

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA INFORMÁTICA

**SOFTWARE EDUCATIVO EN ENTORNO 3D QUE  
PERMITE EJERCITAR LOS CONOCIMIENTOS DE LA  
PROGRAMACIÓN ORIENTADA A OBJETOS CON  
JAVA**

**MICHELLE ANGELICA BOURGEOIS SALGADO**

INFORME FINAL DEL PROYECTO  
PARA OPTAR AL TÍTULO PROFESIONAL DE  
INGENIERO EJECUCIÓN EN INFORMÁTICA

NOVIEMBRE 2017

Pontificia Universidad Católica de Valparaíso  
Facultad de Ingeniería  
Escuela de Ingeniería Informática

**SOFTWARE EDUCATIVO EN ENTORNO 3D QUE  
PERMITE EJERCITAR LOS CONOCIMIENTOS DE LA  
PROGRAMACIÓN ORIENTADA A OBJETOS CON  
JAVA**

**MICHELLE ANGÉLICA BOURGEOIS SALGADO**

Profesor Guía: **Claudio Cubillos Figueroa**

Profesor Co-referente: **Rafael Mellado Silva**

Carrera: **Ingeniería Ejecución en Informática**

Noviembre

2017

# Dedicatoria

*Dedico este proyecto a mi madre que siempre ha estado para mí apoyándome en todas las cosas que me propongo, a mi familia que siempre están para mí dándome un apoyo incondicional, a mis amigos que estuvieron en este proceso apoyándome y dando ánimo.*

*Sin todas las personas maravillosas que me rodean no hubiera podido conseguir terminar este proyecto, muchas gracias por confiar en mí.*

# Índice

<b>Resumen.....</b>	<b>iv</b>
<b>Abstract.....</b>	<b>iv</b>
<b>Lista de Figuras .....</b>	<b>v</b>
<b>Lista de Tablas .....</b>	<b>viii</b>
<b>1      Introducción .....</b>	<b>1</b>
<b>1.1    Objetivos .....</b>	<b>1</b>
1.1.1    Objetivo General .....	1
1.1.2    Objetivos Específicos.....	1
1.1.3    Objetivos de Aprendizaje.....	2
<b>1.1    Metodología .....</b>	<b>2</b>
<b>1.2    Plan de Trabajo.....</b>	<b>2</b>
<b>2      Estado del Arte.....</b>	<b>3</b>
<b>2.1    Softwares Educativos Orientados a Objetos.....</b>	<b>3</b>
2.1.1    Scratch .....	3
2.1.1.1    Estudio.....	4
2.1.1.2    Similares .....	4
2.1.2    Alice.....	4
2.1.2.1    Estudio.....	5
2.1.2.2    Similares .....	5
2.1.3    Greenfoot .....	5
2.1.3.1    Estudios .....	6
2.1.4    Comparación .....	6
<b>3      Marco Teórico .....</b>	<b>8</b>
<b>3.1    Formas de Enseñar Orientación a Objetos.....</b>	<b>8</b>
3.1.1    Cursos de POO.....	8
3.1.1.1    Aprendizaje por Recepción.....	8
3.1.1.2    Aprendizaje por Descubrimiento .....	8
3.1.1.3    Aprendizaje Colaborativo .....	9
3.1.1.4    Aprendizaje Basado en Problemas.....	9
3.1.2    Video Tutoriales.....	9
3.1.2.1    Como Crear un Video Tutorial .....	10
3.1.2.2    Requisitos Para su Efectividad.....	10
3.1.3    Mediante Cuentos .....	10
3.1.3.1    Funcionamiento .....	10
3.1.3.2    Ventajas .....	12
3.1.4    Software Educativo .....	12

