

Universidad Autónoma de Baja California

Facultad de Ciencias Químicas e Ingeniería
Maestría y Doctorado en Ciencias e Ingeniería



Tesis

PAEPOO: Plataforma para el Aprendizaje y Enseñanza de Programación Orientada a Objetos

Presentada por
Saúl Marrufo López
Para la obtención del grado de Maestro en
Ciencias por la Universidad Autónoma
de Baja California

Dirigida por el
Dr. Guillermo Licea Sandoval

Co-dirigida por el
Dr. Luis Guillermo Martínez Méndez

Tijuana, Enero 2015

Universidad Autónoma de Baja California
FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA
COORDINACIÓN DE POSGRADO E INVESTIGACIÓN

FOLIO No. 128

Tijuana, B. C., a 17 de octubre de 2014

C. Saúl Marrufo López
Pasante de: Maestro en Ciencias
Presente

El tema de trabajo y/o tesis para su examen profesional, en la
Opción TESIS

Es propuesto, por los CC. Dres. Guillermo Licea Sandoval y Luis Guillermo
Martínez Méndez.

Quienes serán los responsables de la calidad del trabajo que usted presente,
referido al tema: "PAEPOO: Plataforma para el Aprendizaje y Enseñanza de
Programación Orientada a Objetos".

el cual deberá usted desarrollar, de acuerdo con el siguiente orden:

- I.- INTRODUCCIÓN
- II.- MARCO TEORICO
- III.- PRODUCTOS DESARROLLADOS
- IV.- PRUEBAS Y RESULTADOS
- V.- CONCLUSIONES Y TRABAJO FUTURO

UNIVERSIDAD AUTÓNOMA
DE BAJA CALIFORNIA



FACULTAD DE CIENCIAS
QUÍMICAS E INGENIERÍA

P.A.

Q. Noemí Hernández Hernández
Sub-Director Secretario

Dr. Guillermo Licea Sandoval
Director de Tesis

Dr. Luis Guillermo Martínez Méndez
Co-Director

Dr. Luis Enrique Palafox Maestre
Director

Dedicatoria

*Para mi novia Diana que siempre estuvo
apoyándome para la redacción de este trabajo.*

AGRADECIMIENTOS

A mi director de tesis que me guió durante todo el transcurso del desarrollo y redacción, de la aplicación y del trabajo de tesis, respectivamente;

A mis compañeros de trabajo que siempre me daban una lluvia de ideas por hacer y así mejorar este trabajo;

Una especial mención a la maestra Leticia Palacios, que gracias a ella se pudo llevar a cabo la prueba de la aplicación en un curso real de programación.

Una especial mención al doctor Manuel Castañón, que siempre me ayudaba a mejorar mi trabajo.

Especial agradecimiento a CONACYT por financiar mi investigación y desarrollo durante dos años de estudio.

RESUMEN

La incorporación de las tecnologías móviles en la vida cotidiana de las personas, permite un acceso rápido y eficaz a la información, facilitando su distribución entre los usuarios. Sin embargo, poco ha explorado en el área educativa, específicamente para la enseñanza de programación orientada a objetos.

El objetivo principal de este proyecto de tesis es la creación de una plataforma que permita a los estudiantes utilizar sus propios dispositivos móviles para tener acceso a aquella información que les pueda ser útil para poder aprender el paradigma teniendo la comodidad de acceder desde su teléfono o tableta.

La aportación principal de este proyecto de tesis, dentro del objetivo principal, se puede resumir de la siguiente manera. En primer lugar se desarrolla una aplicación basada en web para visualización por medio de dispositivos móviles a través de Internet.

La segunda aportación, y considerada la aportación principal, es obtener el interés de aquellos estudiantes que no estén interesados en aprender a programar y aquellos que están seguros que no aprenderán a programar, ya que se utiliza el aprendizaje móvil; los estudiantes utilizan sus teléfonos para hacer prácticamente cualquier cosa, usando esta portabilidad, esta familiaridad, ahora para poder aprender a programar.

Por último, se lleva a cabo una evaluación de las aplicaciones que se utilizan en la plataforma para conocer qué tipo de estudiantes son los que más la utilizan, en qué momentos la utilizan y, en general, la percepción que tienen sobre ella.

ABSTRACT

The incorporation of mobile technologies in people's daily life, allows quick and efficient access to information, facilitating their distribution among users. However, little has been explored in education, specifically for teaching object-oriented programming.

The main objective of this thesis project is to create a platform that allows students to use their own mobile devices to access information that may be useful to learn the paradigm, having access from the comfort of their phone or tablet.

The main contribution of this thesis project within the main objective can be summarized as follows. First, a web-based application is developed for visualization using mobile devices via Internet.

The second contribution, and considered the main contribution is to obtain the interest of students who are not interested in learning to program and those who are sure they will not learn to program, because mobile learning is used; students use their phones to do almost anything, using this portability, this familiarity, now to learn programming.

Finally, we carried out an evaluation of the applications that are used in the platform, to know what kind of students they are most use it, at which times the use and, in general, the perception of it.

ÍNDICE GENERAL

1. Introducción	1
1.1. Antecedentes.....	1
1.1.1. Trabajo Previo.....	3
1.2. Protocolo.....	4
1.2.1. Planteamiento del problema.....	4
1.2.2. Justificación.....	4
1.2.3. Hipótesis.....	5
1.2.4. Objetivos.....	5
1.2.5. Metas.....	6
1.2.6. Metodología.....	6
2. Marco teórico	8
2.1. Estilos de aprendizaje.....	8
2.2. Tipos de estudiantes.....	10
2.3. Programación orientada a objetos al natural.....	12
2.4. Aprendizaje Flexible.....	13
2.4.1. E- learning.....	13
2.4.2. Mobile Learning.....	14
2.4.2.1. Dispositivos móviles.....	17
2.5. Colaboración.....	23
3. Productos desarrollados	24
3.1. Definer: aplicación con un compendio de definiciones de los temas selectos de programación orientada a objetos con ejemplos y ejercicios.	31
3.2. CodeFactory: fábrica de código que genera a partir de preguntas cotidianas.....	38
3.3. ErrorFinder: biblioteca de errores para el lenguaje Java.....	48
4. Pruebas y resultados	54
4.1. Pruebas realizadas.....	54
4.2. Análisis de resultados.....	56
5. Conclusiones y trabajo futuro	73

5.1. Conclusiones..... 73
5.2. Trabajo futuro.....76

ÍNDICE DE FIGURAS

1.1 Subconjuntos del aprendizaje flexible.....	16
1.2 ¿Tiene teléfono celular?	18
1.3 ¿Alguno es Smartphone?	18
1.4 Sistema operativo del smartphome.....	19
1.5 ¿Tienes Tablet?	19
1.6 Sistema operativo de las tabletas.....	20
1.7 Tendencia.....	20
1.8 Tendencia vs Tiempo.....	21
1.9 Opinión personal sobre dispositivos móviles de las tabletas.....	22
1.10 Diagrama principal que muestra la estructura de PAEPOO	25
1.11 Arquitectura de PAEPOO.....	26
1.12 Pantalla principal de Paepoo.....	29
1.13 Pantalla de selección de aplicación.....	30
1.14 Menú principal con los temas de orientación a objetos.....	34
1.15 Menú por tema seleccionado, para elección de definiciones o ejemplo.....	35
1.16 Definición del tema seleccionado.....	36
1.17 Lista de ejemplos del tema seleccionado.....	36
1.18 Ejemplo seleccionado de la lista.....	37
1.19 Nivel de dominio.....	21
1.20 Selección de modelado del mundo real.....	40
1.21 Selección de entidades para modelar.....	40
1.22 Versión para tableta mostrando preguntas y código generado.....	41
1.23 Selección de atributos para la entidad.....	42
1.24 Selección de métodos para la entidad.....	43
1.25 Código final. Nivel principiante.....	43
1.26 Pantalla inicial. Nivel intermedio.....	44
1.27 Selección de atributos y métodos para la clase.....	45
1.28 Código final de la creación de un objeto.....	46

1.29	Selección de entidad para la herencia.....	47
1.30	Código final de la aplicación.....	48
1.31	Pantalla inicial de ErrorFinder.....	49
1.32	Lista predeterminada de errores.....	50
1.33	Traducción y explicación del error seleccionado.....	51
1.34	Lista de soluciones posibles a ese error.....	52
1.35	Porcentaje de uso de aplicaciones.....	58
1.36	Niveles de dominio de los usuarios en CodeFactory.....	58
1.37	Modelados más utilizados por los usuarios.....	59
1.38	Clases más utilizadas por los usuarios.....	60
1.39	Temas con más consultas en Definer.....	60
1.40	Características más utilizadas en Definer.....	61
1.41	Utilización de PAEPOO con horas y días del mes de marzo.....	63
1.42	Opinión general sobre utilización de dispositivos móviles.....	64
1.43	Opinión general sobre ventajas y desventajas de PAEPOO.....	65
1.44	Utilización de PAEPOO fuera de clase.....	66
1.45	Utilización de PAEPOO durante un examen.....	66
1.46	Estadísticas sobre que quisieran los estudiantes en PAEPOO.....	67
1.47	Estadísticas sobre contraseña olvidada.....	68
1.48	Estadísticas utilizar PAEPOO en otras asignaturas.....	69
1.49	Opinión de qué tan útil fue para los estudiantes PAEPOO.....	70
1.50	Estadísticas sobre utilización de PAEPOO con más utilidades.....	71
1.51	Opinión sobre responder encuestas en papel o digitalmente.....	72
1.52	Definición y ejemplos activos. Ejercicios inactivos.....	77

ÍNDICE DE CUADROS

1.1 Comparación de estilos de aprendizaje en distintas universidades.....	9
---	---

INTRODUCCIÓN

Según Lahtinen, uno de los principales temas en el aprendizaje de computación es el de la programación de computadoras, ya que es parte fundamental en la educación universitaria. Para los principiantes, aprender a programar en cualquier lenguaje computacional es, para la mayoría, uno de los elementos más complejos y estresantes que los estudiantes de las universidades pueden tener en su curso (Lahtinen, Ala-Mutka, & Järvinen, 2005). Hoy en día, un estudiante graduado que tiene una capacidad nula o muy escasa al escribir un programa muy simple, sin importar en qué lenguaje, es tema que no está muy fuera de la realidad y el que ocurra es algo muy común (Jenkins, 2001). En las últimas dos décadas la demanda de programadores y estudiantes interesados en la programación ha crecido rápidamente, y cursos de iniciación en la programación se han hecho cada vez más populares (Robins, Rountree, & Rountree, 2003).

1.1 Antecedentes

Diversos autores y en distintos documentos describen técnicas y metodologías para cautivar a los estudiantes o de cómo la programación se puede enseñar casi por medio de excusas, por medio de un juego. Todas estas son buenas ideas, pero rara vez hay alguna señal de evidencia concreta de que estas formas de enseñanza tienen algún impacto en el aprendizaje. Es cierto que los estudiantes lo disfrutan y que ofrece una buena respuesta, pero hay poca evidencia de que han aprendido algo que no aprenderían de una lectura tradicional (Jenkins, 2002).

En años recientes muchas universidades han adoptado enseñar a sus estudiantes la programación orientada a objetos y han discutido acerca del problema que tienen los estudiantes en aprender al nuevo paradigma. Todos estos problemas son similares en las instituciones a pesar de estar en distintos países (Weir, Vilner, Mendes, & Nordström, 2005).

Durante los últimos años se ha debatido sobre qué lenguaje de programación se debe adoptar como primer curso de programación. Ayudaría a la clase a atraer más estudiantes hacia el aprendizaje de un curso de programación, si se adoptara Java desde el primer curso; Java es un lenguaje moderno (Duke, Salzman, Burmeister, Poon, & Murray, 2000).

En muchos contextos, la "primera vez" es muy importante. La huella de la primera experiencia durará por mucho. A través de los años, se han hecho dos observaciones (Jacquot, 2002) en cuanto a la primera exposición a la ciencia computacional. Primero, tiene un gran y duradero impacto en el marco conceptual construido por estudiantes para organizar y entender conceptos computacionales, y en la forma en lo que estudiantes consideran programar. Segundo, el lenguaje de programación tiene un fuerte impacto en el aprendizaje.

Un problema que tiene gran significancia, es la cantidad de estudiantes que entran a cursos de programación en los primeros meses de universidad. Todos los estudiantes aprenderán a programar de distintas maneras, aunque se les haya enseñado y evaluado de una sola. (Jenkins & Davy, 2002).

En la experiencia se demuestra que aprender a programar es más complicado que enseñar a programar. Como disciplina académica, computación está aún en etapa infantil. A pesar de esto, se está expandiendo muy rápidamente y no hay señales que esta tendencia llegue a su fin. Esta expansión trae consigo una mayor financiación y una mayor importancia para la disciplina, sino también trae consigo una gran diversidad. Por lo tanto, es aún más complejo para los profesores encontrar una forma adecuada de introducir a los estudiantes la programación.

Una clase que está fallando claramente en aprender a programar, puede simplemente considerar que no están motivados. Esta es una forma fácil y atractiva de proponer una solución, pero no es verdad, y un caso de un mal trabajador culpa a sus herramientas auxiliares para la enseñanza.

1.1.1 Trabajo previo

A través de muchos años, con la finalidad de facilitar las labores a profesores de programación, se han desarrollado herramientas que apoyen a los alumnos para el entendimiento de conceptos teóricos y para el desarrollo de programas a través de ejercicios y prácticas o tareas dentro del curso. Diversos autores proponen gran variedad de técnicas, métodos y herramientas para estudiantes principiantes en programación orientada a objetos.

Algunos autores como J. Brown y G. Baatard proponen métodos y estrategias que ayuden a los profesores y alumnos para cursos donde la cantidad de estudiantes sea muy grande, donde se asista al estudiante a través de la interpretación de código y manejador de errores que les indique su problema y les muestre una posible solución al mismo, no sólo informativo; también insisten en ayudar al estudiante más con código que sólo teoría y se proponen libros de texto y recursos en línea para ejemplos de código; con esto los alumnos pueden recibir ayuda sin esperar al final de clase después de que se hayan auxiliado a todos los alumnos, pues esto es para guiar al estudiante sin ayuda externa y siendo más directos con la solución de sus problemas (Brown & Baatard, 2008).

I. Allison y P. Orton desarrollaron una herramienta que hace más intuitiva la introducción a la programación, además de más simple, reduciendo la complejidad de las tareas y un simple ejercicio. Se trabaja con el lenguaje de programación C++, proveyendo una interfaz virtual de C++. Ellos proponen ejercicios donde son correctos pero les falta un poco de información, así los estudiantes tienen que encontrar ese código faltante, es decir, lo único que realizan es una edición del código y su posterior compilación (Allison & Orton, 2002).

1.2 Protocolo

1.2.1 Planteamiento del problema

Enseñar programación orientada a objetos ha sido un reto durante mucho tiempo. Estudiantes ya graduados que tiene una habilidad totalmente nula para codificar el más simple programa es muy común. Este artículo explora la problemática en la enseñanza en cursos de programación en las universidades donde los cursos se imparten en un número grande de estudiantes y como identificar los diversos tipos de estudiantes, para poder así ofrecer una solución óptima para cada nivel de conocimiento dependiendo la experiencia en programación que han desarrollado. ¿Qué características debe tener una herramienta que pueda ayudar a principiantes y expertos a ser más efectivos en el aprendizaje de programación orientada a objetos?

1.2.2 Justificación

La dificultad en el aprendizaje de la programación orientada a objetos no sólo recae en la dificultad del curso y la planeación del mismo, un profesor debe hacer su papel de motivador/profesor, atender a todas las necesidades personales de los estudiantes, abarcar las distintas culturas y necesidades de cada uno de ellos. Es necesario que en grupos grandes en el curso de programación, los mismos estudiantes sean maestros y estudiantes, la colaboración ha demostrado ser una ayuda eficaz en el rendimiento del aprendizaje, en especial en personas que inician en el tema.

Para cubrir todas estas dificultades, hay que dejar que los mismos estudiantes encuentren la forma ideal para aprender ellos mismos, siendo guiados por el profesor que a la vez hace su papel de motivador; a través de las tecnologías móviles, creemos que podemos abarcarlas, casi el total de estudiantes tiene un teléfono celular, está familiarizado con él y puede utilizarlo en cualquier momento, debemos aprovechar esta ventaja para introducir el aprendizaje en todo

momento; dejándolos con la libertad de hacerlo a su manera, con todos los elementos que les facilite el profesor.

El aprendizaje creemos debe estar en la cotidianidad de las cosas, el aprendizaje de la programación orientada a objetos debe facilitarse si deja de ser lineal y se vuelve dinámico.

1.2.3 Hipótesis

La familiaridad que los estudiantes tienen en el uso de dispositivos móviles, muy específicamente en teléfonos inteligentes, generará un mayor interés en usar dichos dispositivos en el aula de clase, se cree que inducir estos dispositivos dentro del aula de clase, en cursos de programación, acercará de manera directa a los estudiantes con los temas del curso, teniendo siempre disponible la información necesaria para ellos. Obteniendo un interés por usar sus propios dispositivos móviles, podemos trabajar en facilitar el aprendizaje.

1.2.4 Objetivos

- Principal:
 - Diseñar una herramienta en la asistencia del aprendizaje utilizando las tecnologías móviles, específicamente en cursos de programación orientada a objetos, para la Universidad Autónoma de Baja California, que tome la atención de los estudiantes para generar interés en el curso.
- Particulares:
 - Facilitar el acceso a las herramientas de estudio que sean vistas en el curso de programación por medio de sus teléfonos móviles.
 - Proporcionar una mayor comprensión del entorno real comparado con el paradigma orientado a objetos.

- Aumentar el interés de los estudiantes en utilizar dispositivos móviles como herramienta en el apoyo del aprendizaje.
- Introducir a los estudiantes a utilizar las tecnologías para la educación.
- Identificar los aspectos más importantes en los cuales los estudiantes de primer grado tienen mayor dificultad para la comprensión del paradigma orientado a objetos.

1.2.5 Metas

- Conocer la problemática de las herramientas existentes en el área de educación para la programación orientada a objetos.
- Utilizar una metodología de enseñanza actual, para proponer una metodología de enseñanza utilizando tecnologías móviles.
- Desarrollar en los estudiantes un mayor interés en el aprendizaje por la introducción de teléfonos celulares en los cursos.
- Aumentar el rendimiento de los estudiantes en el curso.
- Implementar el prototipo en temas seleccionados en un curso de programación para estudiantes de licenciatura.

1.2.6 Metodología

La metodología para este trabajo se llevará a cabo de la siguiente manera:

- Revisión bibliográfica de herramientas actuales
- Estudio estadístico del uso de tecnologías móviles en el área de educación.
- Evaluar las estadísticas de la Universidad sobre el uso de teléfonos móviles en los estudiantes.
- Proponer hipótesis
- Evaluación estadísticas en universidades sobre el uso de dispositivos móviles en los estudiantes

- Comparación de metodologías en enseñanza
- Propuesta de solución
- Realizar un prototipo
- Evaluar el prototipo en un curso de programación de la universidad en un tema en específico
- Evaluar los resultados

MARCO TEÓRICO

2.1 Estilos de aprendizaje

Una comparación cultural de los estilos de aprendizaje es particularmente importante en ciencias de la computación por una variedad de razones. Las ciencias computacionales utilizan una gran cantidad de material de instrucción y técnicas de origen en Occidente con valores No Occidentales. Si los estilos de estudiantes no Occidentales varían considerablemente de los estudiantes Occidentales, la efectividad de esos materiales puede ser comprometida. Estas diferencias culturales en estilos de aprendizaje pueden verse reflejadas en el rendimiento, retención y compromiso de los estudiantes en un curso de Programación (Zuolkernan, Allert, & Qadah, 2006). El diseño pedagógico del curso debe ser adecuado en todos los niveles y culturas. En cuadro 1 se muestra cómo en diversas universidades alrededor del mundo el porcentaje de los estudiantes varía dependiendo los factores activo, sensitivo, visual y secuencial.

Este comparativo nos ayuda a obtener un mejor panorama de cómo es que los estudiantes se sienten más confortables a la hora de estudiar y aprender, el desarrollo de aplicaciones en el apoyo a la educación debe intentar abarcar el mayor rango posible de estudiantes.

Institución	Activo	Intuitivo	Visual	Secuencial
AUS, USA	51%	64%	79%	71%
U. of Minnesota, USA	46%	65%	90%	70%
Ryerson U., USA	53%	66%	86%	72%
U. Belo, Brazil	65%	81%	79%	67%
U. of Puerto Rico, USA	47%	61%	82%	67%
U. of Sao Paulo, Brazil	57%	68%	80%	51%
U. of Kingston, England	51%	64%	79%	71%

Cuadro 1. Comparación de estilos de aprendizaje en distintas universidades.

2.2 Tipos de estudiantes

La enseñanza de programación en cualquier lenguaje y ambiente es una dificultad que se ha venido teniendo desde hace muchos años. Pero en cursos donde la cantidad de alumnos es relativamente grande, de 20 en adelante, es aún más compleja esta tarea. Una clase típicamente puede ser dividida (C. Lin and C. Lin, 2008) en diversos tipos:

- Aquellos que están ansiosos de programar.
- Aquellos que están convencidos que son incapaces de aprender a programar.
- Los que no están interesados en aprender a programar
- Los que ya conocen la programación.

