13], donde cada curso de todas las escuelas profesionales tiene un soporte virtual. Esta aula virtual fue fundamental para el desarrollo de las actividades académicas incluso antes de la pandemia [3] y se puede acceder vía web o por el Moodle App [14] para smartphones con sistema operativo Android o iPhone Operative System (IOS).

Se desarrollaron las actividades geolocalizadas para la unidad de aprendizaje descrita previamente, con el objetivo de despertar la expectativa, la motivación y romper barreras en los estudiantes de esta generación, permitiendo que se sientan involucrados en su aprendizaje, se sientan motivados a aprender y lo realicen utilizando las herramientas que están tan acostumbrados a utilizar cotidianamente.

Para determinar estas actividades se siguieron las sugerencias de [15] sobre técnicas para los estilos aprendizaje y enseñanza en la educación de ingenieros. Además, se siguieron las sugerencias de [16] y [17] que plantean metodologías de aprendizaje activo para la enseñanza de la programación de computadoras.

Se plantearon las siguientes actividades realizadas con smartphone, Google Maps App y Moodle App para lograr una interacción activa y permitir que los estudiantes se sientan involucrados y motivados en su proceso enseñanza/aprendizaje y alcancen un aprendizaje efectivo. Estas actividades se implementaron en el aula virtual de la UNSA utilizando un cuestionario y banco de preguntas de Moodle.

A. Ir al escenario de la actividad

Usando el Google Maps, ir al parque o plaza más cercano a su geoposicionamiento, tomar una captura de pantalla del Google Maps y una foto del escenario de la actividad. Subir ambas evidencias usando el Moodle App. Ver Fig. 2 y Fig. 3.



Fig. 2 Evidencia del mapa utilizado (escenario) por el estudiante según su geoposicionamiento



Fig. 3 Evidencia del escenario en que el estudiante realizará las misiones según su geoposicionamiento

B. Determinar el objeto de referencia

Tomar una foto a un automóvil cualquiera y subirla usando el Moodle App. Ver Fig. 4.

Se muestra el mensaje de retroalimentación: “Has tomado una foto a un objeto de tipo Automóvil. Como podrás ver hay varios objetos de tipo Automóvil en la zona”.



Fig. 4 Ejemplo de evidencia del objeto de tipo Automóvil

C. Determinar el objeto propio

Tomar una foto a un objeto NO Automóvil (objeto propio), subirla usando Moodle App y etiquetarla.

Algunos ejemplos de objetos propios elegidos por los estudiantes se ven en Fig. 5.



Fig. 5 Ejemplos de evidencia del objeto de tipo NO Automóvil

Se muestra el mensaje de retroalimentación: “BIEN!!! Has tomado una foto a un objeto de tipo Ave”.

D. Indicar 3 características del automóvil

Usando el Moodle App, ingresar 3 características cualquiera del automóvil. Ver Fig. 6.

Los estudiantes ingresaron características tales como color, placa, marca, etc.



Fig. 6 Ejemplos de atributos del objeto de tipo Automóvil

Se muestra el mensaje de retroalimentación: “BIEN!!! a las características de un objeto las llamaremos Atributos”.

E. Indicar 3 atributos del objeto propio

Usando el Moodle App, ingresar 3 atributos cualquiera del objeto propio que eligieron.

Por ejemplo, para objetos de tipo Ave ingresaron: color, tamaño, especie, etc.

F. Indicar 3 acciones que se pueden realizar con el automóvil

Usando el Moodle App, ingresar 3 acciones cualquiera del automóvil.

Los estudiantes ingresaron acciones tales como prender, avanzar, frenar, etc.

Se emite el mensaje de retroalimentación: “BIEN!!! a las acciones que puede realizar un objeto las llamaremos Métodos”.

G. Indicar 3 métodos del objeto propio

Usando el Moodle App, ingresar 3 métodos cualquiera del objeto propio que eligieron.

Por ejemplo, para objetos de tipo Ave ingresaron: volar, caminar, cantar.