	<b>3.2 Dificultades de Enseñar POO .....</b>	<b>13</b>
	3.2.1 Entorno de Programación.....	13
	3.2.2 Cambio de Paradigma .....	13
	<b>3.3 Gamificación.....</b>	<b>13</b>
	3.3.1 Mecánicas de la Gamificación .....	13
	3.3.2 Dinámicas de Juego de la Gamificación .....	14
<b>4</b>	<b>Tecnologías .....</b>	<b>16</b>
	<b>4.1 Unity .....</b>	<b>16</b>
	4.1.1 Blender.....	16
	<b>4.2 Photoshop.....</b>	<b>17</b>
	<b>4.3 JDK.....</b>	<b>17</b>
	<b>4.4 LMMS .....</b>	<b>17</b>
<b>5</b>	<b>Dominio.....</b>	<b>18</b>
	<b>5.1 Temas de Aprendizaje .....</b>	<b>18</b>
<b>6</b>	<b>Problemática Propuesta .....</b>	<b>19</b>
	<b>6.1 Historia.....</b>	<b>19</b>
	<b>6.2 Niveles .....</b>	<b>20</b>
	6.2.1 Introducción .....	20
	6.2.2 Nivel 1.....	21
	6.2.3 Nivel 2.....	21
	6.2.4 Nivel 3.....	22
	6.2.5 Nivel 4.....	22
	6.2.6 Nivel 5.....	22
	6.2.7 Nivel 6.....	23
	6.2.7.1 Reglas del Juego .....	24
	<b>6.3 Puntaje .....</b>	<b>24</b>
	<b>6.4 Requerimientos.....</b>	<b>24</b>
	6.4.1 Requerimientos Funcionales .....	25
	6.4.2 Requerimientos No Funcionales .....	28
	<b>6.5 Evaluación.....</b>	<b>29</b>
<b>7</b>	<b>Diseño .....</b>	<b>30</b>
	<b>7.1 Diagrama de Actividad .....</b>	<b>30</b>
	<b>7.2 Diagrama de Clases.....</b>	<b>33</b>
	<b>7.3 Diagrama Casos de Uso .....</b>	<b>39</b>
<b>8</b>	<b>Pruebas de usuario.....</b>	<b>40</b>
	<b>8.1 PreTest .....</b>	<b>40</b>
	<b>8.2 PosTest .....</b>	<b>46</b>

<b>9</b>	<b>Pruebas Realizadas a Estudiantes sin Conocimiento de POO.....</b>	<b>54</b>
<b>9.1</b>	<b>Respuestas de los Participantes por cada Pregunta PreTest.....</b>	<b>55</b>
9.1.1	Conclusiones de las Respuestas .....	70
<b>9.2</b>	<b>Respuestas de los Participantes por cada Pregunta PosTest.....</b>	<b>71</b>
9.2.1	Conclusiones de las Respuestas .....	85
<b>9.3</b>	<b>Comparación PreTest y PosTest por Temáticas .....</b>	<b>86</b>
9.3.1	Temática: Clases .....	86
9.3.2	Temática: Arreglos.....	87
9.3.3	Temática: Métodos.....	89
9.3.4	Temática: Atributos.....	90
9.3.5	Temática: Objetos .....	91
<b>10</b>	<b>Pruebas de Usuario con Conocimientos Previos a POO .....</b>	<b>93</b>
<b>10.1</b>	<b>Respuestas de los Participantes por cada Pregunta PreTest.....</b>	<b>93</b>
10.1.1	Conclusiones de las Respuestas .....	107
<b>10.2</b>	<b>Respuestas de los Participantes por cada Pregunta PosTest.....</b>	<b>107</b>
10.2.1	Conclusiones de las Respuestas .....	120
<b>10.3</b>	<b>Comparación PreTest y PosTest por Temáticas .....</b>	<b>121</b>
10.3.1	Temática: Clases .....	121
10.3.2	Temática: Arreglos.....	122
10.3.3	Temática: Atributos.....	123
10.3.4	Temática: Métodos.....	124
10.3.5	Temática: Objetos .....	125
<b>11</b>	<b>Conclusiones de Ambas Pruebas .....</b>	<b>126</b>
<b>12</b>	<b>Conclusión .....</b>	<b>127</b>
<b>13</b>	<b>Referencias.....</b>	<b>128</b>
<b>14</b>	<b>Anexo.....</b>	<b>132</b>
<b>14.1</b>	<b>A: Carta Gantt .....</b>	<b>133</b>

## Resumen

En el presente documento se dará a conocer un software educativo que tendrá como nombre Ashy.alRescate() que tiene como objetivo enseñar y ejercitar los conocimientos de la programación orientada a objetos (POO) con el lenguaje de programación Java. Para llevar a cabo este software, se hará una investigación con las distintas formas de enseñar orientación a objetos (OO), para elegir cual de estas metodologías es la más adecuada para la implantación del videojuego. Adicionalmente se buscarán herramientas similares a la propuesta antes señalada, con el fin de ver la repercusión en el público y utilizar alguna estrategia utilizada por estas. Se espera obtener un software educativo que motive al público interesado y que cumplan con los objetivos acordados.