Aquellos estudiantes que son ansiosos por aprender a programar son aquellos que no se ponen ninguna barrera mental para aprender y normalmente son a quienes los profesores gustan enseñar (Brown & Baatard, 2008). Por otra parte, aquellos estudiantes que están convencidos que son incapaces de aprender a programar, vienen con barreras mentales incluso desde antes de iniciar el curso, por lo que en algún momento pierden el interés en la materia y se convencen a sí mismos que no podrán aprender a programar.

Los alumnos que no están interesados en aprender a programar, son aquellos que están en el curso únicamente por que su programa de licenciatura indica que deben estar ahí, para ellos la programación no es importante y sólo esperan poder terminar para pasar al siguiente curso. Los que ya conocen la programación son aquellos que en cursos o escuelas anteriores se les enseñó a programar y prácticamente no existe ninguna dificultad al momento de cursar programación orientada a objetos, ya conocen los temas del curso.

Es claramente imposible diseñar ambientes para satisfacer todos los posibles subgrupos de estudiantes (en cualquier caso, esto probablemente resultaría en un entorno individual para cada estudiante).(Jenkins & Davy, 2002).

Una visión alternativa de aptitud ve a los estudiantes de manera muy amplia como "académica" y "no académica" (Biggs, 1999), y considerar cómo es la mejor manera de apoyar a estos grupos contrastantes. El argumento propuesto es que el estudiante más "académico" aprenderá, en esencia, no importa lo que haga el maestro, mientras que el menos académico necesitará más apoyo y tendrá que ser forzado a tener compromiso con el curso. Esta visión presenta un argumento más para el tratamiento de estos grandes grupos de estudiantes por separado.

Pero esta visión es más amplia que la propuesta anteriormente por C. Lin. Mientras que C. Lin se enfoca en dividir aún más los grupos de estudiantes, la de Biggs engloba a "Aquellos que no están interesados en aprender a programar" y "Aquellos que están convencidos que son incapaces de aprender a programar". Este trabajo de tesis utilizará la clasificación de C. Lin para intentar obtener una mayor definición de qué grupo de estudiantes les llama la atención el aprender a programar orientado a objetos.

2.3 Programación orientada a objetos al natural

Kim B. Bruce comenta que el paradigma Orientado a Objetos (POO) requiere una nueva forma de pensar, y de enseñar; muestra cómo se pueden elegir distintas metodologías que pueden ser exitosas. Una de ellas es enseñar programación no orientada a objetos en el primer curso, por ejemplo Pascal o C, ya que ha demostrado que profesores requieren buenos ejemplos de objetos para poder enseñar bien y se requiere un buen entendimiento de lo que es programación, es donde aplica la dificultad de enseñar primero el POO por más experimentados que estén los profesores de programación (Bruce, 2004). Para los estudiantes graduados es confortable tener habilidad en más de un lenguaje de programación. Es inapropiado tener un lenguaje de programación orientado a objetos y usarlo para enseñar el paradigma tradicional de programación por procedimientos. El problema con la programación por procedimientos es que ignora la realidad que hay tras la programación orientada a objetos moderna: la reutilización de clases (Duke et al., 2000).

Aquellos que están a favor de los métodos orientado a objetos frecuentemente hacen la afirmación que el paradigma de programación orientado a objetos es "más natural" que el paradigma orientado a procedimientos porque el mundo real está lleno de cosas (objetos) que tienen atributos y comportamientos. Esta afirmación también indica que la manera de pensar en los niveles de abstracción más altos es relativamente simple. Se asume que "más natural" significa más intuitivo, más fácil de entender y más consistente con los patrones existentes (Neubauer, 2002). Más sin embargo, existen formas en las que la orientación a objetos no es "más natural". Neubauer describe los siguientes:

- Enseñanza de matemáticas
- Actividades de la vida diaria
- Lenguaje humano
- Familias y organizaciones
- Deportes

- Videojuegos
- Caos y complejidad
- Religión

Donde el paradigma orientado a procedimientos es más natural que el orientado a objetos. La programación orientada a objetos requiere una forma de pensar diferente que la programación tradicional de procedimientos. Las técnicas de orientación a objetos facilitan la construcción de sistemas complejos y los programadores veteranos tienen dificultad de hacer la transición a métodos orientados a objetos. Para la mayoría de los programadores es "más natural" la programación orientada a Procedimientos que el paradigma orientado a objetos. Neubauer cree que por diversas razones los estudiantes tienden a pensar en términos de estructuras de control centralizado en lugar de una coordinación descentralizada del comportamiento de objetos autónomos.

2.4 Aprendizaje flexible

El aprendizaje flexible es un conjunto de filosofías y sistemas educacionales, que ofrecen al estudiante una mayor elección, comodidad y personalización de la forma en que aprenden. Es la que proporciona opciones de dónde, cuándo y cómo se produce el aprendizaje.

Durante la última década se ha estado familiarizado con el término "e-learning", y ahora m-learning está surgiendo.

2.4.1 E Learning

La siguiente definición de Urdan and Weggen , muestra una definición básica para distinguir entre m-learning y e-learning: "El término e-learning abarca un amplio conjunto de aplicaciones y procesos, incluyendo aprendizaje basado en computadoras, aprendizaje basado en web, aulas virtuales y colaboración digital.

Se define e-learning como la entrega de contenido por todos los medios electrónicos, incluidos la Internet, Intranet, Extranet, radio fusión satelital, cintas de audio y video, televisión interactiva y discos compactos (Urdan, T., Weggen, 2000).

La creación de un curso E-learning es complicada y de alto tiempo de dedicación porque los profesores deben reevaluar sus cursos y elegir las técnicas y herramientas pedagógicas más apropiadas para las aplicaciones de e-learning. Para facilitar el procesos de desarrollo del curso, las universidades recomendaron a los profesores ir introduciendo gradualmente las herramientas apropiadas de e-learning (Pirani, 2004).

2.4.2 Mobile Learning

M-learning es un subconjunto de e-learning. E-learning es el macro concepto que incluye los ambientes de aprendizaje móvil y aprendizaje en línea. Para esto, una definición simple de (Quin, 2001) es muy útil: "M-learning es e-learning por medio de dispositivos móviles: Palms, smartphones y tablets".

La figura 1.1 muestra visualmente la relación entre m-learning y e-learning.

El aprendizaje flexible engloba todas las técnicas y filosofías de aprendizaje, dentro de este gran conjunto se encuentra el Contact Learning y el Distance Learning; el primero es aquel que se lleva a cabo en las aulas de clases, donde existe un profesor y un estudiante, la relación es cara a cara; el segundo es aquel donde el aprendizaje sucede a distancia. De este último engloba dos subconjuntos, el e-learning y el paper-based, el segundo es únicamente el tipo de aprendizaje donde al estudiante se le da todos los medios para estudiar y lo hace fuera de un aula de clase. Mientras el primero es el aprendizaje que se lleva a cabo por medio de cualquier dispositivo electrónico, que sea a distancia. Dentro de este conjunto existen dos subconjuntos, el Online Learning y el M-Learning, de los cuales ambos serán usados para este trabajo de tesis, haciendo énfasis en m-learning, que es el aprendizaje que se lleva a cabo por medio de dispositivos

móviles, ya sean teléfonos, palms, tabletas, handheld.... Mientras que el E-Learning, es aquel que se lleva a cabo a través de Internet.

Actualmente, los dispositivos móviles están muy inmersos en la vida cotidiana, las personas realizan la mayoría de sus actividades con ayuda de su teléfono inteligente, usualmente utilizan la alarma para iniciar el día, para calendarizar eventos y actividades, tomar fotografías y video de situaciones que ocurren día a día, y las funciones básicas de un teléfono cualquiera, enviar y recibir llamadas, así como mensajes de texto. Todas estas actividades hace más de una década eran muy limitadas, ahora se facilitan gracias a las tecnologías móviles, y para aprovechar esta inmersión de las personas en esta tecnología, se utilizará en el área educación, aprender a programar el paradigma orientado a objetos.

Kristiansen (Kristiansen, 2001) hizo la siguiente declaración sobre el crecimiento esperado de M-Learning: "Analistas de Trend esperan que las terminales móviles sean el principal dispositivo para acceder a Internet antes de 2003. Teniendo en cuenta la enorme influencia que el uso de Internet ha tenido y tendrá cada vez más en el aprendizaje, no hay duda que Internet para móviles será muy importante en el futuro.

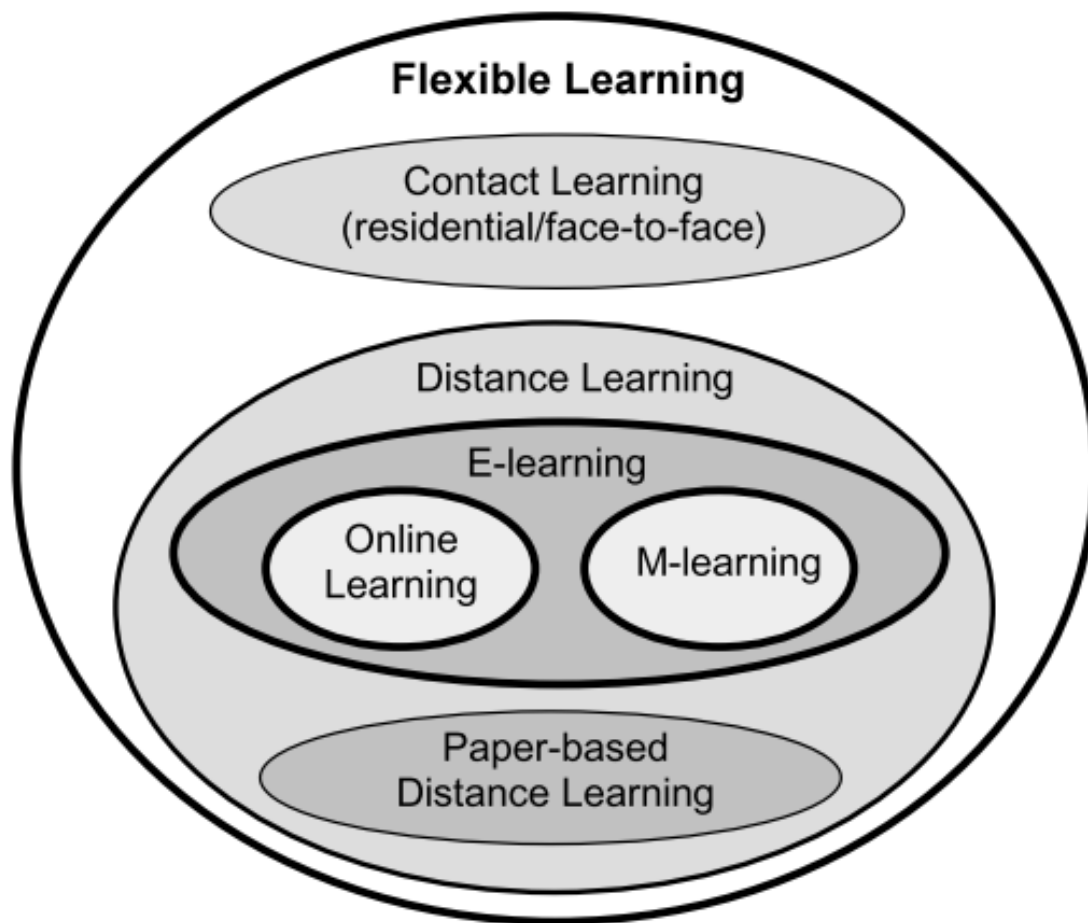


Figura 1.1. Subconjuntos del Aprendizaje Flexible.

2.4.2.1 Dispositivos móviles

Prensky, citado en Motlik 2008, declaró que había alrededor de 1.5 mil millones de teléfonos móviles en el mundo (Motlik, 2008), Salaberry (2007) coloca el número más alto, en los 2.5 mil millones (Motlik, 2008). En la semana del informe IRRODL que fue impreso, el número total de teléfonos móviles fue reportado en llegar a los 3.3 mil millones, lo que es la mitad de la población mundial (Economist.com, 2008).

En México las líneas de telefonía celular totalizaron 101 millones 339,000 suscripciones en el primer trimestre del año, casi un 7% más en comparación al trimestre inmediato anterior, el último de 2012, cuando entonces sólo se perfilaba la existencia de 100 millones de líneas (Eleconomista.com, 2013).

Lo anterior implica un crecimiento de suscripciones de 6.5% entre el último trimestre de 2012 y el primero de este año, de acuerdo con un estudio de Competitive Intelligence Unit (The CIU).

Por su parte, la penetración de las telecomunicaciones móviles entre la población mexicana alcanzó 85.6% al primer trimestre de 2013, en un universo de 118.4 millones de personas.

Un reciente estudio a estudiantes de nuestra universidad mostró que más del 50% de los estudiantes posee un teléfono inteligente con la capacidad de tener un navegador para móviles y alrededor de un 25% una tableta con las mismas posibilidades, y esta tendencia está creciendo para el segundo semestre de estudios.

La encuesta realizada en la Universidad Autónoma de Baja California arrojó los siguientes resultados:

- Cantidad de estudiantes que poseen cualquier teléfono celular sin importar sus características e indicando si poseen más de uno (Figura 1.2)

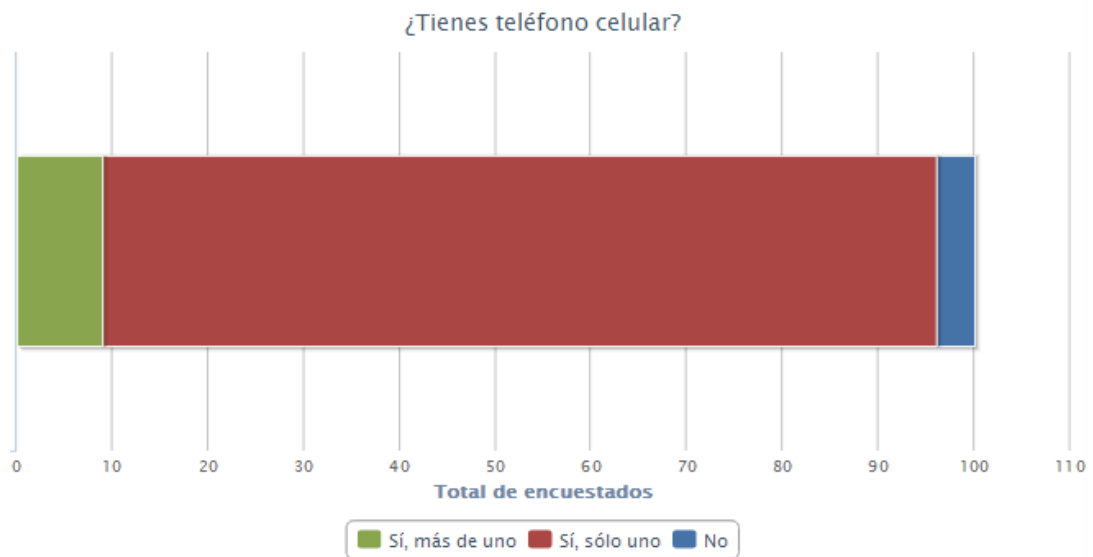


Figura 1.2. ¿Tiene teléfono celular?

- Cantidad de estudiantes que poseen específicamente un teléfono inteligente (Smartphone) de cualquier tipo (Figura 1.3).

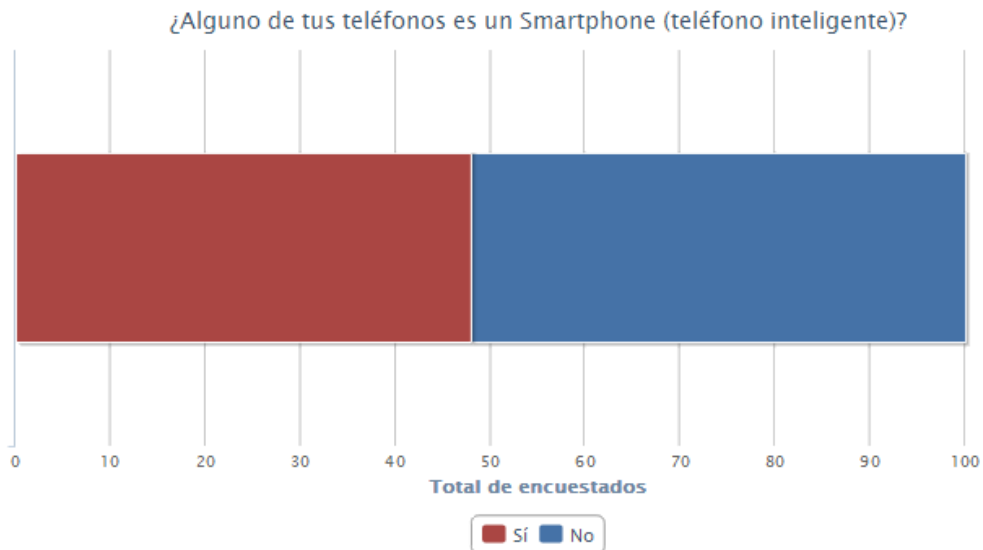


Figura 1.3. ¿Alguno es Smartphone?

- Sistema operativo predominante en los teléfonos móviles. Siendo Android que domina la encuesta con un 56.6% a favor. (Figura 1.4)

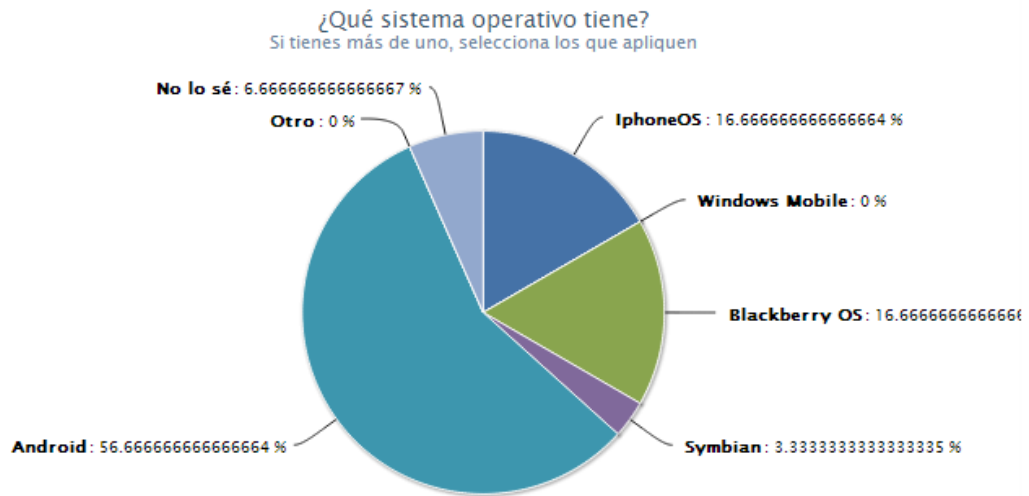


Figura 1.4. Sistema operativo del Smartphone

- Cantidad de estudiantes que poseen una tableta (Tablet) de cualquier tipo (Figura 1.5). Donde los resultados fueron mínimos para aquellos que son dueños de una.

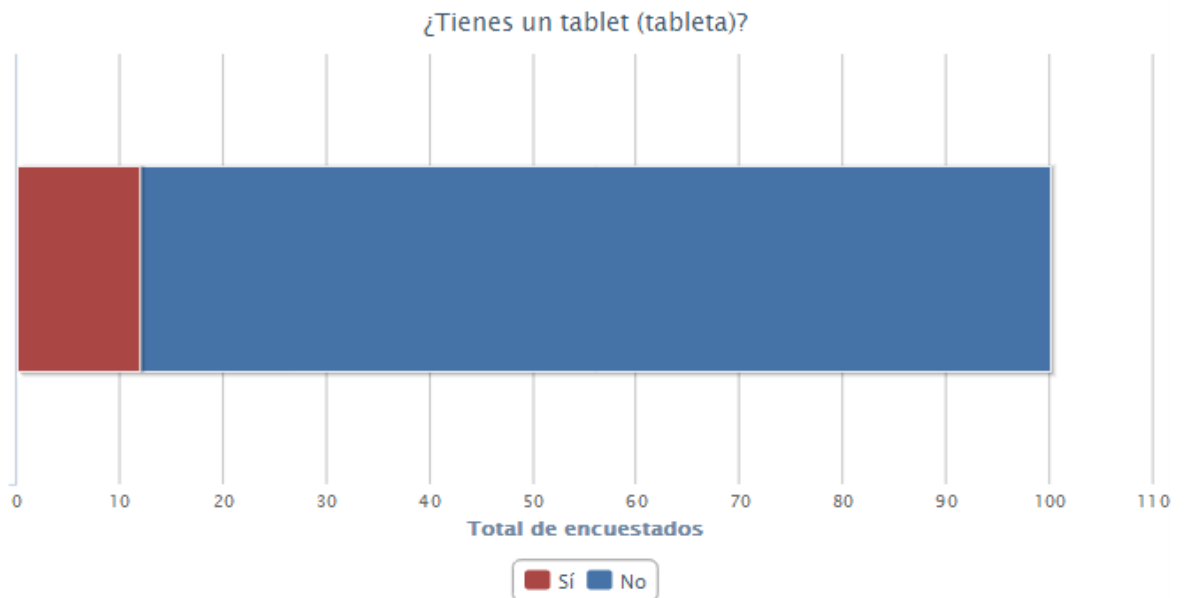


Figura 1.5. ¿Tienes Tablet?