H. Vamos a clasificar los objetos

Usando el Moodle App, responder ¿cómo podría clasificar a todos los objetos de tipo Automóvil que puede ver en ese momento en su escenario?. Ver Fig. 7.



Fig. 7 Ejemplos de objetos de tipo Automóvil

Los estudiantes ingresaron Automóvil, Auto, Carro o Coche.

Se emite el mensaje de retroalimentación: “BIEN!!! Todos los objetos automóbiles que ves se pueden clasificar como de tipo Automóvil”.

“Se creará la clase Automóvil a partir de la cual se podrán crear n objetos. Una clase se representará gráficamente en UML con un rectángulo y su nombre”.



Fig. 8 Representación de una clase en UML

I. Vamos a clasificar los otros objetos

Usando el Moodle App, subir fotos de 3 objetos de la clase NO Automóvil elegida y responder ¿cómo podría clasificar a todos esos objetos?

Por ejemplo, para los objetos en la Fig. 9, los estudiantes ingresaron Ave y para los objetos de la Fig. 10 ingresaron Montaña.



Fig. 9 Evidencia de 3 objetos de clase NO Automóvil. Clase Ave

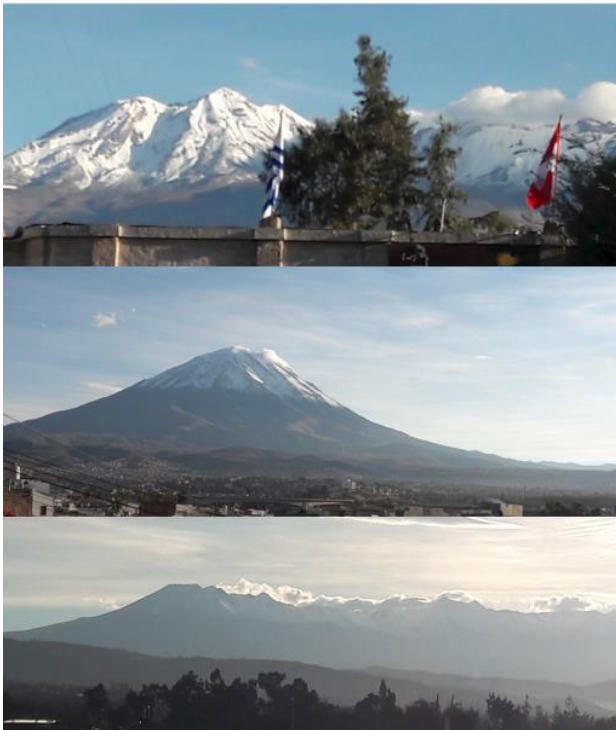


Fig. 10 Evidencia de 3 objetos de clase NO Automóvil. Clase Montaña

J. Crear la clase del objeto propio que elegiste

Usando el Moodle App, subir una captura de pantalla de la representación UML de una clase del objeto propio (utilizar cualquier herramienta libre para su creación). Ver Fig. 11.

Ave

Montaña

Fig. 11 Ejemplos de representación de clases en UML

K. Crear la clase del objeto propio que elegiste considerando atributos y métodos

Usando el Moodle App, se muestra como ejemplo la representación completa de la clase Automóvil, considerando tanto atributos como métodos. Ver Fig. 12.

Automóvil
color placa marca
prender() avanzar() frenar()

Fig. 12 Representación completa de una clase en UML. Incluye atributos y métodos

Usando el Moodle App, subir una captura de pantalla de la representación completa UML de la clase del objeto propio (utilizar cualquier herramienta libre para su creación) considerando 3 atributos y 3 métodos. Ver Fig. 13.

Ave
color tamaño especie
volar() caminar() cantar()

Fig. 13 Ejemplo de representación completa de una clase del objeto propio en UML. Incluye atributos y métodos

L. Indicar órdenes que se puede enviar a un automóvil en particular

Usando el Moodle App, suponer que puedes conducir un automóvil utilizando órdenes de reconocimiento de voz. Los métodos del automóvil que se pueden utilizar son: prender, avanzar, girar derecha, girar izquierda y frenar. El objeto que va a conducir se llama: elAuto01.