*Palabras-claves: programación orientada a objetos, orientada a objetos, lenguaje de programación Java.*

## Abstract

This document will present an educational software called Ashy.alRescate () that aims to teach and exercise the knowledge of object-oriented programming (OOP) with the Java programming language. In order to carry out this software, an investigation will be done with the different ways of teaching object orientation (OO), to choose which of these methodologies is the most appropriate for the implementation of the video game. In addition, tools similar to the aforementioned proposal will be sought, in order to see the impact on the public and to use some strategy used by them. It is hoped to obtain educational software that motivates the interested public and that they fulfill the agreed objectives.

*Keywords:* Object-oriented programming, object-oriented, Java programming language.

## Lista de Figuras

Figura 1.1 Resumen de la carta Gantt .....	2
Figura 2.1 Implementación de Scratch [6] .....	4
Figura 2.2 Implementación de Alice [13] .....	5
Figura 2.3 Implementación Greenfoot [16] .....	6
Figura 3.1 Diagrama de clases [31].....	12
Figura 3.2 Mecánicas de la Gamificación [34] .....	14
Figura 3.3 Dinámicas de juego más destacadas [34] .....	15
Figura 5.1 Ranking de los lenguajes de programación más usados. ....	18
Figura 6.1 Demostración del comic [45].....	21
Figura 6.2 Demostración elección de personajes (elaboración propia) .....	22
Figura 6.3 Prototipo de tutorial para objetos (elaboración propia) .....	23
Figura 6.4 Prototipo tutorial para objetos 2 (elaboración propia) .....	23
Figura 7.1 Inicio del diagrama de actividad.....	30
Figura 7.2 Continuación del diagrama de flujo parte 1 .....	31
Figura 7.3 Continuación del diagrama de flujo parte 2.....	32
Figura 7.4 Finalización del diagrama de actividad. ....	33
Figura 7.5 Clases que no contienen asociaciones .....	34
Figura 7.6 Diagrama de clases con sus asociaciones parte 1 .....	35
Figura 7.7 Diagrama de clases con sus asociaciones parte 2 .....	36
Figura 7.8 Diagrama de clases junto con sus asociaciones .....	37
Figura 7.9 Diagrama de clases junto con sus asociaciones parte 1 .....	37
Figura 7.10 Diagrama de clases junto con sus asociaciones parte 2 .....	38
Figura 7.11 Diagrama de clases junto con sus asociaciones parte 3 .....	38
Figura 7.12 Diagrama de casos de uso.....	39
Figura 8.1 PreTest de la pregunta 1 a la 3 .....	40
Figura 8.2 PreTest de la pregunta 4 a la 6 .....	41
Figura 8.3 PreTest de la pregunta 7 a la 9 .....	42
Figura 8.4 PreTest de la pregunta 10 a la 12 .....	43
Figura 8.5 PreTest de la pregunta 13 a la 14 .....	44
Figura 8.6 PreTest de la pregunta 15 .....	44
Figura 8.7 PosTest de la pregunta 1 a la 3 .....	47
Figura 8.8 PosTest de la pregunta 4 a la 6 .....	48
Figura 8.9 PosTest de la pregunta 7 a la 10 .....	49
Figura 8.10 PosTest de la pregunta 11 a la 13 .....	50
Figura 8.11 PosTest de la pregunta 14 a la 15 .....	51
Figura 9.1 Los participantes haciendo las pruebas.....	54
Figura 9.2 Realización de las pruebas.....	55
Figura 9.3 Grafico de la pregunta 1, usuarios sin saber Java .....	56
Figura 9.4 Grafico de la pregunta 2, usuarios sin saber Java .....	57
Figura 9.5 Grafico de la pregunta 3, usuarios sin saber Java .....	58
Figura 9.6 Grafico de la pregunta 4, usuarios sin saber Java .....	59
Figura 9.7 Grafico de la pregunta 5, usuarios sin saber Java .....	60
Figura 9.8 Grafico de la pregunta 6, usuarios sin saber Java .....	61
Figura 9.9 Grafico de la pregunta 7, usuarios sin saber Java .....	62