- Sistema operativo predominante en las tabletas (tablets). Siendo iOS de Apple que domina la encuesta con un 87.5% a favor. (Figura 1.6).

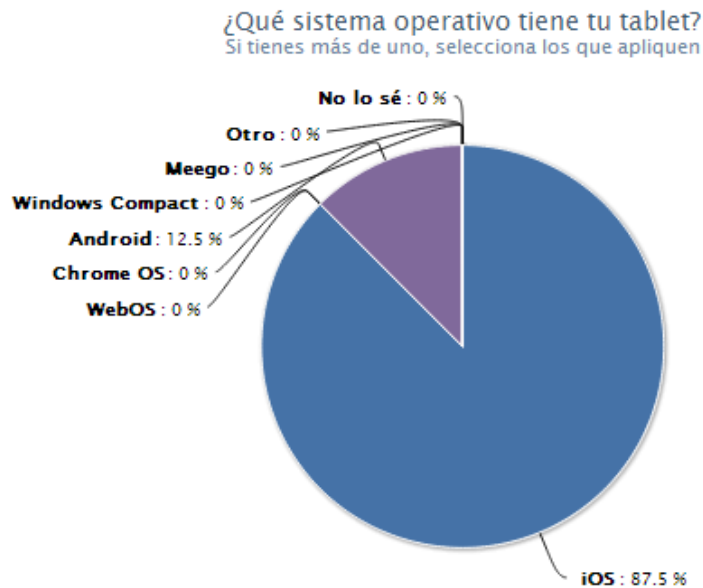


Figura 1.6. Sistema operativo de las tabletas

- Para conocer la tendencia de adquirir algún Smartphone o tablet en el futuro por parte de los estudiantes (figura 1.7), mostrando que los teléfonos inteligentes son aquellos que son más preciados para los estudiantes.

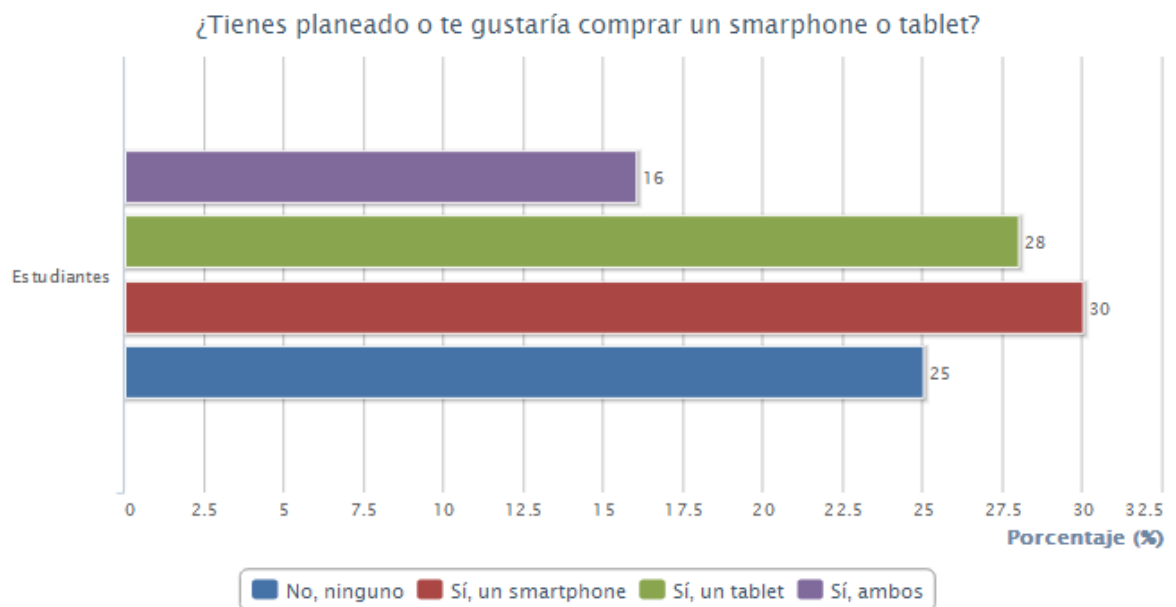


Figura 1.7. Tendencia.

- Para conocer el tiempo en que planean o quieren adquirir estos dispositivos (figura 1.8) Mostrando que dentro de los próximos 6-12 meses es el tiempo en que más usuarios planean comprar un Smartphone o Tablet.

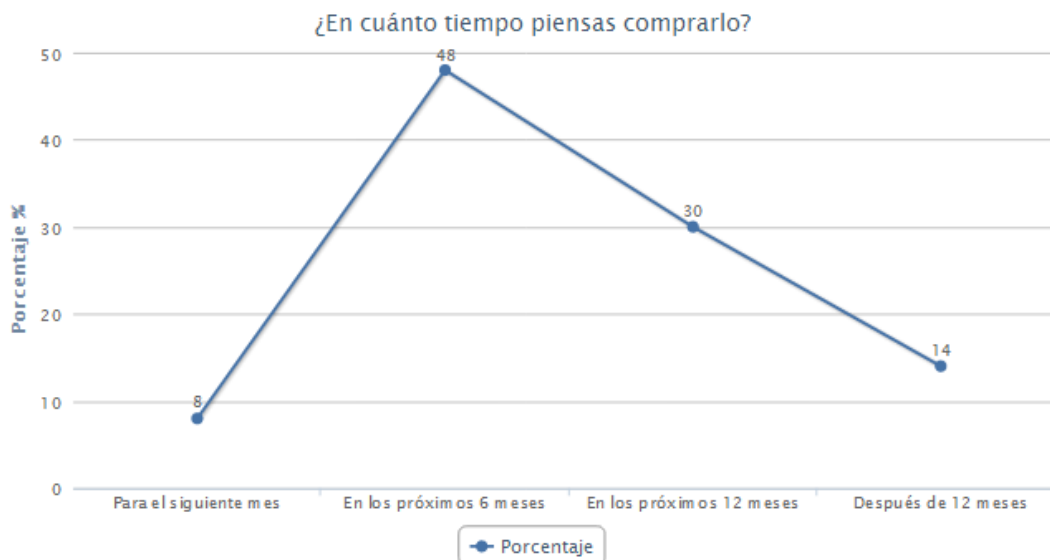


Figura 1.8. Tendencia vs Tiempo

- Se les preguntaba el que tan favorable es su opinión sobre usar sus dispositivos móviles para apoyar la educación en sus clases (figura 1.9). Mostrando una opinión muy favorable sobre usar los dispositivos móviles en las aulas de clases para apoyar la educación, en sí el aprendizaje y la enseñanza como apoyo en sus asignaturas.

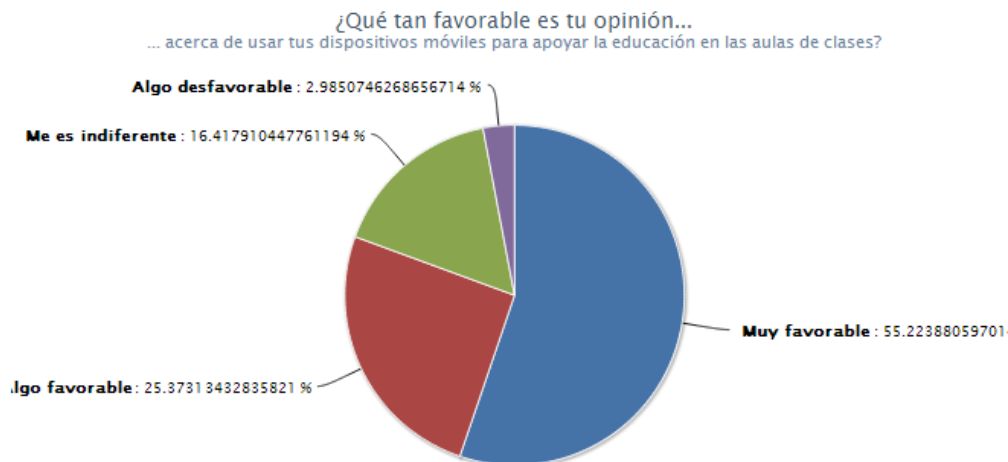


Figura 1.9. Opinión personal sobre dispositivos móviles

Con esto se cree que una integración efectiva de cualquier tecnología en la educación, requiere un gran y variado grupo de educadores participantes. Una evolución podría ser estableciendo salones de clase como los tradicionales puntos de encuentro de sociabilización las universidades (Rößling, Bär, Trompler, & Choi, 2003). Con las aplicaciones dentro de los dispositivos móviles, acceder a ellas es prácticamente desde cualquier lugar, no precisamente dentro de la universidad. Obteniendo puntos estratégicos dentro de la escuela se puede obtener acceso a todas las aplicaciones de forma gratuita y dar seguimiento a las actividades que los estudiantes realizan dentro de ellas, para así mejorar siempre aquellas características que sean deficientes.

Se utilizó una población de 174 estudiantes de todas las licenciaturas. Aplicando la encuesta personalmente por medio de una tableta, y aplicándola en grupos completos de programación por medio de un sitio web.

2.5 Colaboración

Algunos autores como Gary Willis, Hugh Davis y Eric Cooke, enfatizan en estrategias que ayuden a los estudiantes ayudarse a sí mismos, donde se comparte una misma computadora para dos alumnos y conversan entre ellos acerca de los problemas del otro y sus posibles soluciones; ellos han demostrado que el número de estudiantes que ha terminado sus ejercicios en sus laboratorios se incrementó de un 47% a un 54% en un año. Esto se desarrolló en estudiantes que no estaban familiarizados con la computación (Wills, Davis, & Cooke, 2004).

Estudiantes en clases de programación necesitan escuchar la explicación del profesor y observar la demostración o la técnica de la solución, pero también necesitan tener tiempo con el profesor para practicar, comentar y explorar. A través del entrenamiento, el estudiante desarrolla una habilidad más sólida en los conceptos de programación y pueden empezar a completar la mayoría de las tareas de programación independientemente. Esta teoría fue comprobada en la conferencia de la presentación del documento “Helping Novice Programming Students Succeed” (Black, 2006).

El aprendizaje activo está basado en el enfoque de constructivismo para enseñar cuando los estudiantes son participantes activos en el proceso de enseñanza. Este enfoque es más efectivo cuando estudiantes conducen los ejercicios y el profesor los guía en el proceso (Whittington, 2004).

D. Kalles cree que, aunque su enfoque de “Estudiantes trabajando para estudiantes” en cursos de programación, no puede ser comparado con otros, la interacción entre los estudiantes ayudó a evitar muchas dificultades de diversos índoles en su curso de programación. La interacción fue consistente cultivada a través del experimento (Kalles, 2008).

PRODUCTOS DESARROLLADOS

La plataforma es un conjunto de distintas aplicaciones que pretenden auxiliar al estudiante en su curso de programación orientada a objetos por medio de un conjunto de explicaciones, ejemplos y definiciones sobre los temas más relevantes del paradigma, así como ir resolviendo sus dudas conforme van adentrándose en su uso.

Uno de los objetivos de PAEPOO es aprovechar la familiarización que tienen los estudiantes con sus dispositivos móviles para tener forma adicional de agrandar sus conocimientos, ya que PAEPOO contiene conocimiento, aplicaciones y mucha interacción para los usuarios y el paradigma orientado a objetos. Esta plataforma no pretende sustituir al maestro en el aula de clases, pero sí proporcionarle una herramienta más para combatir el problema sobre la dificultad en el aprendizaje del paradigma.

También pretende hacer que los estudiantes utilicen los dispositivos que conocen muy bien y que lo utilizan para casi cualquier cosa en su vida diaria, los estudiantes pueden compartir y alimentar la aplicación donde todos participan y sugieren, no tiene restricciones para que puedan ayudarse entre ellos mismos y pretende ser lo más natural posible, tomando el mundo real y situaciones diarias o fenómenos muy conocidos por todos para lograr una mayor comprensión de lo que programación orientada a objetos significa y cómo se plasma el mundo real en código de Java.

En la figura 1.10 se muestra la estructura de la plataforma donde se observa el conjunto de las aplicaciones y su rol dentro de la misma.

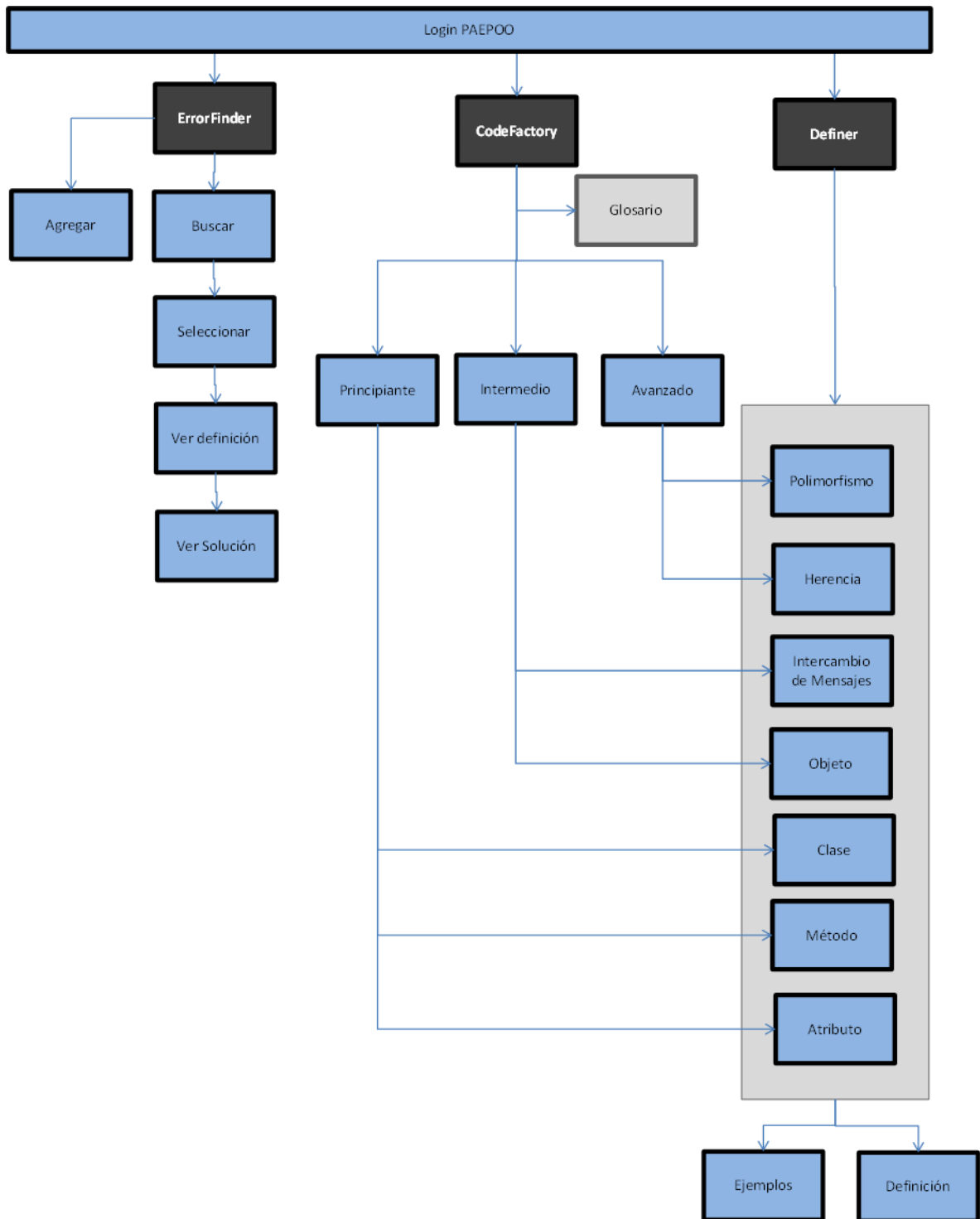


Figura 1.10. Diagrama principal que muestra la estructura de PAEPOO.

El envío de información es por medio de internet, los estudiantes se conectan por medio de sus dispositivos móviles o computadora personal a internet, la plataforma está alojada en un servidor web que envía y recibe información, ya que constantemente se están registrando los movimientos que realizan dentro de ella y actualizando la información cuando se ha cambiado los datos (Figura 1.11).

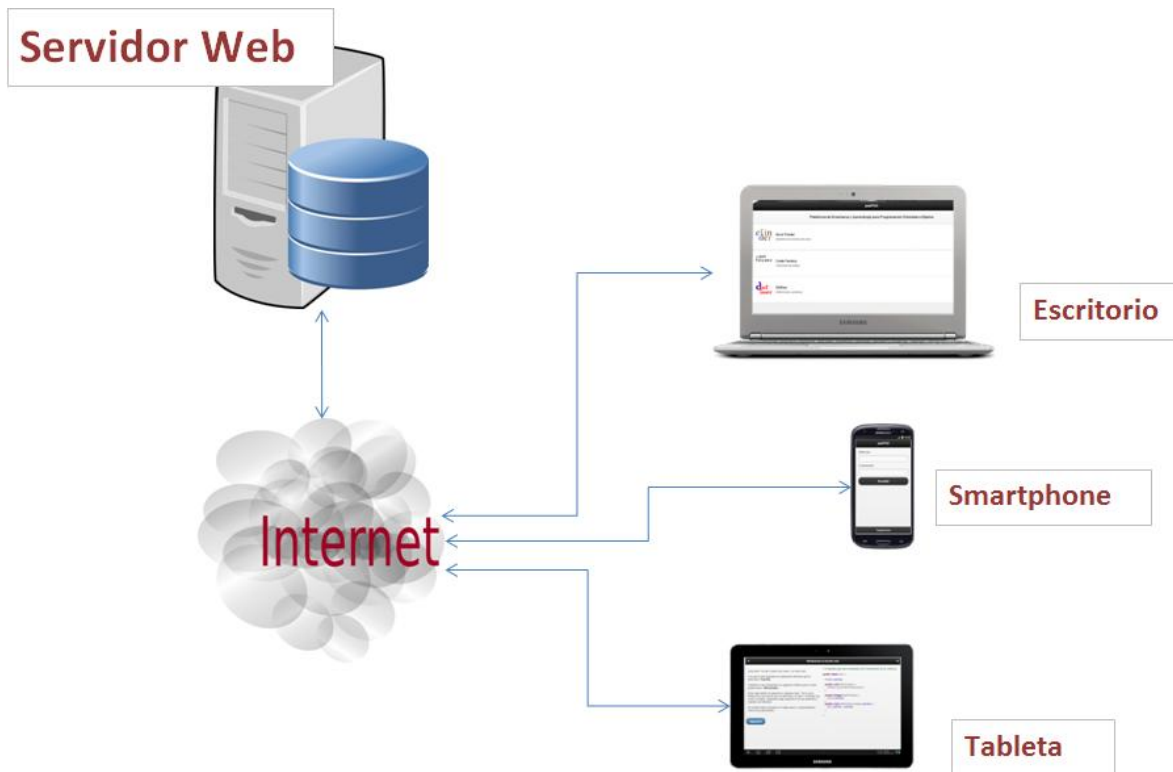


Figura 1.11 Arquitectura de PAEPOO.

Se aplicó una encuesta a estudiantes en la universidad, para todas las licenciaturas al azar, para conocer el alcance que podría tener PAEPOO entre los estudiantes que posean algún Smartphone o Tablet o cualquier dispositivo móvil.

Los resultados de la encuesta inicial mostraron que prácticamente todos los estudiantes que pertenecen a la universidad, poseen teléfono, y más de la mitad

de ellos su teléfono es inteligente, agregados al otro porcentaje que posee alguna tableta. En su mayoría dominaba el sistema operativo Android y OS de Apple.