Ejemplo: elAuto01.avanzar(5) le permite al automóvil avanzar 5 metros. La sintaxis es objeto.método(argumentos)

Ingresar las órdenes a enviar al automóvil para hacer que vaya desde donde se encuentra hasta 100 metros adelante. Ver Fig. 14.

```
elAuto01.prender ()
elAuto01.avanzar (100)
elAuto01.frenar ()
```

Fig. 14 Ejemplo de órdenes a enviar a un objeto de tipo Automóvil

Se emite el mensaje de retroalimentación: “Cuando a un objeto le decimos qué realice una acción invocando algún método, le estamos mandando un Mensaje”.

M. Indicar 3 mensajes que podría enviar al objeto propio para que realice alguna tarea

Usando el Moodle App, ingresar 3 mensajes cualquiera del objeto propio para que realice alguna tarea. Ver Fig. 15.

```
miAve.cantar ()
miAve.caminar (10)
miAve.volar (20)
```

Fig. 15 Ejemplo de órdenes a enviar a un objeto de tipo Ave

N. Concepto de abstracción

Usando el Moodle App, responder: ¿Sabes cómo funciona el automóvil internamente, cómo funciona el motor?

¿Es necesario saberlo o se puede utilizar un automóvil sin conocer los detalles de cada parte? Ver Fig. 16.

La gran mayoría de estudiantes indican que no conocen cómo funciona el automóvil internamente, pero que aun así pueden utilizarlo.

Se emite el mensaje de retroalimentación: “Abstracción es el proceso de extraer sólo los datos relevantes de un objeto para así manejar mejor la complejidad”.



Fig. 16 Concepto de Abstracción utilizando el concepto de Automóvil

O. Aplicación de la abstracción

Usando el Moodle App, responder: ¿Cómo aplicar el concepto de abstracción al objeto propio (NO Automóvil)?

Los estudiantes explican con sus propias palabras cómo se aplica el concepto de abstracción a su objeto propio.

P. Las partes forman un todo

Usando el Moodle App, responder: ¿De qué partes está compuesto un automóvil?. Ver Fig. 17.



Fig. 17 Un objeto de tipo Automóvil y sus partes

Por ejemplo, ingresan: motor, puertas, parabrisas, llantas, etc.

Se emite el mensaje de retroalimentación: “Un automóvil (el todo) está compuesto de motor, puertas, etc. (las partes). A esta relación la llamaremos de agregación/composición”.

Q. Crear el diagrama de clases del objeto propio que elegiste aplicando la relación de agregación/composición

Usando el Moodle App, se muestra como ejemplo el diagrama de clases para la clase Automóvil considerando sus partes (relación agregación/composición). Ver Fig. 18.

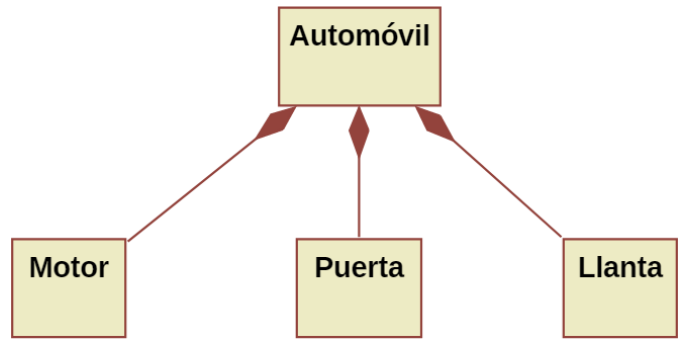


Fig. 18 Diagrama de clases considerando el Automóvil (clase todo) y sus partes (clases parte)

Usando el Moodle App, subir una captura de pantalla del diagrama de clases del objeto propio (utilizar cualquier herramienta libre para su creación). Considerar una clase todo y 3 clases parte.

R. Hay diferentes tipos de vehículos para transportarse

Usando el Moodle App, responder: ¿Qué otras opciones tienen para transportarse usando algún vehículo? Ingresar 2 tipos de transporte.