Figura 9.10 Grafico de la pregunta 8, usuarios sin saber Java.....	63
Figura 9.11 Grafico de la pregunta 9, usuarios sin saber Java.....	64
Figura 9.12 Grafico de la pregunta 10, usuarios sin saber Java.....	65
Figura 9.13 Grafico de la pregunta 11, usuarios sin saber Java.....	66
Figura 9.14 Grafico de la pregunta 12, usuarios sin saber Java.....	67
Figura 9.15 Grafico de la pregunta 13, usuarios sin saber Java.....	68
Figura 9.16 Grafico de la pregunta 14, usuarios sin saber Java.....	69
Figura 9.17 Grafico de la pregunta 15, usuarios sin saber Java.....	70
Figura 9.18 Grafico de la pregunta 1 del posTest, usuarios sin saber Java.....	71
Figura 9.19 Grafico de la pregunta 2 del posTest, usuarios sin saber Java.....	72
Figura 9.20 Grafico de la pregunta 3 del posTest, usuarios sin saber Java.....	73
Figura 9.21 Grafico de la pregunta 4 del posTest, usuarios sin saber Java.....	74
Figura 9.22 Grafico de la pregunta 5 del posTest, usuarios sin saber Java.....	75
Figura 9.23 Grafico de la pregunta 6 del posTest, usuarios sin saber Java.....	76
Figura 9.24 Grafico de la pregunta 7 del posTest, usuarios sin saber Java.....	77
Figura 9.25 Grafico de la pregunta 8 del posTest, usuarios sin saber Java.....	78
Figura 9.26 Grafico de la pregunta 9 del posTest, usuarios sin saber Java.....	79
Figura 9.27 Grafico de la pregunta 10 del posTest, usuarios sin saber Java.....	80
Figura 9.28 Grafico de la pregunta 11 del posTest, usuarios sin saber Java.....	81
Figura 9.29 Grafico de la pregunta 13 del posTest, usuarios sin saber Java.....	82
Figura 9.30 Grafico de la pregunta 14 del posTest, usuarios sin saber Java.....	83
Figura 9.31 Grafico de la pregunta 15 del posTest, usuarios sin saber Java.....	84
Figura 9.32 Grafico temática clases preTest, participantes sin saber Java .....	86
Figura 9.33 Grafico temática clases posTest, participantes sin saber Java.....	86
Figura 9.34 Grafico temática arreglos preTest, participantes sin saber Java .....	87
Figura 9.35 Grafico temática arreglos posTest, participantes sin saber Java.....	88
Figura 9.36 Grafico temática métodos preTest, participantes sin saber Java .....	89
Figura 9.37 Grafico temática métodos posTest, participantes sin saber Java.....	89
Figura 9.38 Grafico temática atributos preTest, participantes sin saber Java.....	90
Figura 9.39 Grafico temática métodos posTest, participantes sin saber Java.....	90
Figura 9.40 Grafico temática objetos preTest, participantes sin saber Java .....	91
Figura 9.41 Grafico temática objetos posTest, participantes sin saber Java.....	92
Figura 10.1 Grafico pregunta 1 preTest, participantes que sabían Java.....	94
Figura 10.2 Grafico pregunta 2 preTest, participantes que sabían Java.....	95
Figura 10.3 Grafico pregunta 3 preTest, participantes que sabían Java.....	96
Figura 10.4 Grafico pregunta 4 preTest, participantes que sabían Java.....	97
Figura 10.5 Grafico pregunta 5 preTest, participantes que sabían Java.....	98
Figura 10.6 Grafico pregunta 6 preTest, participantes que sabían Java.....	99
Figura 10.7 Grafico pregunta 7 preTest, participantes que sabían Java.....	100
Figura 10.8 Grafico pregunta 8 preTest, participantes que sabían Java.....	101
Figura 10.9 Grafico pregunta 9 preTest, participantes que sabían Java.....	102
Figura 10.10 Grafico pregunta 10 preTest, participantes que sabían Java.....	102
Figura 10.11 Grafico pregunta 11 preTest, participantes que sabían Java.....	103
Figura 10.12 Grafico pregunta 12 preTest, participantes que sabían Java.....	104
Figura 10.13 Grafico pregunta 13 preTest, participantes que sabían Java.....	105
Figura 10.14 Grafico pregunta 14 preTest, participantes que sabían Java.....	106
Figura 10.15 Grafico pregunta 15 preTest, participantes que sabían Java.....	107