Los resultados de la encuesta definieron las tecnologías que PAPEOO tendrá, para tener el mayor alcance dentro de los estudiantes:

- HTML 5
- CSS 3
- PHP 5.4.7.
- MySQL 5.5.27
- Apache 2.4.3
- JQuery 1.83
- JQuery Mobile 1.2.0

La elección de estas tecnologías fue por el hecho de algunos poder abarcar la mayor parte de los sistemas operativos de los teléfonos o cualquier dispositivo móvil que tenga acceso a Internet. PAEPOO funciona en los siguientes sistemas operativos:

Funcionamiento total:

- Apple iOS 3.2.* - 6.0
- Android 2.1-2.3
- Android 3.2 (Honeycomb)
- Android 4.0 (ICS)
- Android 4.1 (Jellybeans)
- Windows Phone 7-7.5
- Blackberry 6.0
- Blackberry 7
- Blackberry Playbook
- Palm WebOs
- Palm WebOs 3.0
- Firefox Mobile 15

- Chrome for Andriod 18
- Skyfire 4.1
- Opera Mobile 11.5-12
- MeeGo 1.2
- Tizen
- Samsung Bada 2.0
- UC browser
- Kindle 3 and Fire
- Nook Color 1.4.1.
- Chrome for Desktop 11-21
- Safari Desktop 4-5
- Firefox Desktop 4-15
- Internet Explorer 7-10
- Opera Desktop 10-12

Funcionamiento total, sin transiciones AJAX

- Blackberry 5.*
- Opera Mini 7
- Nokia Symbian 3

HTML Básico, pero funcional

- Blackberry 4.*
- Windows Mobile
- Plataformas más antiguas

La plataforma, funcionará bajo estas tecnologías que son visibles en los dispositivos anteriormente mencionados y además de poder ser vistos en cualquier computadora de escritorio o laptop, mientras exista un navegador que soporte estas tecnologías. Actualmente, todos los navegadores tienen la capacidad de visualizar PAEPOO.

La plataforma solicita la matrícula del estudiante y una contraseña que sólo él conoce, la figura 1.12 muestra la pantalla que el estudiante se encuentra al acceder por primera vez.



Figura 1.12. Pantalla inicial de PAEPOO.

Una vez que el estudiante ha sido autenticado, la plataforma despliega las aplicaciones que están disponibles en una lista de menú (figura 1.13).



Figura 1.13. Pantalla de selección de aplicación.

Al menú principal de las aplicaciones, el estudiante puede volver a él en cualquier momento mientras utilice la plataforma, y cambiar cuantas veces quiera.

3.1 Definir: aplicación con un compendio de definiciones de los temas selectos de programación orientada a objetos con ejemplos y ejercicios

Definer es una aplicación que utiliza los temas más importantes de programación orientada a objetos y muestra una definición amigable de todas y cada una de ellas, además de mostrar una lista de ejemplos que pretenden que el estudiante pueda comprender aún más su utilización, con ejemplos sacados de la vida cotidiana para plasmarlo en código Java. La idea principal de Definer es tener siempre a la mano, ya sea desde tu teléfono inteligente, tu tableta o desde tu laptop, las definiciones de los temas más importantes del tópico de Programación Orientada a Objetos. Las definiciones se muestran de una forma amigable para el usuario, es decir, pretende ser lo más claro y sencillo para el alumno, sin dejar de proporcionar el significado básico de lo que es la palabra.

Los problemas más comunes que tienen los estudiantes según Abdullah 2006 , son los siguientes (Abdullah Mohd Zin; Sufian Idris, 2006):

- Objeto primero vs Sintáxis primero
- Generación de programa vs Comprensión de programa
- Habilidad en instalar el compilador
- Entendimiento nivel objeto
- Entendimiento nivel clase
- Entendimiento nivel paso de mensajes
- Entendimiento nivel método
- Entendimiento nivel encapsulación
- Entendimiento nivel herencia
- Entendimiento nivel polimorfismo
- Entendimiento nivel applet
- Entendimiento nivel Componentes de Interfaz de Usuario
- Entendimiento nivel plantillas de administración
- Entendimiento nivel manejador de eventos

Basado en esta lista, se optó en eliminar los problemas técnicos (instalar compilador, y componentes de interfaz de), estilo de enseñanza (objeto primero vs sintaxis primero, generación de código vs comprensión de código) y temas específicos de Java (applet, plantillas de administración y manejador de eventos), dejando así únicamente los temas específicos de programación orientada a objetos (objeto, clases, métodos, herencia, encapsulación y polimorfismo).

Las definiciones fueron obtenidas de distintas fuentes, para obtener un mejor resultado.

Clase. Una clase es una plantilla desde la cual se pueden crear objetos. Contiene atributos y métodos que definen lo que dicha clase puede tener y hacer. Todos los objetos que se creen de ella tendrán esas características y podrán realizar las funciones que la clase tenga descritas (Pressman, 2005) (Havey, Deitel, & Deitel, 2004).

Objeto. Un objeto es una copia de una clase. Como se dice popularmente, cada objeto es una instancia de una clase. Piensa en un objeto como una variable: almacena datos, permite que se le planteen solicitudes. En teoría, puede tomarse cualquier ejemplo de la vida real (perros, edificios, servicios, etc.) y representarse como un objeto de un programa. (Bruce Eckel, 2007).

Método. Es una porción de código que ejecuta una serie de pasos. Al que puedes llamar cada vez que necesites realizar esas tareas. De esta forma, evitas estar escribiendo siempre lo que hace el método. El método puede devolver algún valor o no devolver nada. También el método puede o no recibir argumentos o valores para utilizarlos, y de igual forma, un método puede llamar a otros métodos y que a su vez, pueden llamar otros métodos. (D. Brueck, S. Tanner, 2001).

Atributo. Todo aquello que describa a una clase lo puedes transformar en un atributo de ella. Los atributos son esas variables que forman parte de una clase y en ellas se almacenan los datos que usa el objeto. El objeto soporta variables de instancia, cuyos valores son específicos del objeto (Bruce Eckel, 2007).

Herencia. La herencia es el proceso que consiste en derivar una clase, que se llama clase derivada, de otra, que se llama clase base (también llamadas clase padre y clase hijo). En otras palabras, si tienes que crear una clase nueva, y tiene los mismos atributos y métodos que una ya existente, en lugar de volver a declararlos nuevamente, utilizas la herencia para utilizar los que tiene esa clase, y posteriormente agregar nuevas funcionalidades a tu clase heredada. (Bruce Eckel, 2007).

Polimorfismo. Polimorfismo es la capacidad para enviar el mismo mensaje a objetos totalmente diferentes, cada uno de los cuales responde a ese mensaje de un modo específico. El mismo nombre del método se puede utilizar con parámetros distintos y se permite que aparentemente el mismo método sea declarado un cierto número de veces dentro de la misma clase (Pressman, 2005).

Encapsulación. La programación orientada a objetos encapsula datos (atributos) y métodos (comportamientos) en objetos. Los objetos tienen la propiedad de ocultación de la información. Esta propiedad significa que aunque los objetos pueden conocer cómo comunicarse unos con otros, no pueden conocer cómo están implementados otros objetos (los detalles de la implementación están ocultos dentro de los propios objetos). Ésta es una situación muy frecuente del mundo real (Kölling & Rosenberg, 2001).

Todas estas definiciones, en su mayoría son sólo de un autor, más sin embargo, se pretende incluir la definición de varios, para proporcionar al estudiante un panorama más amplio, pero sin dejar de ser claros y directos.

En todos los temas, se muestra un ejemplo de cómo declararlo en código Java, la forma básica sin parámetros (donde aplique) y en algunos casos un poco más complejos o flujos alternativos. Se explican las reglas de aplicación o de utilización, se proporcionan sugerencias de cuándo se debe o no utilizar, y en algunos casos mostrar ventajas y desventajas de usarlo en algunas secciones del código.

En la figura 1.14 se muestra la pantalla principal donde se puede elegir cualquier de los temas anteriormente explicados.



Figura 1.14. Menú principal con los temas de Orientación a Objetos.

En la figura 1.15 se muestra la pantalla para elegir si ver la definición, los ejemplos, o los ejercicios, aunque ésta última ha sido desactivada para este trabajo de tesis.



Figura 1.15. Menú por tema seleccionado, para elección de definiciones o ejemplos.

Además de mostrar las definiciones (como se muestra en la figura 1.16), Definer también proporciona una lista de ejemplos de cada una de las palabras, que el estudiante puede leer si la definición no le ha quedado del todo clara. Pues estos ejemplos están basados en situaciones cotidianas de distintos rubros, ejemplos de las situaciones que puede tener un mecánico al arreglar un auto, al preparar algo de comer, el reino animal muestra cómo funciona la herencia, entre otros. (Ver figura 1.17y 1.18).



Figura 1.16. Definición del tema seleccionado, como ejemplo se muestra “Método”, con la posibilidad de volver al menú del tema o al menú de la aplicación.

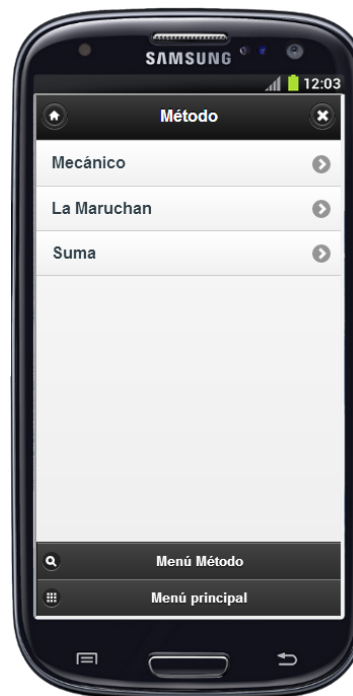


Figura 1.17. Lista de ejemplos para el tema seleccionado



Figura 1.18. Ejemplo seleccionado de la lista que describe como de la vida real se puede plasmar en código Java.

El estudiante puede volver en cualquier momento a los menús de cada tema o al menú principal con los botones que se encuentra siempre disponibles en la parte inferior de la pantalla.

Con el uso constante de PAEPOO, a través de comentarios de los estudiantes se puede llegar a crear un definición clara y concisa que se les haga muy fácil a todos, o la mayoría, de poder comprender, de la misma forma los ejemplos, los estudiantes pueden contribuir para elegir cuáles son aquellos que podrían ejemplificar mejor la vida diaria plasmando en Orientación a Objetos.

3.2 CodeFactory: fábrica de código que genera a partir de preguntas cotidianas

CodeFactory es una aplicación que por medio de una serie de preguntas al estudiante, va generando código en Java. Las preguntas están relacionadas con algún modelado del mundo real y cómo poco a poco se va transformando el ejemplo cotidiano, a código.

Está basado en tres niveles de dominio en los cuales se van agregando más temas de programación orientada a objetos, al iniciar esta aplicación lo primero que se le pide al estudiante es que seleccione el nivel de dominio, la figura X.9 muestra la pantalla donde el estudiante elige el nivel de dominio que posea. Generalmente dependerá de las clases que esté tomando en el curso para poder ir avanzado cada vez más lejos (Licea, Juárez-Ramírez, Gaxiola, Aguilar, & Martínez, 2011). Estos niveles de dominio fueron creados para que los distintos usuarios puedan utilizar esta aplicación sin necesidad de esperar a llegar a esos temas si son estudiantes ya experimentados, o en caso contrario, ir avanzando poco a poco con los primeros temas de programación orientada a objetos. El propósito de estos tres niveles es domino está diseñado para diferentes tipos de usuario (Figura 1.19):

- Principiante: En este nivel los estudiantes aplican los conceptos básicos, clase, atributos y métodos.
- Intermedio: En este nivel los estudiantes generan su primer objeto y seleccionan atributos y realizan las funciones que crearon en el nivel anterior
- Avanzado: Se introduce lo que es la herencia y se crea una subclase para ello, con nuevos métodos y atributos, además de los heredados por la clase padre.

Para los tres niveles se muestra código de lo que están realizando. El formato de código va acorde conforme se enseña en este curso, pudiendo existir la posibilidad de cambiar de acuerdo a las necesidades del maestro.



Figura 1.19. Nivel de dominio.

En el nivel principiante de CodeFactory, el estudiante tiene la posibilidad de elegir qué parte del mundo real quiere modelar y transformarlo a código (figura 1.20).

Los modelos elegidos fueron al azar, por recomendaciones de colegas de la universidad y profesores de programación orientada a objetos de niveles iniciales del curso. Los modelos pretenden plasmar en código Java lo que muestran en el mundo real, algunos de ellos muestran al estudiante una mejor visión en algunos conceptos que otros, es decir, en el modelado del mundo animal, se puede entender con mayor facilidad el concepto de herencia, por ser algo que se aprende desde niños.

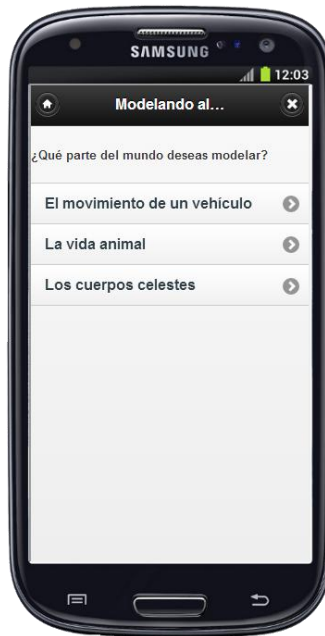


Figura 1.20. Selección de modelado del mundo real.

Una vez elegido el ejemplo a modelar, se le pregunta al estudiante qué entidades desea utilizar en su modelado y desde ese momento aparece la opción de poder ver el código que se va generando (Figura 1.21). Esto para tomar un objeto del mundo real y utilizarlo como ejemplo en el resto del modelo.



Figura 1.21. Selección de entidades para modelar.

En modalidad para tableta, se puede visualizar ambas pantallas al mismo tiempo (Figura 1.22).

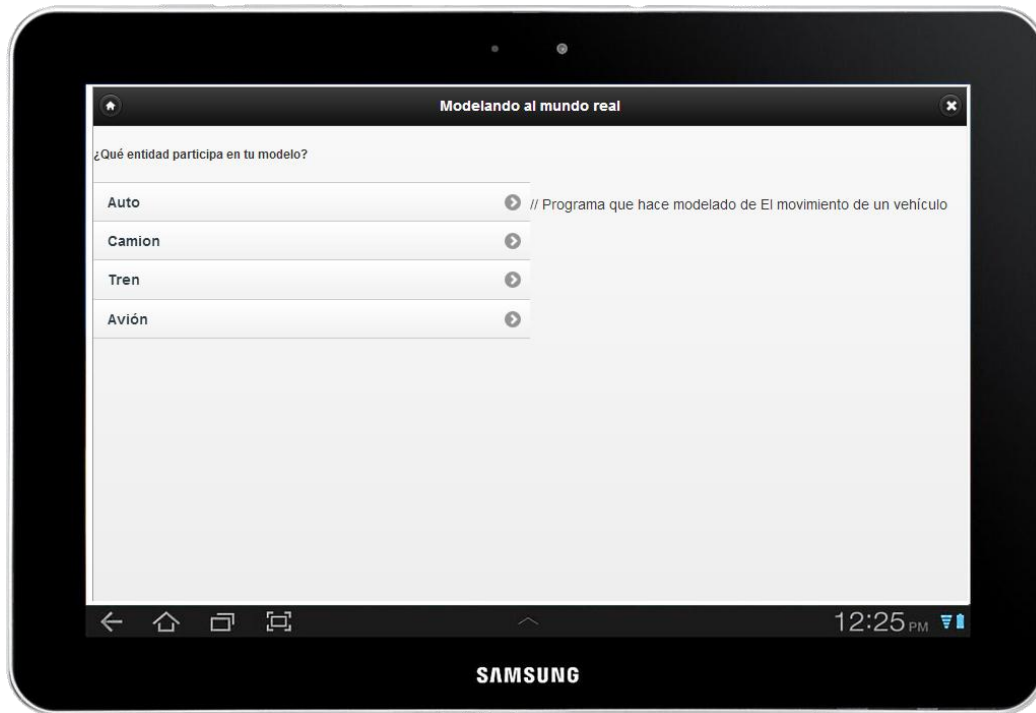


Figura 1.22. Pantalla que muestra las preguntas al estudiante y al mismo tiempo la generación de código en paralelo, sólo para versión en Tablet.

Aquellos estudiantes que posean una tableta pueden ver al mismo tiempo cómo se va generando el código al ir contestando las preguntas, en caso contrario, los estudiantes que la utilicen desde su teléfono móvil, tienen que estar cambiando de pantalla. Aunque la generación sigue siendo en vivo.

La figura 1.21 muestra el botón de “ver código” que no se muestra en la pantalla de tableta.

Una vez seleccionado la entidad, la aplicación mostrará un listado de atributos que puede tener, aquellos que pueden describir lo que es la entidad, el estudiante es libre de elegir la cantidad que desee de atributos y estos se irán plasmando en el código, conforme los va seleccionando. Los atributos son cosas comunes que puede tener dicha entidad (Figura 1.23).

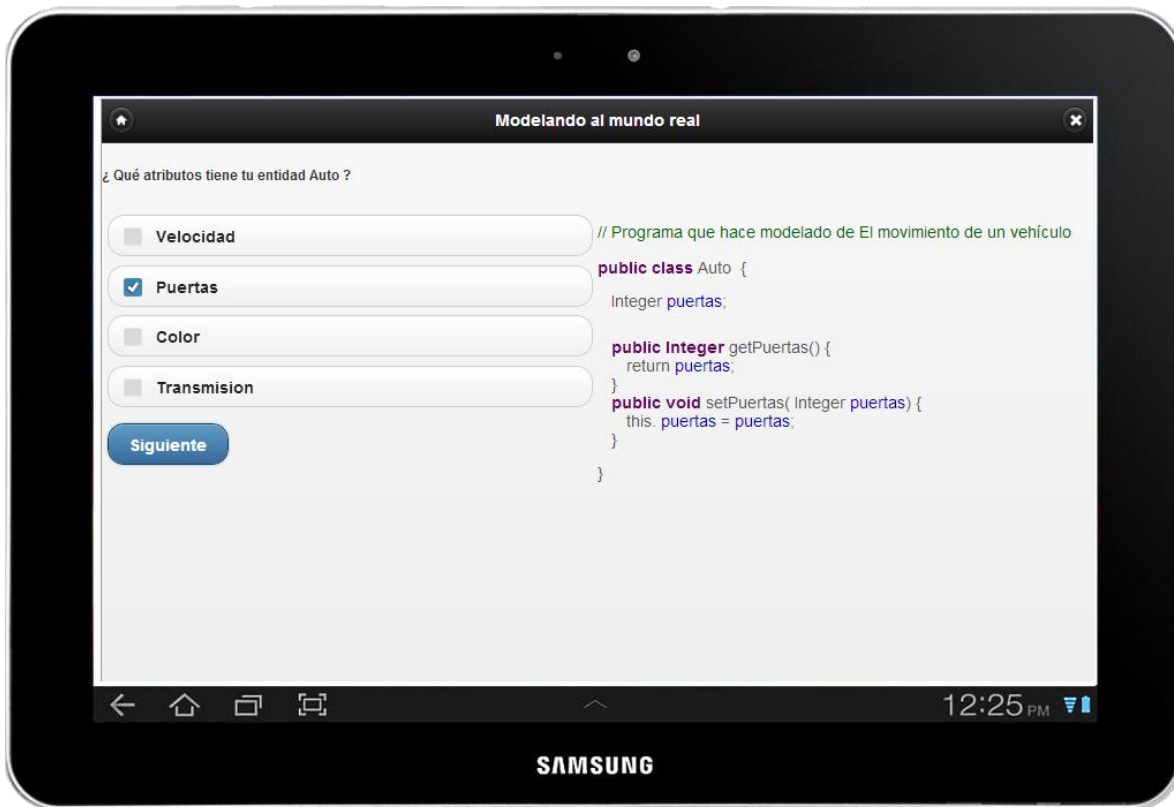


Figura 1.23. Selección de atributos de la entidad

Al elegir aquellos atributos que el estudiante desea que tenga la entidad, la aplicación ahora le mostrará aquellos métodos que su entidad puede realizar (figura 1.24). Las funciones que realiza la entidad, también son muy conocidas por el estudiante, y al ir las seleccionando se va creando código de cada una, con el estilo de aprendizaje seleccionado por el profesor.

Una vez seleccionado los métodos que desea, al estudiante se le indica que ha llegado al final del nivel principiante, donde aprendió a crear un clase, a agregarle atributos y métodos también. Las definiciones de los tipos de variables y métodos, como int, float o private, public, void, se ven en clase. Aunque la aplicación describe todas las palabras que están escritas en su código, para un mejor entendimiento y comprensión, estas se dejan para aprendizaje en clase, y para la aplicación definir que explicar algunas de ellas (Figura 1.25).