Por ejemplo, los estudiantes ingresan: bicicletas, motocicletas, scooters, etc.

Usando el Moodle App, subir capturas de pantalla de 2 objetos de los otros tipos de vehículos indicados y para cada uno ingresar su clase correspondiente. Ver Fig. 19.



Motocicleta

Bicicleta

Fig. 19 Objetos de otros tipos de vehículos

Se emite el mensaje de retroalimentación: “Los automóviles, las motocicletas y las bicicletas son Vehículos”.

S. Crear el diagrama de clases del objeto propio que elegiste considerando el mecanismo de Herencia

Usando el Moodle App, se muestra el diagrama de clases para la clase Automóvil considerando el mecanismo de herencia. Superclase Vehículo y 3 subclases. Ver Fig. 20.

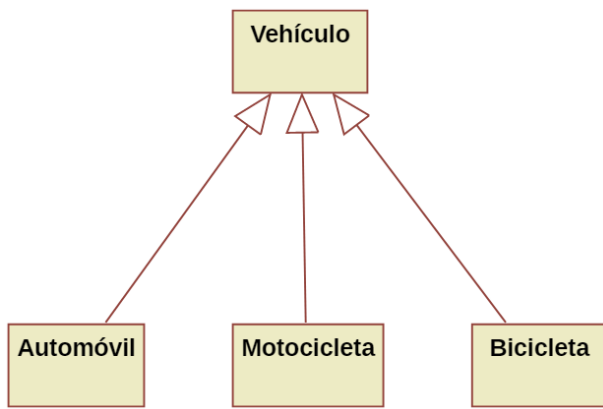


Fig. 20 Diagrama de clases considerando el Vehículo (superclase) y sus subclases

Usando el Moodle App, subir una captura de pantalla del diagrama de clases del objeto propio (utilizar cualquier herramienta libre para su creación). Considerar una superclase y 3 subclases.

T. Pero cada tipo de vehículo tiene su propia forma de utilizar

Se puede avanzar usando cualquier tipo de vehículo, pero la forma en que lo hace cada tipo de vehículo es diferente.

La misma acción “avanzar” se realiza de diferentes formas según el tipo de vehículo, eso se llama polimorfismo.

Usando el Moodle App, indicar qué pasos seguir para hacer que un automóvil avance, que una motocicleta avance y una bicicleta también lo haga.

Se emite el mensaje de retroalimentación: “Cada tipo de vehículo tiene su propia forma de avanzar, aunque se pueden parecer cada uno lo hace a su manera”.

U. Crear el diagrama de clases del objeto propio que elegiste considerando el mecanismo de Herencia y polimorfismo

Usando el Moodle App, se muestra el diagrama de clases para la clase Automóvil considerando el mecanismo de herencia y el método polimórfico “avanzar”, el cual es sobrescrito en cada subclase. Ver Fig. 21.

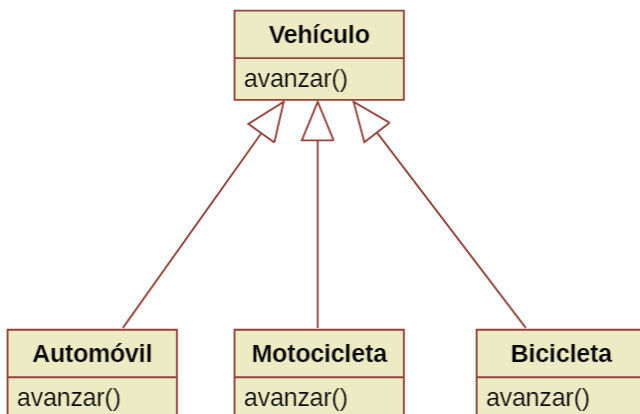


Fig. 21 Diagrama de clases considerando Herencia y Polimorfismo. Método polimórfico “avanzar”

Usando el Moodle App, subir una captura de pantalla del diagrama de clases del objeto propio (utilizar cualquier herramienta libre para su creación). Considerar una superclase, 3 subclases y un método polimórfico.