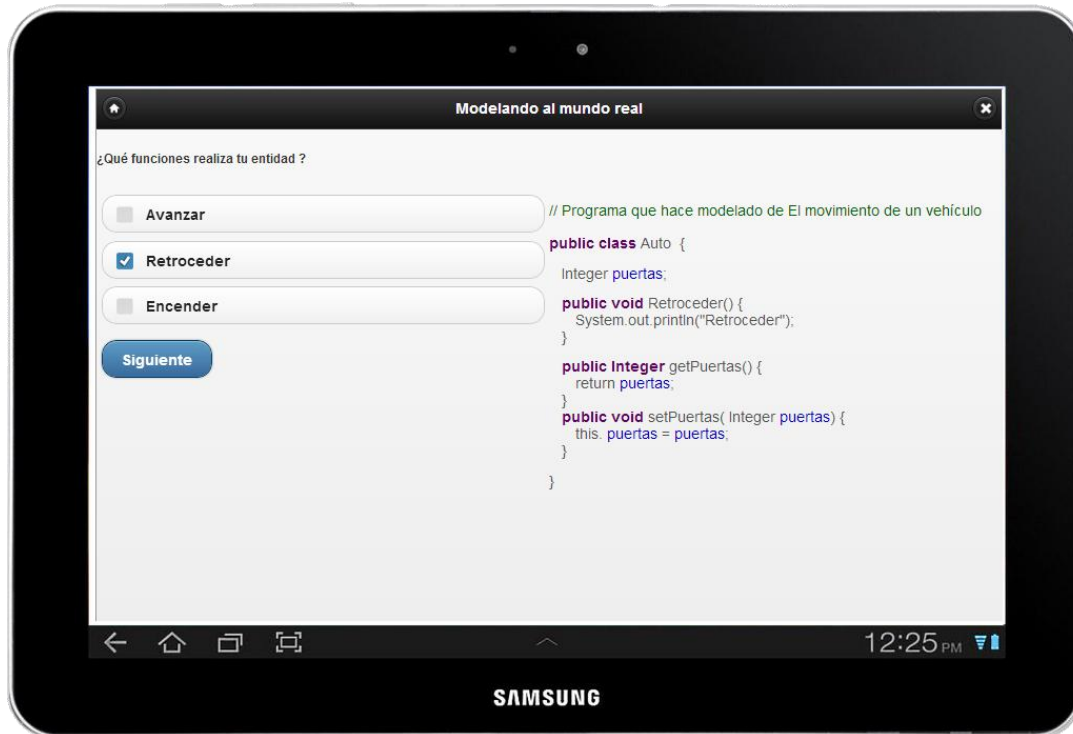


Figura 1.24. Selección de métodos.

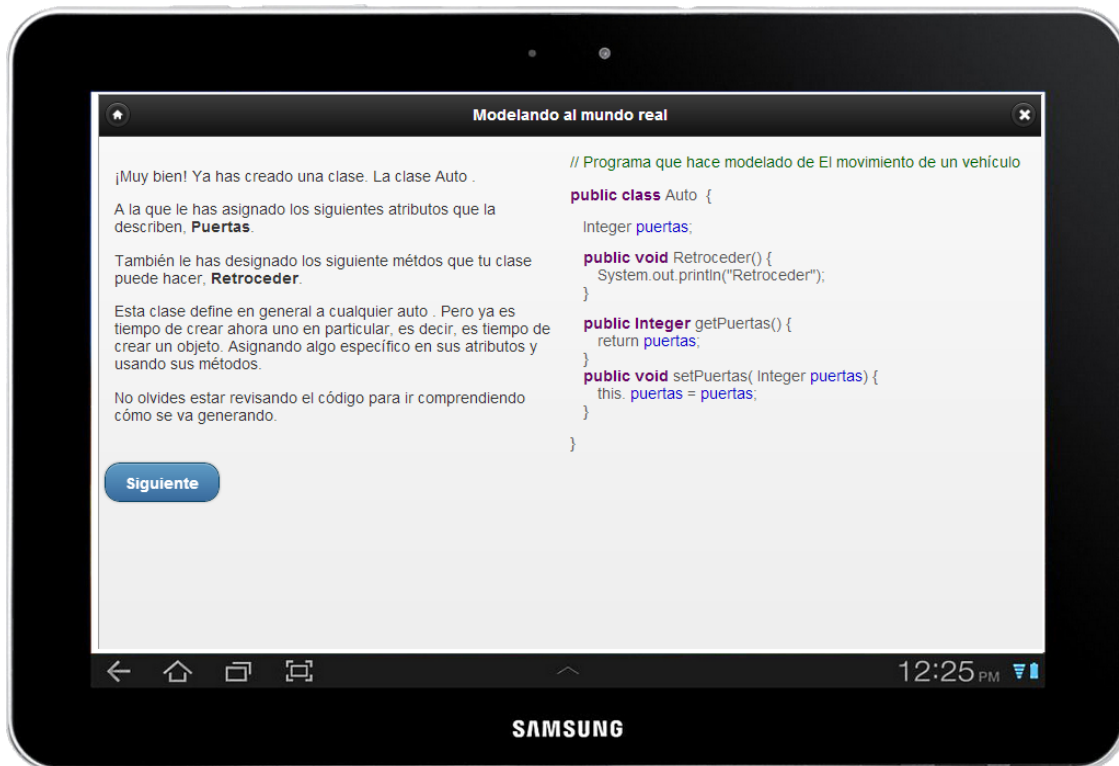


Figura 1.25. Código final de nivel principiante. Con colores para mejor comprensión.

La selección del nivel de domino para intermedio, puede ser desde la pantalla inicial, donde seleccionas los niveles, o al finalizar el nivel principiante, puede continuar con el nivel intermedio. Si el estudiante selecciona desde la pantalla de niveles, automáticamente se selecciona el modelo de auto, si no, continua con la elegida desde el nivel principiante (Figura 1.26).



Figura 1.26. Pantalla inicial del nivel Intermedio

En este nivel, se crea un objeto con las clase ya definida anteriormente, únicamente el estudiante selecciona ejemplos de atributos y métodos que ya había seleccionado para esa clase anteriormente, la figura 1.27 muestra los ejemplos para atributos y métodos de la clase.

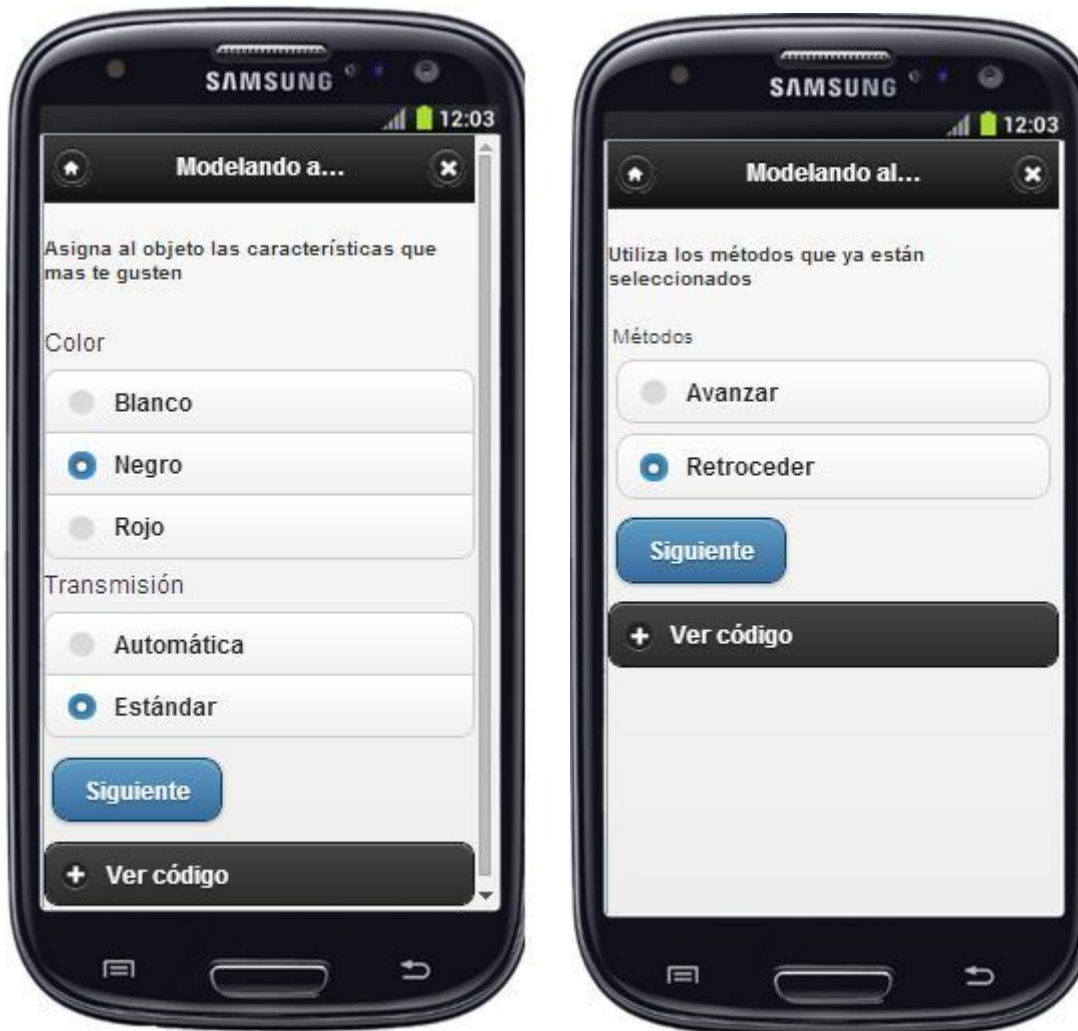


Figura 1.27 Selección de atributos y métodos para la clase.

Una vez seleccionados las características favoritas del estudiante, se le muestra el código final, de cómo es que puede asignarle valores a los atributos en código Java, también cómo puede mandar llamar los métodos y así generar un modelado final de cómo puede crearse un objeto a partir de una clase y cómo utilizar tanto sus atributos como métodos. En cada selección de ejemplos, se va mostrando en directo cómo se genera el código para cada pantalla respectivamente. El código final que se muestra en la pantalla de PAEPOO puede verse en la figura 1.28, donde se le indica al estudiante que el nivel intermedio a finalizado y ahora sabe cómo crear un objeto. También el código final donde cada palabra reservada del código, se le define si toca la palabra.



Figura 1.28 Código final de la creación de un objeto.

La selección del nivel de domino para avanzado, puede ser desde la pantalla inicial, donde seleccionas los niveles, o al finalizar el nivel intermedio, puede continuar con el nivel avanzado. Si el estudiante selecciona desde la pantalla de niveles, automáticamente se selecciona el modelo de auto, si no, continua con la elegida desde el nivel intermedio. Lo primero que se le pregunta al estudiante es qué entidad quisiere utilizar para realizar la herencia (Figura 1.29).



Figura 1.29 Selección de entidad para la herencia.

Al igual que en la creación de la clase, se le va preguntando al estudiante sobre aquellos atributos y métodos que desea que tenga su nueva clase, hija de la clase padre. Se le explica que al momento de crear esta subclase, tendrá los mismo atributos y métodos que tiene su clase padre, y además, tendrá los nuevos métodos y atributos que esté seleccionando. Una vez terminada la selección se le mostrará al estudiante el código final de la aplicación, donde se muestra ambas clases y sus objetos creados de la misma. Esto se ve en la figura 1.30.



Figura 1.30 Código final de la aplicación.

3.3 ErrorFinder: biblioteca de errores para el lenguaje Java

ErrorFinder es una biblioteca de errores que pretende ofrecer al estudiante soluciones para corregir aquellos errores que su compilador les dé al momento de estar programando. La aplicación muestra una lista pequeña de errores comunes, pero tiene la posibilidad de que los mismos estudiantes envíen los errores y posteriormente puedan ofrecerles soluciones.

La figura 1.31 muestra la vista principal de la aplicación donde el estudiante tiene un área para teclear las palabras que su compilador muestre en error.



Figura 1.31. ErrorFinder, pantalla inicial para comenzar la búsqueda o visualizar la lista de errores.

Una vez que el estudiante haya hecho la búsqueda, la aplicación le devolverá un listado con aquellos errores que tengan las palabras que el estudiante haya ingresado, mostrando no únicamente las que tengan las palabras en el orden escrito, si no también aquellas que tiene en algún lugar del texto todas las palabras en común. Esto es para facilitar al estudiante la búsqueda y pueda encontrar fácilmente el error que está buscando exactamente (Figura 1.32).

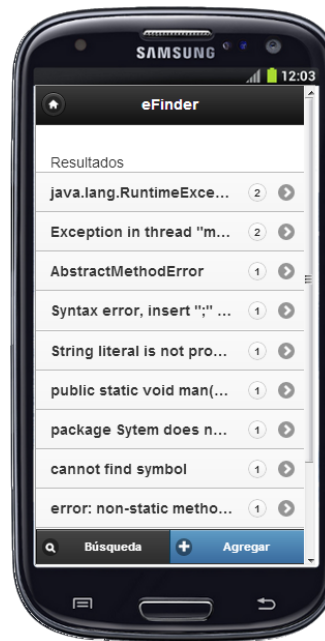


Figura 1.32. Lista predeterminada de errores.

Esta lista muestra el estudiante el error tal cual lo escribe el compilador, ya sean errores de ejecución, de compilación o de sintaxis, todos se muestran aquí, y en cada opción viene también la cantidad de posibles soluciones para poder arreglar ese problema. En todo momento de la aplicación se le ofrece al estudiante la posibilidad de agregar su propio error si es que no aparece en la lista de errores, de esta forma, ErrorFinder crecerá con ayuda de la participación de los estudiantes, para poder cubrir todos y cada uno de los tipos de errores posibles que puede lanzar el compilador a la hora de programar.

Una vez encontrado y seleccionado el error, la aplicación le mostrará en español lo que ese error significa para que el estudiante pueda comprender un poco más qué es su problema. Además, se le explica por qué es generado este error y cuáles son las posibles causas que hayan generado su aparición. Todo es en español para que los estudiantes no tengan ninguna limitación al momento de comprender su origen. La figura 1.33 muestra la pantalla que contiene el error tal cual lo lanza el compilador, su traducción en español y su descripción de cómo es posible que se genere ese error.



Figura 1.33. Traducción y explicación del error seleccionado.

Después de esta pantalla el estudiante tiene la opción de revisar las posibles soluciones que podría ayudar con su problema. Como se muestra en la figura 1.34, el estudiante tiene un listado de posibles soluciones que pudiera corregir el error que les genera. Estas soluciones son proporcionadas por el profesor pero se espera que los estudiantes también puedan agregar sus propias soluciones, así para poder enriquecer cada vez más la aplicación.

Las soluciones pueden ser “puntuadas” cuando ésta les ayudó a corregir su problema, simplemente clicando sobre ella. De esta forma, la próxima vez que vuelvan a revisar ese error, esta solución será la primera en la lista demostrando que es la opción que sí ha resuelto la mayoría de los casos para otros estudiantes; así, se logra a la vez rapidez para solucionar el problema.

También los errores más comunes son los primeros en la lista si es que el alumno ingresa a su descripción, se va sumando a la cantidad global de visitas del error.



Figura 1.34. Lista de soluciones posibles para ese error.

El propósito principal de ErrorFinder es que el alumno no se detenga al momento de programar, que no existan errores que eviten siga programando o pierda el interés por no poder solucionar su problema. ErrorFinder pretende tener la mayor cantidad de errores y soluciones que cualquier excepción lanzada por el compilador pueda generar. El problema del lenguaje para los estudiantes es una limitante cuando están compilando código, pues todas esos errores son mostrados en inglés; ErrorFinder es en español, una vez ingresado el error, es traducido, explicado de porqué se genera y se proporcionan soluciones amigables de cómo corregir el problema.

Todas las aplicaciones que pertenecen a PAEPOO están en constante cambio, retroalimentándose de las opiniones que ofrecen los estudiantes, así como de colegas de la universidad, profesores y amigos.

PAEPOO se ha desarrollado en versión móvil para los navegadores web, tanto de teléfonos como computadoras de escritorio, y una versión media para el sistema operativo Android.

Para Android se desarrolló:

- CodeFactory
- ErrorFinder

Para su versión web:

- CodeFactory
- Definer
- ErrorFinder

Aunque únicamente se esté probando la versión web en el grupo de evaluación.

PRUEBAS Y RESULTADOS

4.1 Pruebas realizadas

La plataforma se probó en un grupo de Programación Orientada a Objetos de tercer semestre. Es la primer materia que llevan con orientación a objetos. El grupo consta de 15 estudiantes, de los cuales todos se registraron en la plataforma para poder utilizar las aplicaciones.

Antes de empezar a utilizar PAEPOO se les hizo una encuesta para saber sus conocimientos acerca de programación orientada a objetos, conocer qué tipo de dispositivos móviles utilizan y qué tan interesados están en aprender a programar. Los resultados se muestran en la figura X.X, ahí se puede apreciar cómo la mayoría de los estudiantes están ansiosos por aprender y pocos que ya saben, la investigación se enfoca en aquellos estudiantes que de alguna forma creen que se les complicará un poco, donde, como se observa, es un 37.5% del total. También se muestra cómo es que la mayoría de los estudiantes tienen un conocimiento de programación orientada a objetos por debajo de nivel intermedio, lo que demuestra que no han tenido anteriormente clases de orientación a objetos. En adición, casi el total de los estudiantes poseen un Smartphone en cual podrán utilizar la plataforma y una cuarta parte, también posee un Tablet.

Sabiendo esto se inició con las pruebas, donde al inicio del curso se les explicó detalladamente la plataforma, el cómo usarla y que podían acceder a ella incluso fuera de clase. No se les obligaba a usarla ni se les recordaba, porque otro objetivo de esta investigación es que los estudiantes se interesen por sí solos en la plataforma y en que quieran aprender por ellos mismos con las herramientas que se les facilitan.

El grupo está definido dentro de las siguientes estadísticas:

- La edad promedio es de 21 años.
- La mayoría son hombres (sólo dos mujeres).

Para hacer más limpia esta investigación, no se les dijo a los estudiantes que todos los movimientos que realicen dentro de PAPEOO serán registrados. Tampoco se les forzará a utilizar la aplicación, únicamente, cada semana se les recordará que pueden utilizarla cuando quieran.

4.2 Análisis de resultados

Todas las acciones que los estudiantes realizaban dentro de PAEPOO fueron registradas y estuvieron siempre bajo un control. Por lo que todos los resultados fueron generados con el simple uso de la plataforma. Otros resultados fueron creados por medio de encuestas antes, durante y después de la utilización de PAEPOO. Lo que se pretende con este registro es contabilizar la cantidad de veces que utilizan las aplicaciones de la plataforma, basados en dos parámetros:

1) el primero identifica la cantidad de veces que el estudiante ingresó a dicha aplicación. Los resultados fueron los siguientes:

- a. CodeFactory : 42%
- b. ErrorFinder : 26%
- c. Definer : 32%

Indicando que la aplicación que más llegaron a utilizar fue CodeFactory.

2) El segundo identifica la cantidad de tiempo que dura utilizando una aplicación. Los resultados fueron los siguientes:

- a. CodeFactory :39%
- b. ErrorFinder :16%
- c. Definer :21%

Indicando nuevamente que CodeFactory es la aplicación en la que más permanecen los estudiantes.

Con estas estadísticas se pretende saber cuál de las aplicaciones utilizan más, sabiendo esto, podemos identificar el mayor problema de los estudiantes:

- CodeFactory: Se les complica plasmar el mundo real en código orientado a objetos.
- Definer: Se les complica comprender las bases de programación Orientada a Objetos.

- ErrorFinder: Se les complica escribir código, específicamente en Java, al momento de trabajar en sus prácticas, la sintaxis y semántica pueden ser problemas técnicos individualmente de cada estudiante.

CodeFactory es la aplicación a la que más recurren los estudiantes, tanto por veces de utilización, como cantidad de tiempo invertida en ella. Esta aplicación es la base principal de la plataforma, pues es aquí donde los estudiantes empiezan a ver los primeros intentos de modelaje del mundo real en código de Java para programación orientada a objetos. La aplicación enriquecida con amplios ejemplos, les ayuda a sentirse cómodos con el modelado más apropiado para ellos, con el que se identifiquen, aquel que les satisfaga o simplemente que les guste o estén muy relacionados.

Ambas estadísticas se reflejan en la figura 1.35, donde se muestra que CodeFactory obtiene la mayor cantidad del diagrama de pastel, en ambos tópicos seleccionados.

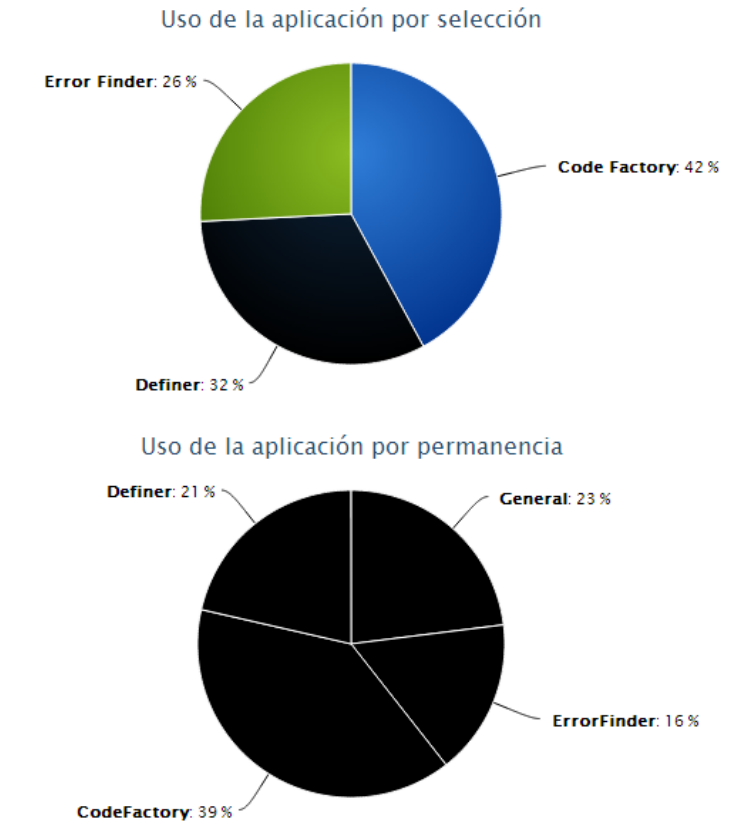


Figura 1.35. Porcentaje del uso de las aplicaciones.

CodeFactory contiene tres niveles de dominio, aquellos que están un nivel inicial, principiante; los intermedios y los avanzados, donde se tratan temas un poco más complejos de Programación Orientada a Objetos (Figura 1.36).

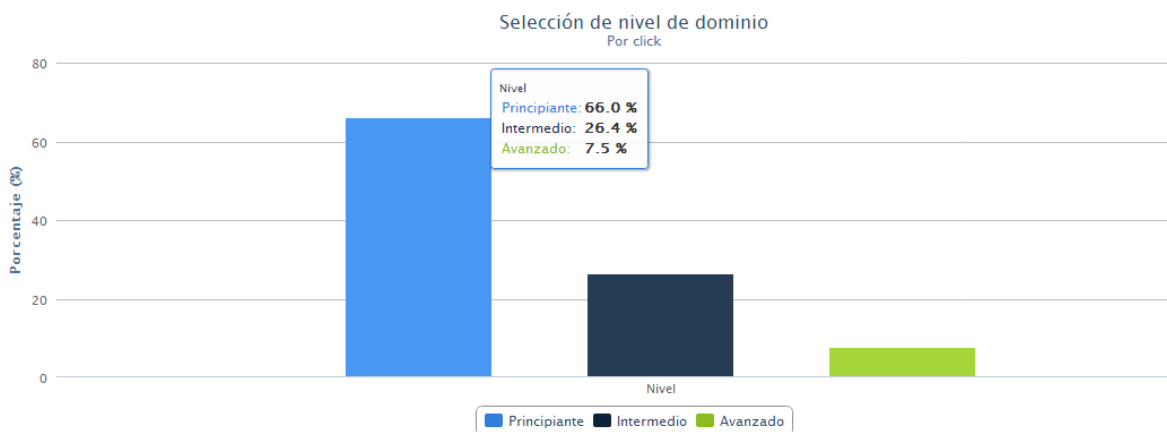


Figura 1.36. Niveles de dominio de los usuarios en CodeFactory.

El nivel principiante fue el que más estuvo presente, pues la clase de prueba eran estudiantes de primer grado de la universidad, en tronco común, en algunos casos, jamás habían tenido contacto con programación, o en su defecto, programación orientada a objetos.

Los modelajes utilizados fueron los ejemplos más comunes en la mayoría de los cursos de programación orientada a objetos, modelado del movimiento de un auto, el reino animal, y uno adicional que habla sobre cuerpos celestes.

La figura 1.37 muestra que el modelado preferido por los estudiantes es el movimiento de un auto, debido a su facilidad de comprensión en todos los temas, mientras que el reino animal y los cuerpos celestes tuvieron muy poco impacto en los estudiantes.

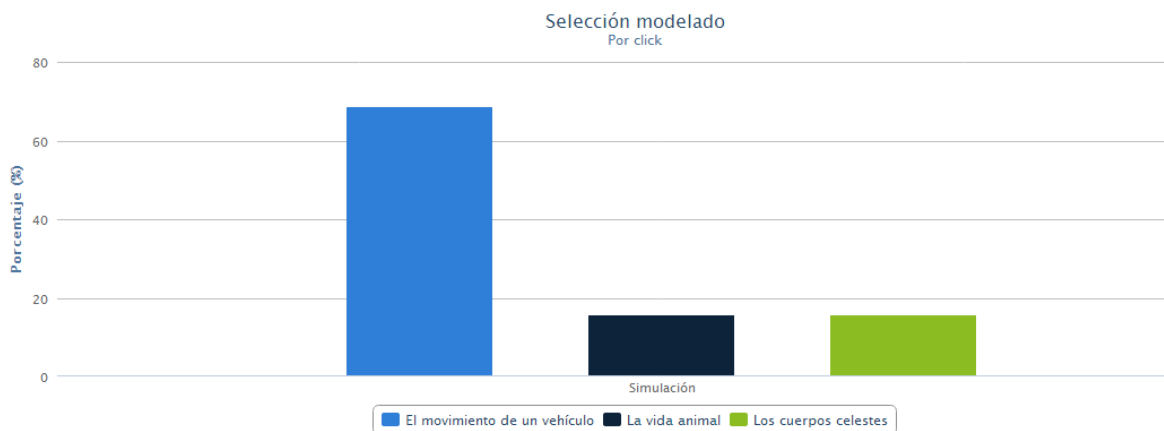


Figura 1.37. Modelados más utilizados por los usuarios.

Una vez elegido el modelado, se creaban las clases que serían utilizadas a lo largo de toda la aplicación, para poder comprender clases, objetos, atributos, herencia y polimorfismo. La clase que predominó entre los estudiantes fue la clase auto, que pertenece al modelado de movimiento de un vehículo, el resto, estuvo muy por debajo del porcentaje.

La figura 1.38 muestra la gran diferencia que existió entre las entidades utilizadas. Al ser algo tan cotidiano, un auto, fue la elección de la mayoría, al tener

aquellos atributos y métodos con que uno está familiarizado, que se han visto desde que son niños. Es una forma muy sencilla de introducir lo que uno ya conoce al mundo de la programación.

Para la aplicación Definer, el tema más visitado fue polimorfismo y encapsulación, que son de los últimos temas que se ven en los primeros cursos de programación orientada a objetos, seguido de herencia y método, la figura 1.39 muestra una mayor visión de los temas más vistos por los estudiantes.

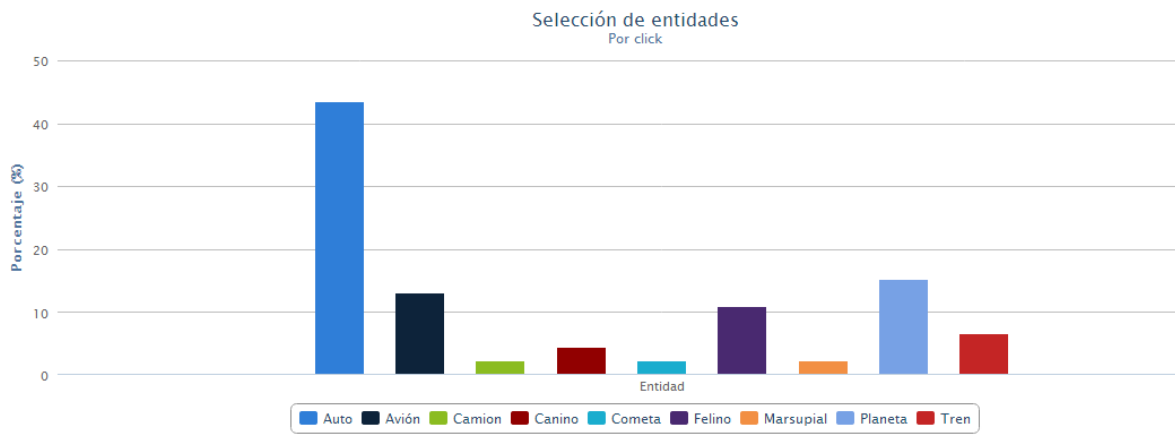


Figura 1.38. Clases más utilizadas por los usuarios.

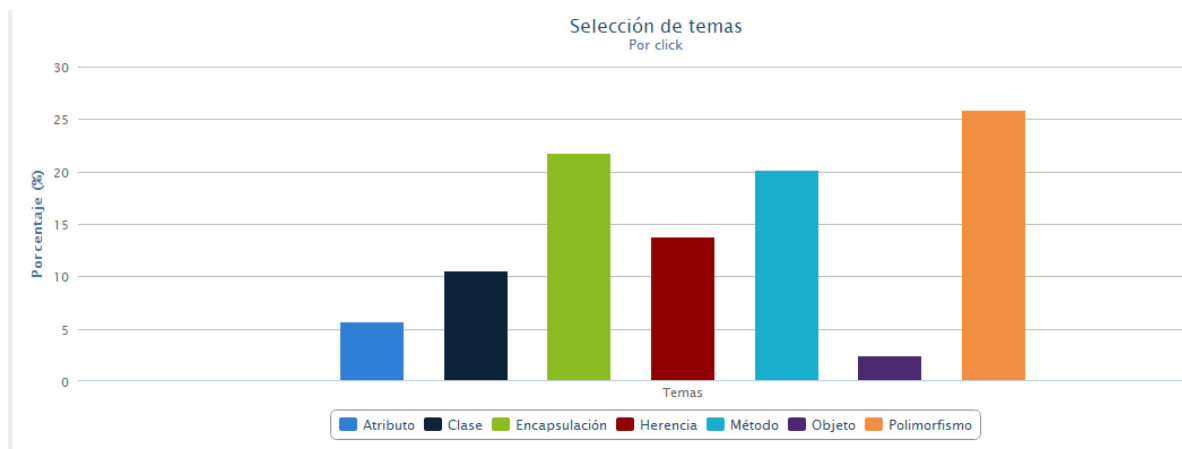


Figura 1.39. Temas con más consultas en Definer.

Polimorfismo es el tema que predominó durante toda la aplicación, este tema fue introducido un mes después de haber iniciado las pruebas en el curso, a

pesar de eso, fue el más visitado con la desventaja de antigüedad ante el resto de los temas.

La aplicación Definer contiene dos características a utilizar, las definiciones amigables y los ejemplos cotidianos. La figura 1.40 muestra la cantidad de visitas que tuvo cada uno, mostrando que la sección de definiciones fue la dominante, pocas fueron las revisiones de ejemplos que tuvo Definer durante el curso. La múltiple selección de definiciones de distintas fuentes ayudó a los estudiantes a elegir más esta sección que la de ejemplos

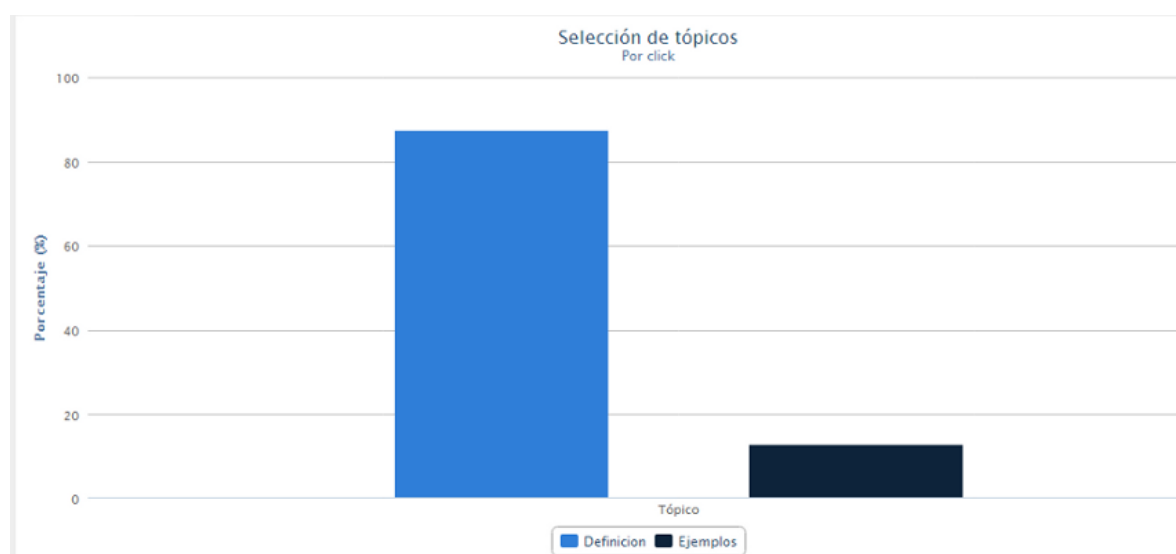


Figura 1.40. Características más utilizadas en Definer.

La utilización de la plataforma no fue obligatoria para los estudiantes, al inicio del curso se les mostró que tendrían siempre disponible (incluso fuera de horario de clases) la plataforma con las mencionadas características para que la pudieran utilizar durante el curso, se les explicó que es un proceso de prueba y que crecería regularmente con sus intervenciones y aportaciones. Se les recordaba únicamente ciertos viernes a la semana de que usaran la aplicación, en especial si existía un bajo uso de la misma. La figura 1.41 muestra lo que fue la utilización de PAEPOO durante un mes completo, en el ejemplo se muestra el mes de marzo, el mes más visitado para la plataforma. En esta figura se muestra los días y horarios que más utilizaban la plataforma. Este mes inicia en 4 y termina en

30, en las barras verdes muestra los días y hora en que les toca la clase de programación, son dos días a la semana. En las barras azules se muestran las horas en que cualquier estudiante ingresa a la aplicación. Sólo hay una franja amarilla en el calendario, que es el día de aplicación del examen.

Esta figura nos explica que independientemente el día que sea, acceden a la plataforma para consulta o práctica, aunque sí existe un aumento leve de utilización alrededor de la hora de clase, específicamente de laboratorio de cómputo. También se muestra cómo horas antes del examen revisaron por última vez sus anotaciones y ejemplos y definiciones de la plataforma. Las aplicaciones de PAEPOO no se cerraron durante la aplicación de ningún examen, sin embargo, ningún estudiante la utilizó en este tiempo. Sábados y domingos, como se muestra en la figura, fue similar en el resto de los meses, no hubo mucha actividad, más que esporádicas y que duraban sólo unos minutos. La mayoría de los inicios de sesión se generaban en horarios de clase y por las noches alrededor de las 22:00. El mes con menos actividad fue mayo, que a pesar de ser de los últimos del semestre donde más trabajo tiene los estudiantes, fue donde hubo menos actividad. La plataforma es para principiantes, para los primeros temas, una vez aprendido o entendido esas definiciones o temas básicos, la aplicación pasa a ser obsoleta.

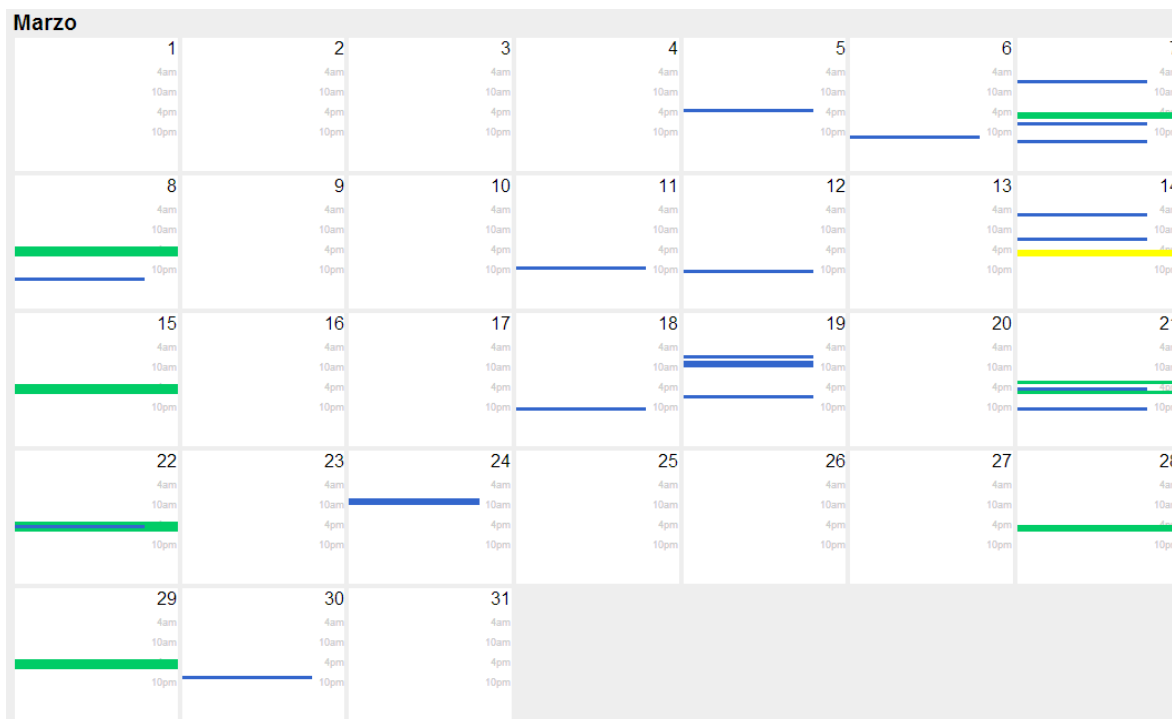
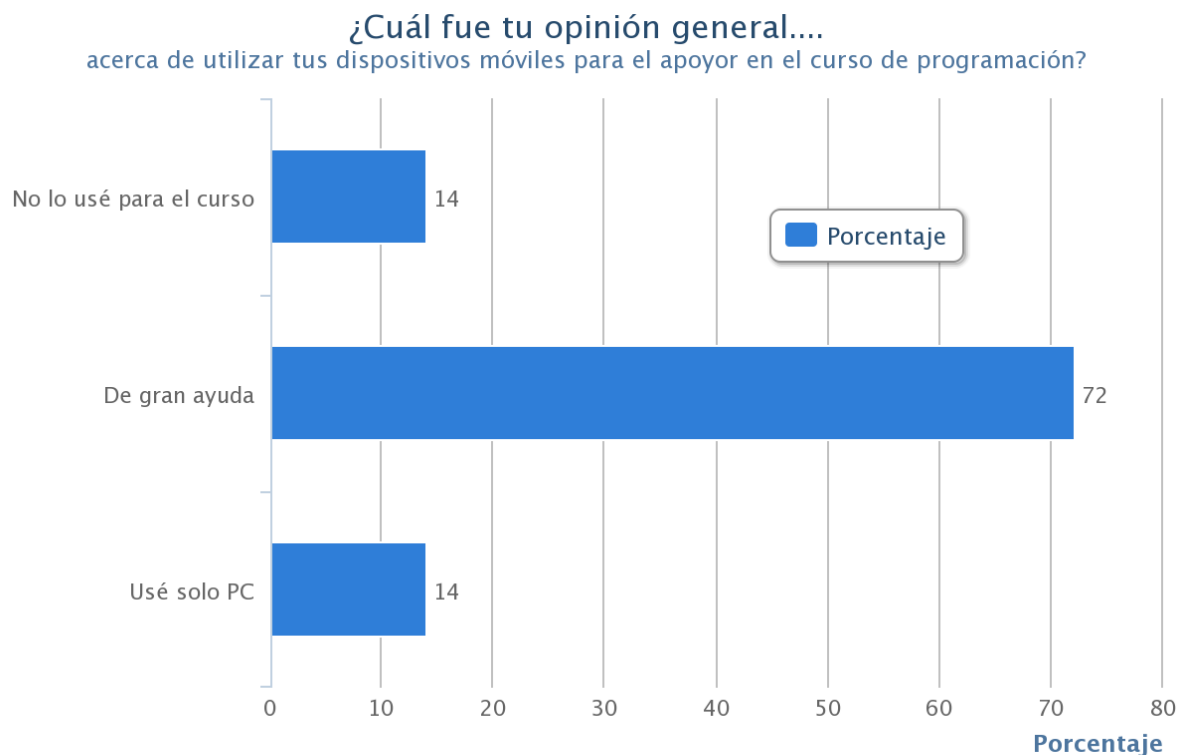


Figura 1.41. Utilización de PAEPOO con horas y días durante el mes de marzo. Franja azul: uso de Paepoo. Franja Verde: Clase de programación. Franja Amarilla: Examen.

Una vez finalizado el curso, cuando los estudiantes dejaron de utilizar la plataforma se les realizó una encuesta para conocer su opinión acerca de su experiencia durante el semestre de cómo utilizaron PAEPOO, sus resultados, impresiones y sugerencias o críticas de cualquier tipo que pudieran tener. La encuesta fue realizada en papel y todas las preguntas fueron abiertas para que los estudiantes plasmaran su idea tal cual la tenían pensada. Los resultados, en su gran mayoría fueron satisfactorios. Las respuestas se intentaron colocar en un tópico general para utilizar un solo parámetro de evaluación.

La figura 1.42 muestra la opinión general de los estudiantes acerca de utilizar sus dispositivos móviles como apoyo para el curso de programación orientada a objetos. La respuesta global que predomina fue que les sirvió de mucha ayuda, estadísticamente bajo aquellos que no la utilizaron para nada o que no tenían ningún dispositivo móvil.



1.42. Opinión general de utilización de dispositivos móviles como apoyo en un curso de programación.

Para intentar comprender las necesidades de los estudiantes se les preguntó qué beneficios o desventajas notaban mientras utilizaban la plataforma, intentar conocer aquellas cosas que les hacían falta a ellos en especial, o eso que les gustó mucho y que no quisieran cambiarlo. Las respuestas se englobaron de la siguiente forma: ayuda a entender conceptos y ejemplos, ayuda a encontrar tus errores, necesita más ejemplos; y una opción neutral que les fue indiferente para los estudiantes o no tenían opinión. Estas respuestas están graficadas en la figura 1.43. La mayoría de las respuestas tenían sus necesidades muy específicas, pero se intentó colocar dentro de un parámetro cada una para poder evaluar el rendimiento de la plataforma y su contenido. Para poder mejorar a grandes rasgos intentando cubrir a la mayoría de los estudiantes. Las opiniones fueron totalmente abiertas son restricciones algunas.