V. EVALUACIÓN Y ANÁLISIS DE RESULTADOS

El proceso enseñanza/aprendizaje de esta unidad de aprendizaje siempre tuvo inconvenientes a lo largo de los años y como medida de mejora continua es que se aplicaron estas actividades geolocalizadas.

Aplicar estas actividades geolocalizadas generó un entorno motivador y efectivo para los estudiantes, de tal forma que ellos construyeron su conocimiento de forma incremental y experimental al ir cumpliendo las misiones encomendadas.

En la Tabla IV se muestran los resultados de la encuesta sobre el nivel de satisfacción de los estudiantes del grupo experimental con la aplicación de las actividades geolocalizadas. Un 95% de los estudiantes indicaron que tenían un nivel de satisfacción entre moderado y extremo con la experiencia.

En la Tabla V se muestra la comparación de los resultados obtenidos en la evaluación de la unidad de aprendizaje correspondiente, evaluación que considera los fundamentos teóricos y la aplicación de los mismos a la solución de problemas (calificación vigesimal). Se observa un mayor rendimiento de los estudiantes pertenecientes al grupo experimental sobre el rendimiento de los estudiantes del grupo de control, mejorando todas las métricas, principalmente la cantidad de estudiantes aprobados y desaprobados (eje principal), promedio y desviación estándar (eje secundario). Ver Fig. 22.

TABLA IV
RESULTADOS DE LA ENCUESTA SOBRE EL NIVEL DE SATISFACCIÓN DE LOS ESTUDIANTES CON LAS ACTIVIDADES GEOLOCALIZADAS

	Frecuencia	Porcentaje
No satisfecho	0	0,0%
Poco satisfecho	2	5,0%
Moderadamente satisfecho	11	27,5%
Muy satisfecho	25	62,5%
Extremadamente satisfecho	2	5,0%
Total	40	100,00%

TABLA V
RESULTADOS DEL RENDIMIENTO DE LA EVALUACIÓN DE LA UNIDAD DE APRENDIZAJE

	Grupo de Control	Grupo Experimental
Nº Estudiantes	40	40
Aprobados	37,50%	72,50%
Desaprobados	62,50%	27,50%
Promedio	8,75	12,50
Desviación Estándar	4,27	3,76
Nota Máxima	16	18
Nota Mínima	2	3

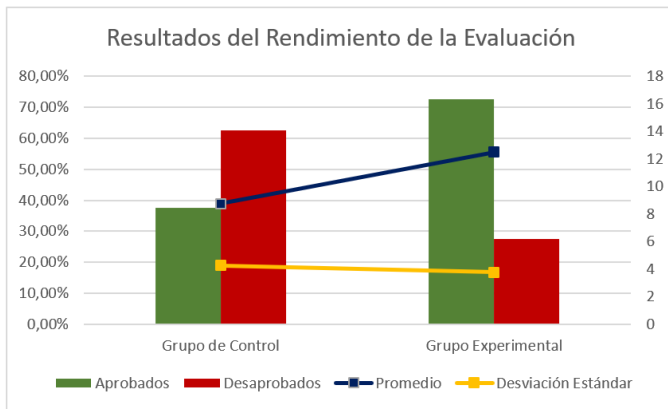


Fig. 22 Comparación del nivel de rendimiento en la evaluación de la unidad de aprendizaje considerando el grupo de control y el grupo experimental

CONCLUSIONES

En el presente artículo se presentó la aplicación de actividades geolocalizadas para la enseñanza de los fundamentos de la programación orientada a objetos y del modelado de software.

Se expusieron las actividades a realizar por los estudiantes (misiones), las que sirvieron para estimular su motivación y alcanzar un aprendizaje más efectivo.

Cambiar el sedentarismo típico por actividades geolocalizadas generó expectativa y una motivación intrínseca en los estudiantes por su novedad y por tomar en cuenta las características y estilos de aprendizaje de su generación. Se obtuvo un nivel de satisfacción alto en los estudiantes al realizar esta experiencia.