¿Qué beneficios o desventajas notaste al utilizar PAEPOO durante el curso?

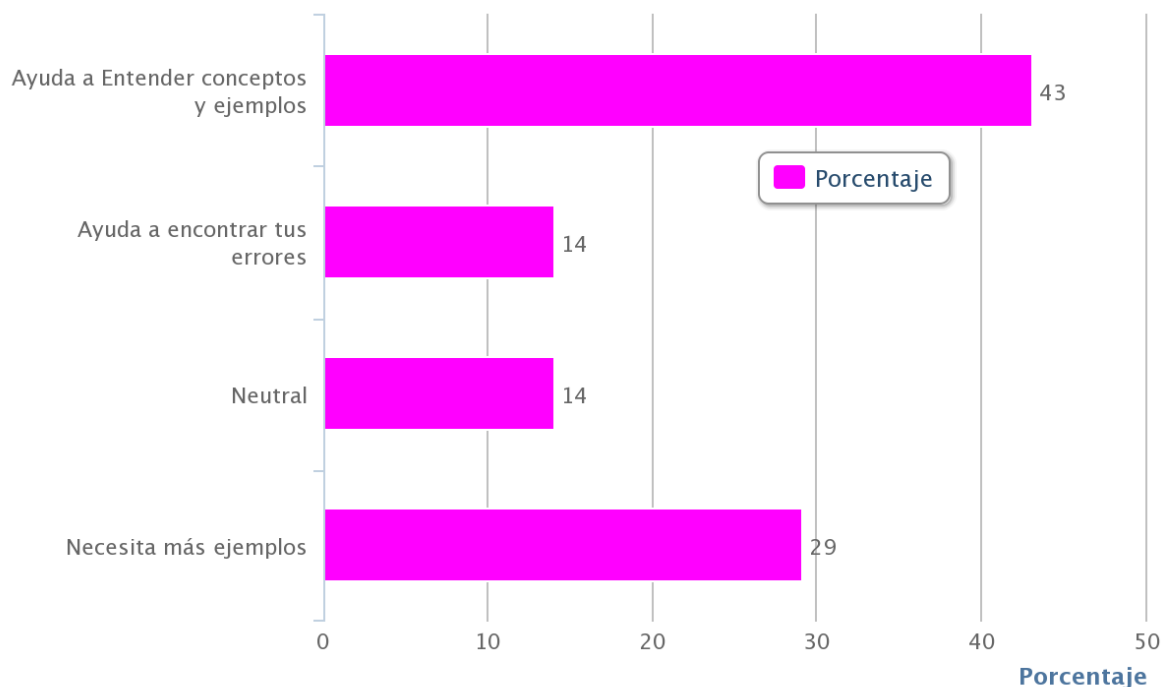


Figura 1.43. Opiniones generales sobre ventajas o desventajas encontradas por los estudiantes durante la utilización de la plataforma.

A pesar de registrar todo acceso a la aplicación, se les preguntó si accedían a PAEPOO fuera de clase, como control de calidad, intentando crear respuestas más honestas. La figura 1.44 muestra esta estadística.

De igual forma, se les preguntó si habían utilizado PAEPOO durante un examen. La respuesta fue negativa, dejando en cero por ciento la opción “Sí” (Figura 1.45).

Tal y como se demuestra en el calendario y horario de uso de la plataforma, ningún estudiante utilizó la plataforma durante ningún examen, tal y como respondieron esta pregunta. De forma similar, se muestra en el calendario que utilizaron en repetidas ocasiones la plataforma en horarios fuera de clase, incluyendo sábados y domingos.

Las estadísticas de PAEPOO que fueron registradas y las respuestas de los estudiantes por medio de esta encuesta concuerdan, por lo que se asume que el resto de las respuestas, tienen un mayor porcentaje de probabilidad que sean ciertas y nos ayude a generar los resultados para este trabajo de tesis.

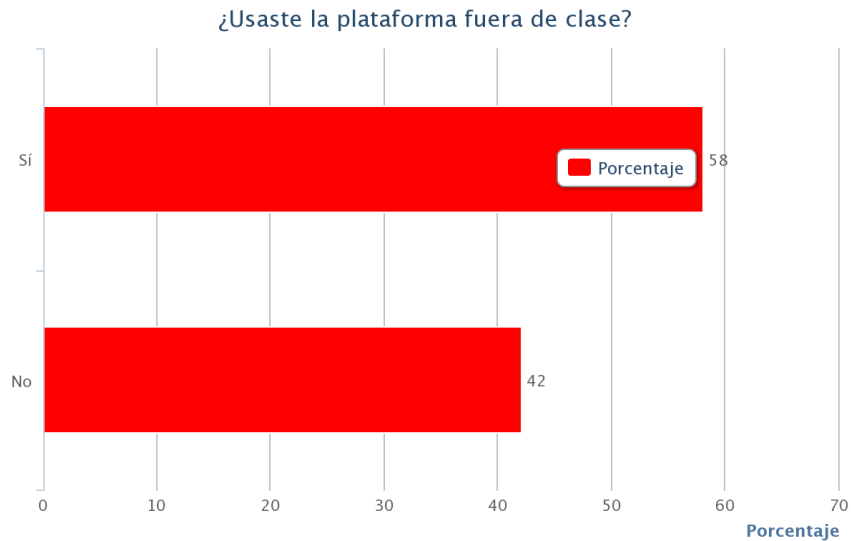


Figura 1.44. Utilización de PAEPOO fuera de clase.

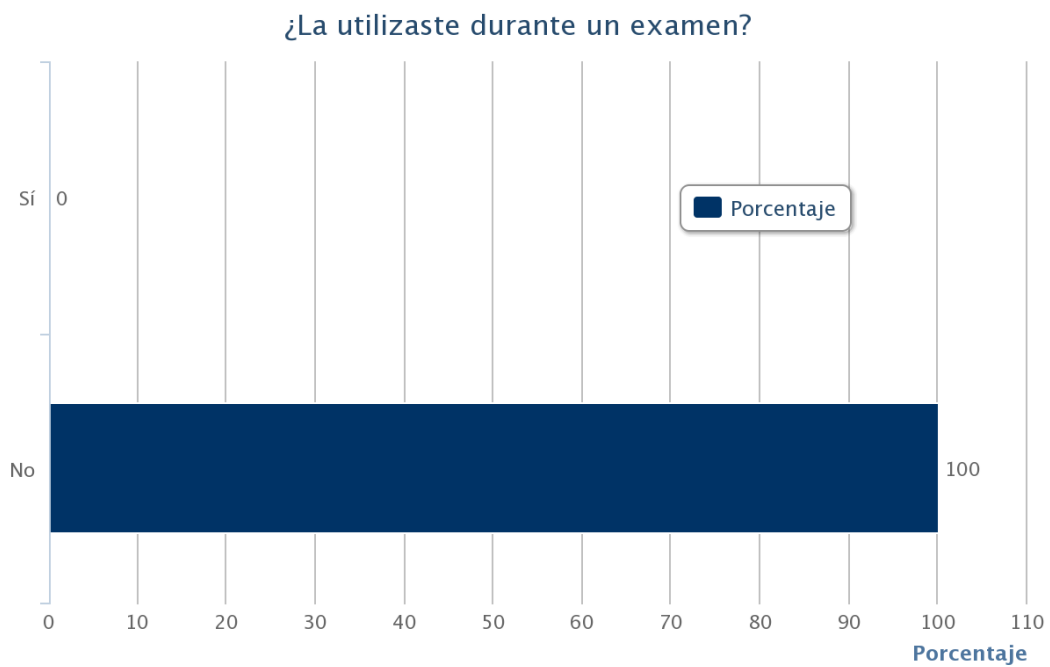
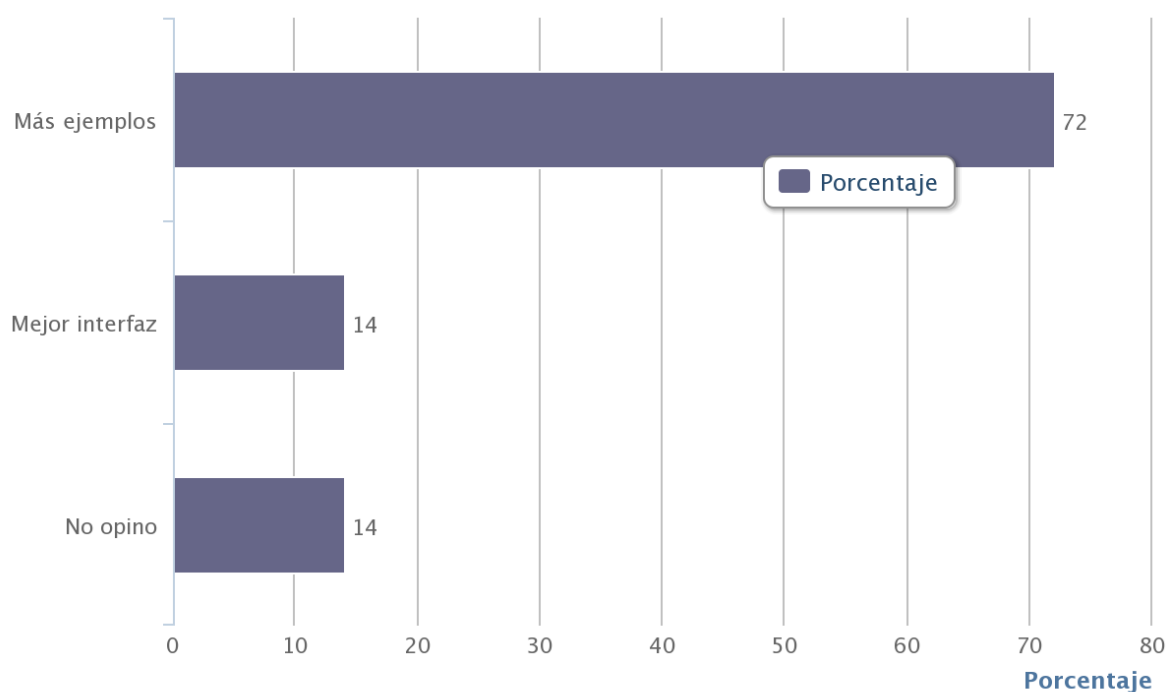


Figura 1.45. Utilización de PAEPOO durante un examen.

Se pretendía conocer directamente aquello que los estudiantes sintieran que le hacía falta a PAEPOO, de forma global, la figura 1.46 muestra las respuestas. Lo que les gustaría que tuviera para que la plataforma fuera más de su agrado. En su mayoría respondieron que tuviera más ejemplos, para poder basarse en su código. Porcentajes muy bajos fueron abstemios y sobre mejorar la interfaz de las aplicaciones, hacerla más amigable, más colorida.

¿Qué te gustaría que tuviese la plataforma para que fuera más de tu agrado?



1.46. Opinión de los estudiantes para sugerir aquello que quisiera que PAEPOO tuviera y que sea un poco más de su agrado.

Para asegurarnos que los estudiantes no accedían por gusto a la aplicación se les preguntó si siempre recordaban su contraseña. Todas las respuestas afirmativas mostraron que aquellos que dejaron de acceder a la plataforma lo hacían por gusto, independientemente sin razón de haber aprendido, aburrido o cualquier otra. La intención es asegurarse que no fueron problemas técnicos la

razón de dejar de utilizar la PAEPOO. Puesto que se les recordaba en ocasiones durante las horas de laboratorio de cómputo que tenían a su disposición en cualquier momento la aplicación para repasar o practicar lo visto en clase. La figura 1.47 muestra lo descrito anteriormente.

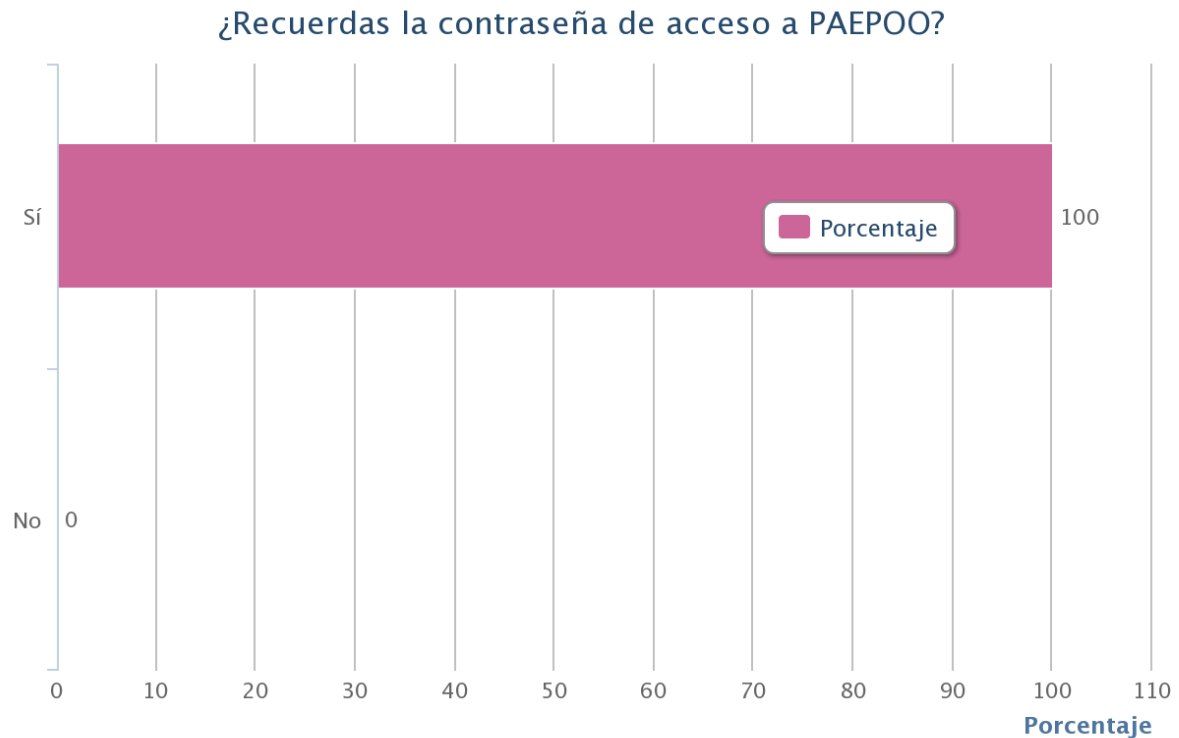


Figura 1.47. Estadísticas sobre aquellos estudiantes que llegaron a olvidar su contraseña.

Como detalle adicional, era interesante conocer si la aplicación PAEPOO les sería útil en otras asignaturas de su curso, esto ayudaba a saber si en realidad les había ayudado en su curso de programación y que de ser así lo quisieran también aplicar en alguna otra materia que se les complicara (Figura 1.48).

Las asignaturas que figuraron dentro de la escuela fue la siguiente:

- Circuitos digitales con un 43%.
- Electrónica con un 28%
- Métodos con un 14%
- Algoritmos con un 14%

- El resto de los estudiantes les fue indiferente o la preferían en el resto con un 28%

De acuerdo a este listado, las asignaturas en las que necesiten la utilización de sus dispositivos móviles están relacionadas con el área de electrónica.

¿En qué otras materias te gustaría utilizar una herramienta como PAEPOO?

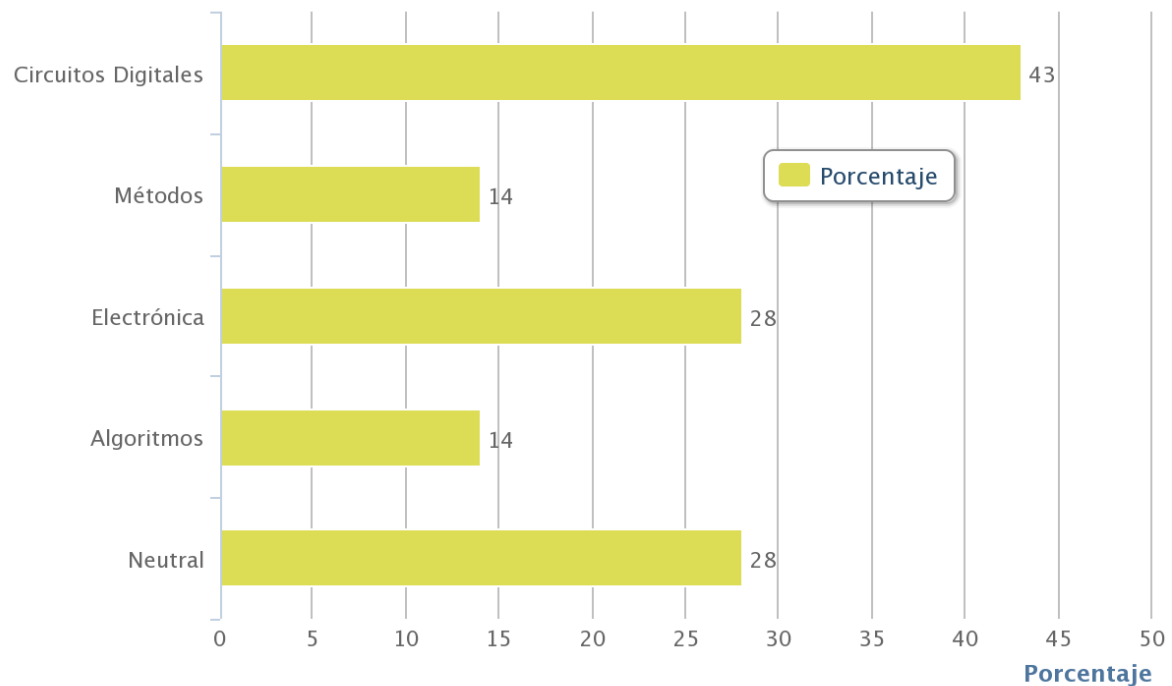


Figura 1.48. Estadísticas sobre utilizar PAEPOO para otras asignaturas.

Para conocer en sí qué tan útil fue el contenido de las aplicaciones para los estudiantes, hablando sobre definiciones, ejemplos, ejercicios, actividades, se les preguntó directamente si todo eso, para ellos fue útil, para qué les sirvió, el cómo es que llegó a tener un impacto sobre su aprendizaje. De forma global la mayoría de los estudiantes opinaron que todas estas características les ayudaron a entender mejor los conceptos básicos de la programación orientada a objetos, gracias a las definiciones de Definer y la interacción con el código de CodeFactory les facilitó llegar a comprender un poco mejor lo que se ve en clase. Otros

simplemente dijeron que ha sido útil, en general, o que ha sido de un poco de ayuda. Y otros muy pocos gracias a los ejemplos comprendieron y el resto en su minoría prefirió no opinar. (Figura 1.49).

Con estos resultados, es la tercera ocasión que los estudiantes hacen énfasis en los ejemplos.

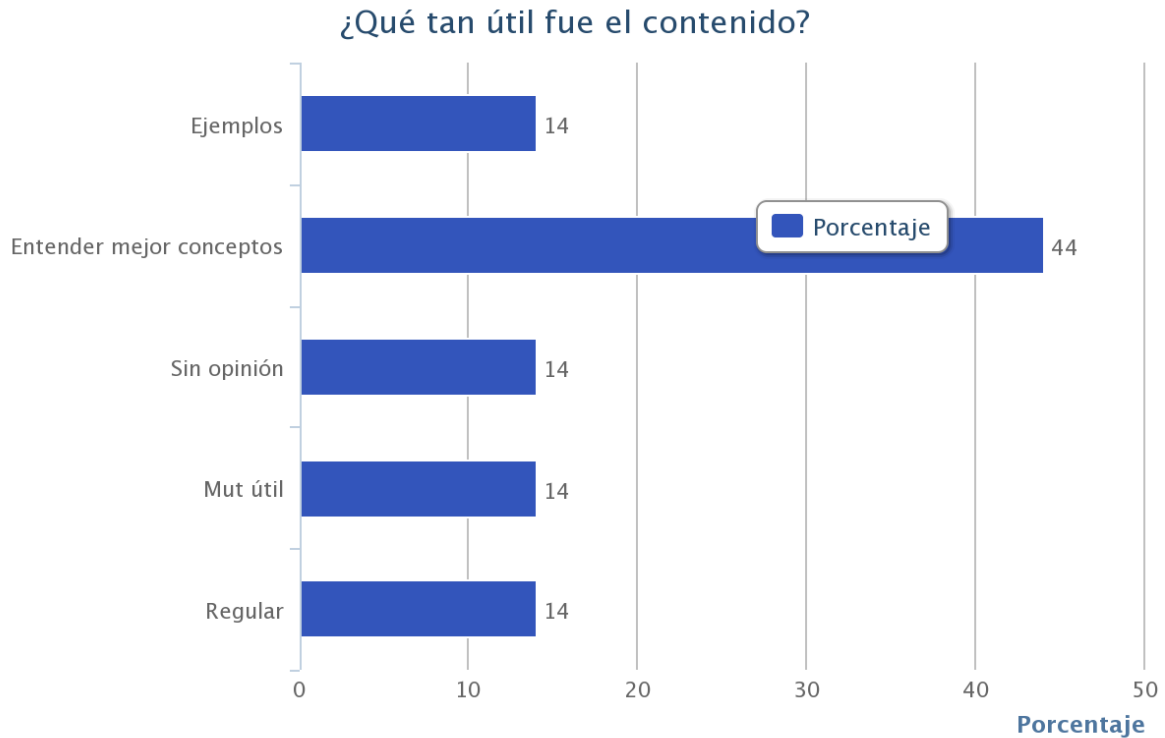


Figura 1.49. Opinión de los estudiantes sobre qué tan útil para ellos haber utilizado PAEPOO.