Se demostró que la enseñanza de los fundamentos de la programación orientada a objetos y del modelado de software aplicando actividades geolocalizadas logra mejores resultados en el rendimiento de los estudiantes que son parte de la Generación Centennial, los que dan más valor a las actividades en las que desempeñan una función activa.

El modelado de software tradicionalmente se enseña en años más avanzados, pero los fundamentos pueden enseñarse desde primer año sin inconvenientes.

Los conocimientos adquiridos por los estudiantes constituyeron las bases para los siguientes cursos de la línea de programación de computadoras del plan de estudios, los que utilizan a la programación orientada a objetos como paradigma.

Esta experiencia puede ser replicada en cualquier escuela profesional de ingeniería o ciencia, nacional o internacional, que considere a la programación de computadoras usando el paradigma orientado a objetos como conocimiento para su formación profesional.

REFERENCIAS

- [1] A. Pérez-Escoda, I. Aguaded, and M.J. Rodríguez-Conde, "Digital generation vs. Analogic school. Digital skills in the compulsory education curriculum," *Digital Education Review*, no. 30, pp. 165-183, 2016.
- [2] N. López Vidales and L. G. Rubio, "Trends in Youth Behaviour Change in the Face of the Media: Millennials Vs Generation Z," *Estudios sobre el Mensaje Periodístico*, vol. 27, no. 2, pp. 543-552, 2021.
- [3] M. Aedo, E. Vidal, and E. Castro, "Experience in the Use of a Virtual Classroom for Teaching Computer Programming: Development of Algorithmic Thinking and Learning a Programming Language," in 18th Latin American and Caribbean Conference for Engineering and Technology (LACCEI) - International Multi-Conference for Engineering, Education, and Technology, Argentina, 2020.
- [4] M. Maloni, M. S. Hiatt, and S. Campbell, "Understanding the work values of gen Z business students," *International Journal of Management Education*, vol. 17, no. 3, 2019.
- [5] B. Shatto and K. Erwin, "Teaching millennials and generation Z: Bridging the generational divide," *Creative Nursing*, vol. 23, no.1, pp. 24-28, 2017.
- [6] D. E. Yawson and F. A. Yamoah, "Understanding satisfaction essentials of E-learning in higher education: A multi-generational cohort perspective," *Heliyon*, vol. 6, no. 11, 2020.
- [7] M. Aedo, E. Vidal, and E. Castro, "Experience in the teaching of fundamentals of object-oriented programming through the implementation of a strategy videogame," in 17th Latin American and Caribbean Conference for Engineering and Technology (LACCEI) - International Multi-Conference for Engineering, Education, and Technology, Jamaica, 2019.
- [8] I. V. Bogomaz, E. A. Peskovsky, and E. A. Stepanov, "Digital didactic products for Generation Z specialized education," *Journal of Physics: Conference Series*, vol. 1691, no. 1, 2020.
- [9] Google Maps. <https://www.google.com/maps/>
- [10] S. Gallastegui, "Immersion and emersion in digital games. an analytical model," *Icono14*, vol. 16, no. 2, pp. 51-73, 2020.
- [11] T. Althoff, R. W. White, and E. Horvitz, "Influence of Pokémon Go on physical activity: Study and implications," *Journal of Medical Internet Research*, vol. 18, no. 12, 2016.
- [12] Aula Virtual UNSA. <https://aulavirtual.unsa.edu.pe/>
- [13] Moodle. <https://moodle.org/>
- [14] Moodle App. <https://download.moodle.org/mobile>
- [15] R. M. Felder and R. Brent, "Understanding student differences," *Journal of Engineering Education*, vol. 94, no. 1, pp. 57-72, 2005.
- [16] S. R. Sobral, "Two different experiments on teaching how to program with active learning methodologies: A critical analysis," in Iberian Conference on Information Systems and Technologies (CISTI), España, 2020.
- [17] M. Aedo, E. Vidal, E. Castro, and 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.