Además de ser una plataforma de apoyo que únicamente sirva de referencia y consulta, se les preguntó a los estudiantes si estarían de acuerdo en utilizar PAEPOO un poco más lejos, es decir, realizar tareas y prácticas desde su teléfono, incluso exámenes de sus dispositivos móviles o preferían seguir utilizando la computadora de escritorio o el portátil para realizar estas tareas. La figura 1.50 muestra que la mayoría de los estudiantes aún siguen prefiriendo seguir trabajando en sus computadoras, ya que codificar en los dispositivos móviles será demasiado complicado, además del software empleado para depurar, compilar y ejecutar código sólo se encuentra disponible para computadoras de escritorio o portátiles, mientras que para móviles son muy

escasas y las características ofrecidas no son lo suficientemente ricas para apoyar a los estudiantes con sus necesidades.

¿Te hubiera gustado que pudieras realizar exámenes, tareas o prácticas...
... desde tu teléfono o prefieres seguir utilizando la PC/Laptop para hacerlo?

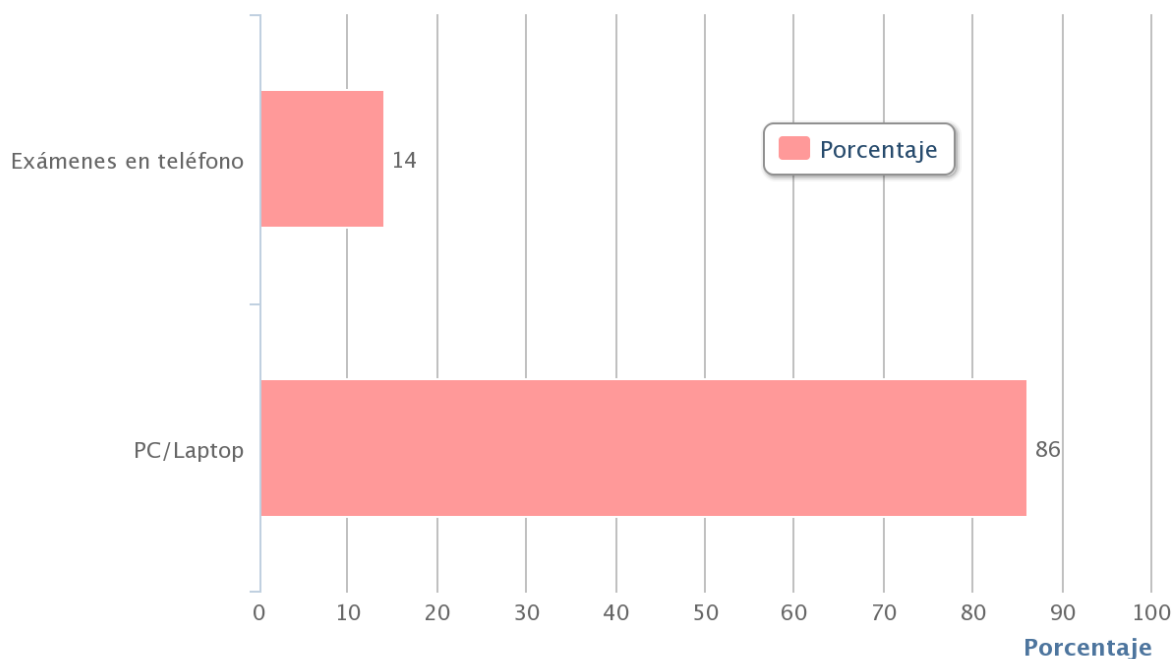


Figura 1.50. Opinión de los estudiantes sobre utilizar PAEPOO con otras utilidades.

Finalmente para conocer las preferencias de los estudiantes sobre tener cada vez más cosas que se muden a lo digital, se les preguntó si preferían responder las encuestas digitalmente o de la forma tradicional en papel. La figura 1.51 demuestra que la mayoría de los estudiantes prefieren responder las encuestas de forma digital, pues para ellos es más fácil responder desde la comodidad de su teléfono inteligente o su tableta que tomar un papel, lápiz y comenzar a contestar las preguntas.

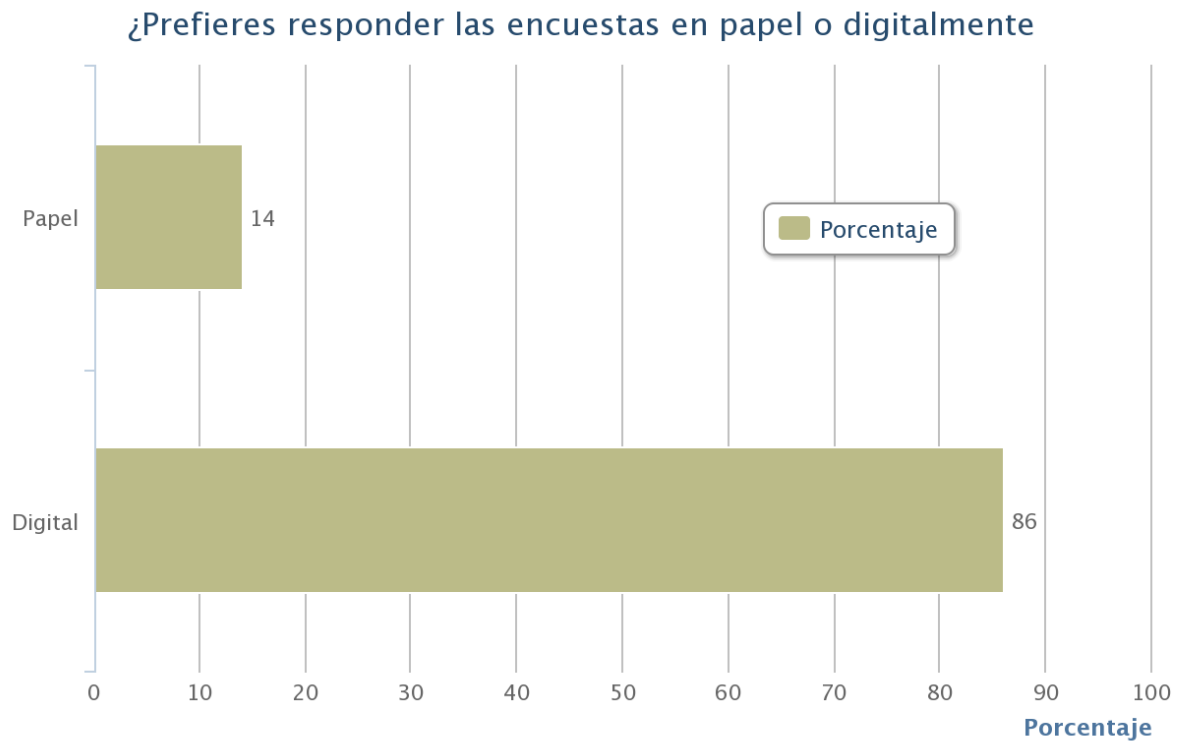


Figura 1.51. Opinión de los estudiantes sobre obtener encuestas en papel o digital

CONCLUSIONES Y TRABAJO FUTURO

5.1 Conclusiones

La dificultad en el aprendizaje de la programación orientada a objetos no sólo recae en la dificultad del curso y la planeación del mismo, un profesor debe hacer su papel de motivador/profesor, atender a todas las necesidades personales de los estudiantes, abarcar las distintas culturas y necesidades de cada uno de ellos. Es necesario que en grupos grandes en el curso de programación, los mismos estudiantes sean maestros y estudiantes, la colaboración ha demostrado ser una ayuda eficaz en el rendimiento del aprendizaje, en especial en personas que inician en el tema.

Para cubrir todas estas dificultades, hay que dejar que los mismos estudiantes encuentren la forma ideal para aprender ellos mismos, siendo guiados por el profesor que a la vez hace su papel de motivador; a través de las tecnologías móviles, creemos que podemos abarcarlas, casi el total de estudiantes tiene un teléfono celular, está familiarizado con él y puede utilizarlo en cualquier momento, debemos aprovechar esta ventaja para introducir el aprendizaje en todo momento; dejándolos con la libertad de hacerlo a su manera, con todos los elementos que les facilite el profesor.

El aprendizaje creemos debe estar en la cotidianidad de las cosas, el aprendizaje de la programación orientada a objetos debe facilitarse si deja de ser lineal y se vuelve dinámico.

Con este trabajo creemos que los estudiantes están muy entusiasmados en utilizar sus teléfonos móviles como apoyo en el aprendizaje de sus asignaturas, específicamente de programación orientada a objetos, aquellos estudiantes que no estaban interesados en aprender a programar tuvieron un fuerte interés en

utilizar la aplicación ya que podían acceder a ella desde su teléfonos. La plataforma fue muy útil, pues ayudó a los estudiantes a comprender los conceptos básicos de programación orientada a objetos.

Creemos que las tendencias van dirigidas a volver digitales las clases poco a poco en la universidad, gracias a los dispositivos móviles. Los estudiantes ponen de su parte para colaborar y mejorar las clases en las universidades para tener un apoyo extra en sus cursos, y no sólo de programación orientada a objetos, pues ellos mismos sugieren que sean introducidas en otras asignaturas.

En tres preguntas de la encuesta que se llevó a cabo al finalizar las pruebas de la plataforma, gran porcentaje de los estudiantes hicieron mucha énfasis en que a las aplicaciones les hacían falta más ejemplos para poder comprender todos los términos que engloban a programación orientada a objetos. En la aplicación Definer, todas las definiciones contaban con ejemplos, unas más que otras, aun así los estudiantes querían más ejemplos en esta sección. También en la aplicación CodeFactory, los ejemplos eran más limitados, incluso llegaron a sugerir qué tipo de ejemplos les gustaría poder tener en PAEPOO.

La realización de un focus group para la obtención de una encuesta con mayor número elevado de porcentaje acertado, que realmente se necesitan en nuestro estudio, sí estuvo presente al menos en ideas, sin embargo, por la población pequeña que se tenía, alrededor de 15 estudiantes, no se pensó que fuera necesario la realización del focus group.

La encuesta fue realizada en torno a las inquietudes que satisfacción las necesidades de la investigación. Por tal motivo las preguntas fueron totalmente abiertas y así dejar que se expresaran tal cual se sentían, sin dirigir sus respuestas hacia un grupo en específico.

La agrupación de respuestas no fue compleja, pues a pesar de ser abiertas, en su mayoría los estudiantes proporcionaron las mismas respuestas o muy similares, lo que hizo fácil la agrupación que se mostró en las gráficas anteriores, de la figura 1.42 a 1.51.

En general, creemos que las tendencias móviles están introduciéndose en las áreas de educación y es muy bien acogida por los estudiantes, específicamente por medio de su teléfono celular.

5.2 Trabajo Futuro

PAEPOO fue diseñado para dispositivos móviles que posean un navegador web, en la mayoría de los casos, PAEPOO funcionaba en su totalidad, ya que trabaja con HTML5 y CSS3 que la mayoría de los navegadores son capaces de leer, en especial los más recientes.

Android fue la idea original para realizar la plataforma, por ser el sistema operativo que se está introduciendo cada vez más en la sociedad y que ha llegado a dominar el mercado a nivel mundial. Pero las encuestas realizabas demostraban que para obtener la mayor cantidad de población para el trabajo de tesis, Android no era la opción adecuada, ya que un porcentaje muy bajo de estudiantes poseían este sistema operativo en su teléfono inteligente o en su tableta. Aunque la misma encuesta demostraba como las tendencias hacia adquirir un teléfono inteligente crecían cada vez más en un lapso de tiempo muy corto, puede ser posible que dentro de cierto periodo más se pueda realizar la plataforma sobre el sistema operativo Android y obtener la mayor cantidad de características disponibles que este sistema nos ofrece.

Objective-C para Iphone también es una opción pensada que podría ser la secuela de PAEPOO para Android. La idea es poder entrar en todos los ambientes que puedan operar cualquier dispositivo móvil, y con el navegador web para móviles ya hay una gran introducción en ello.

Utilizar PAEPOO de una forma más interactiva, donde los estudiantes puedan realizar pequeños y simples ejercicios dónde pongan a prueba sus conocimientos adquiridos dentro de la misma plataforma. Así ellos podrán ser autoevaluados y mejorar en aquellos puntos donde tengan deficiencia de conocimientos sobre el tema. La aplicación Definer ya estaba abierta con esta posibilidad, pero fue descartada a mitad de desarrollo de la plataforma para poder tener un control más amplio con las características que ya estaban disponibles en

todo PAEPOO, definiciones y ejemplos fue lo que perduró (figura X.1). Más sin embargo, la adición de ejercicios para esta aplicación será un cambio que mejorará aún más el rendimiento y utilidad de la plataforma. Una forma de evaluación donde los estudiantes vayan respondiendo preguntas, relacionando palabras, completando sentencias; si se requiere de una forma más divertido, pensando en crucigramas o sopas de letras de programación orientada a objetos, donde no se pierda en ningún momento la interacción con los estudiantes, incluso de una forma colaborativa donde entre aplicaciones, puedan comunicarse los estudiantes entre sí, sin importar el lugar donde hagan sesión o la hora a la que se conecten.



Figura 1.52. Definición y ejemplos activos, Ejercicios inactivos.

PAEPOO, como su nombre lo indica (Plataforma para el apoyo en el Aprendizaje y Enseñanza de Programación Orientada a Objetos) fue desarrollado únicamente con la intención de ayudar en el tema de programación orientada a objetos. En la encuesta los estudiantes la piden para poder utilizarla en otras asignaturas de su curso. La anexión de módulos para materias en PAEPOO podría significar un cambio drástico para poder agregar cualquier tipo de asignatura desde el lado del profesor, independientemente la asignatura que vaya a ofrecer.

La idea principal es utilizar todos los medios disponibles para apoyar en la educación de los estudiantes en cualquier área, en esta ocasión, fue programación orientada a objetos y lo que falta para la plataforma.

Dispositivos móviles han cautivado a todas las personas alrededor del mundo, hay que aprovechar esta familiaridad con esos dispositivos para utilizarlos a favor de la educación.

BIBLIOGRAFÍA

- Abdullah Mohd Zin; Sufian Idris, N. K. S. (2006). Seaair Annual Conference; Langkawi, September 5 – 7 2006.
- Allison, I., & Orton, P. (2002). A virtual learning environment for introductory programming. *Changes*, 48-52. Retrieved from <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:A+VIRTUAL+LEARNING+ENVIRONMENT+FOR+INTRODUCTORY#3>
- Biggs, J. (1999). Teaching for Quality Learning at University. *SRHE and Open University Press Buckingham*.
- Black, T. R. (2006). Helping novice programming students succeed. *Journal of Computing Sciences in Colleges*, 22(2), 109–114. Consortium for Computing Sciences in Colleges. Retrieved from <http://portal.acm.org/citation.cfm?id=1181922>
- Brown, J., & Baatard, G. (2008). Teaching Strategies in Large Class Programming Courses. *Sciences-New York* (p. 6).
- Bruce, K. B. (2004). Controversy on How to Teach CS 1 : A Discussion on the SIGCSE-members Mailing List #, 36(4), 29-35.
- C. Lin and C. Lin. (2008). Instructional strategies and methods of e-learning for nurturing appreciative, expressive, and creative abilities. *Journal of American Academy of Business*, 13.
- Duke, R., Salzman, E., Burmeister, J., Poon, J., & Murray, L. (2000). Teaching Programming to Beginners choosing the language is just the first step. *Computer*, 79-86.
- Economist.com. (2008). Halfway There: How to promote the spread of mobile phones among the world's poorest. Retrieved from http://www.economist.com/business/displaystory.cfm?story_id=11465558
- Eleconomista.com. (2013). México rebasa los 100 millones de celulares. Retrieved from <http://eleconomista.com.mx/tecnociencia/2013/07/02/mexico-rebasa-100-millones-celulares>
- Havey, M., Deitel, P., & Deitel, J. (2004). *Cómo Programar en C/C++ y Java* (p. 1154).
- Jacquot, J. P. (2002). Which use for Java in introductory courses? *Proceedings of the inaugural conference on the Principles and Practice of programming, 2002*

and *Proceedings of the second workshop on Intermediate representation engineering for virtual machines, 2002* (pp. 119–124). National University of Ireland. Retrieved from <http://portal.acm.org/citation.cfm?id=638500>

Jenkins, T. (2002). On the difficulty of learning to program. *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences* (Vol. 4, pp. 53–58). Retrieved from <http://www.ics.heacademy.ac.uk/Events/conf2002/tjenkins.pdf>

Jenkins, T., & Davy, J. (2002). Diversity and motivation in introductory programming. *Innovation in Teaching and Learning in Information and Computer Sciences*, 1(1), 9. Retrieved from <http://www.ics.heacademy.ac.uk/italics/issue1/tjenkins/003.html>

Kalles, D. (2008). Students working for students on programming courses. *Computers & Education*, 50(1), 91-97. doi:10.1016/j.compedu.2006.03.003

Kristiansen, T. (2001). M-learning: Experiences from the use of WAP as a supplement in learning. *Unpublished report on a pilot project in Co-operation between Insite, Ericsson, Telenor, Mobil and It Fornebu Knowation*.

Kölling, M., & Rosenberg, J. (2001). Guidelines for teaching object orientation with Java. *Proceedings of the 6th annual conference on Innovation and technology in computer science education - ITiCSE '01*, 33-36. New York, New York, USA: ACM Press. doi:10.1145/377435.377461

Lahtinen, E., Ala-Mutka, K., & Järvinen, H. M. (2005). A study of the difficulties of novice programmers. *ACM SIGCSE Bulletin*, 37(3), 14–18. ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1067453>

Licea, G., Juárez-Ramírez, R., Gaxiola, C., Aguilar, L., & Martínez, L. G. (2011). Teaching object-oriented programming with AEIOU. *Computer Applications in Engineering Education*, n/a-n/a. doi:10.1002/cae.20556

Motlik, S. (2008). Technical Evaluation Report# 63: Mobile Learning in Developing Nations. *Review of Research in Open and Distance Learning*, (2004). Retrieved from <http://www.doaj.org/doaj?func=abstract&id=273971>

Neubauer, B. (2002). The object-oriented paradigm: more natural or less familiar? *Journal of Computing Sciences in*, 280-289. Retrieved from <http://portal.acm.org/citation.cfm?id=771141.771186>

Pirani, J. (2004). Supporting E-learning in higher education. *EDUCAUSE Center for Applied Research*. Retrieved, 3, 88. Retrieved from <http://net.educause.edu/ir/library/pdf/ERS0303/ecm0303.pdf>

- Pressman, R. S. (2005). *Ingeniería Del Software. Un Enfoque Práctico* (p. 980).
- Quin, C. (2001). mLearning: Mobile, Wireless, In-Your-Pocket Learning.
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137–172. Taylor & Francis. Retrieved from <http://www.tandfonline.com/doi/abs/10.1076/csed.13.2.137.14200>
- Rößling, G., Bär, H., Trompler, C., & Choi, C.-M. (2003). Mobile computing in education. *ACM SIGCSE Bulletin*, 35(3), 227. doi:10.1145/961290.961583
- Urdan, T., Weggen, C. (2000). Corporate E-Learning: Exploring a new frontier. *WRHAMBRECHT+CO*.
- Weir, G. R. S., Vilner, T., Mendes, A. J., & Nordström, M. (2005). Difficulties teaching Java in CS1 and how we aim to solve them. *ACM SIGCSE Bulletin*, 37(3), 344. doi:10.1145/1151954.1067543
- Whittington, K. J. (2004). Infusing active learning into introductory programming courses. *Journal of Computing Sciences in Colleges*, 19(5), 249–259. Consortium for Computing Sciences in Colleges. Retrieved from <http://portal.acm.org/citation.cfm?id=1060111>
- Wills, G., Davis, H., & Cooke, E. (2004). Paired Programming for Non-Computing Students. *Higher Education*, 5. Retrieved from <http://eprints.ecs.soton.ac.uk/9658>
- Zualkernan, I. a., Allert, J., & Qadah, G. Z. (2006). Learning Styles of Computer Programming Students: A Middle Eastern and American Comparison. *IEEE Transactions on Education*, 49(4), 443-450. doi:10.1109/TE.2006.882366