Figura 10.16 Grafico pregunta 1 posTest, participantes que sabían Java. ....	108
Figura 10.17 Grafico pregunta 2 posTest, participantes que sabían Java. ....	108
Figura 10.18 Grafico pregunta 3 posTest, participantes que sabían Java. ....	109
Figura 10.19 Grafico pregunta 4 posTest, participantes que sabían Java. ....	110
Figura 10.20 Grafico pregunta 5 posTest, participantes que sabían Java. ....	111
Figura 10.21 Grafico pregunta 6 posTest, participantes que sabían Java. ....	112
Figura 10.22 Grafico pregunta 7 posTest, participantes que sabían Java. ....	112
Figura 10.23 Grafico pregunta 8 posTest, participantes que sabían Java. ....	113
Figura 10.24 Grafico pregunta 9 posTest, participantes que sabían Java. ....	114
Figura 10.25 Grafico pregunta 10 posTest, participantes que sabían Java. ....	115
Figura 10.26 Grafico pregunta 11 posTest, participantes que sabían Java. ....	116
Figura 10.27 Grafico pregunta 12 posTest, participantes que sabían Java. ....	117
Figura 10.28 Grafico pregunta 13 posTest, participantes que sabían Java. ....	118
Figura 10.29 Grafico pregunta 14 posTest, participantes que sabían Java. ....	119
Figura 10.30 Grafico temática clases PreTest, Alumnos que saben Java .....	121
Figura 10.31 Grafico temática clases PosTest, Alumnos que saben Java.....	121
Figura 10.32 Grafico temática arreglos PreTest, Alumnos que saben Java.....	122
Figura 10.33 Grafico temática arreglos PosTest, Alumnos que saben Java .....	122
Figura 10.34 Grafico temática atributos PreTest, Alumnos que saben Java.....	123
Figura 10.35 Grafico temática atributos PosTest, Alumnos que saben Java .....	123
Figura 10.36 Grafico temática métodos PreTest, Alumnos que saben Java .....	124
Figura 10.37 Grafico temática métodos PosTest, Alumnos que saben Java.....	124
Figura 10.38 Grafico temática objetos PreTest, Alumnos que saben Java .....	125
Figura 10.39 Grafico temática objetos PosTest, Alumnos que saben Java.....	125
Figura A.1 Carta Gantt Iteración 1 .....	133
Figura A.2 Carta Gantt Iteración 2.....	134
Figura A.3 Carta Gantt Iteración 3.....	135
Figura A.4 Carta Gantt Iteración 4.....	136
Figura A.5 Carta Gantt Iteración 5.....	137
Figura A.6 Carta Gantt Iteración 6.....	138
Figura A.7 Carta Gantt Iteración 7.....	139
Figura A.8 Carta Gantt Iteración 8.....	140
Figura A.9 Carta Gantt Iteración 9.....	141
Figura A.10 Carta Gantt Iteración 10.....	142
Figura A.11 Carta Gantt Iteración 11.....	143

## Lista de Tablas

Tabla 2.1 Comparación entre los softwares educativos actuales (elaboración propia).....	7
Tabla 3.1 Identificación problemática (elaboración propia) .....	11
Tabla 3.2 Identificador de actividades (elaboración propia).....	11
Tabla 6.1 Acciones que realizaran los personajes (elaboración propia) .....	20
Tabla 6.6.2 Requerimiento funcional 1 .....	25
Tabla 6.6.3 Requerimiento funcional 2 .....	26
Tabla 6.6.4 Requerimiento funcional 3 .....	26
Tabla 6.6.5 Requerimiento funcional 4 .....	27
Tabla 6.6.6 Requerimiento funcional 5 .....	27
Tabla 6.6.7 Requerimiento no funcional 1 .....	28
Tabla 6.6.8 Requerimiento no funcional 2 .....	29
Tabla 8.1 Tabla de identificación de contenidos del PreTest.....	45
Tabla 8.2 Tabla que identifica las categorías de las preguntas del posTest .....	51
Tabla 9.1 Tabla de la pregunta 1, usuarios sin saber Java .....	55
Tabla 9.2 Tabla de la pregunta 2, usuarios sin saber Java .....	56
Tabla 9.3 Tabla de la pregunta 3, usuarios sin saber Java .....	57
Tabla 9.4 Tabla de la pregunta 4, usuarios sin saber Java .....	58
Tabla 9.5 Tabla de la pregunta 5, usuarios sin saber Java .....	59
Tabla 9.6 Tabla de la pregunta 6, usuarios sin saber Java .....	60
Tabla 9.7 Tabla de la pregunta 7, usuarios sin saber Java .....	61
Tabla 9.8 Tabla de la pregunta 8, usuarios sin saber Java .....	62
Tabla 9.9 Tabla de la pregunta 9, usuarios sin saber Java .....	63
Tabla 9.10 Tabla de la pregunta 10, usuarios sin saber Java .....	64
Tabla 9.11 Tabla de la pregunta 11, usuarios sin saber Java .....	65
Tabla 9.12 Tabla de la pregunta 12, usuarios sin saber Java .....	66
Tabla 9.13 Tabla de la pregunta 13, usuarios sin saber Java .....	67
Tabla 9.14 Tabla de la pregunta 14, usuarios sin saber Java .....	68
Tabla 9.15 Tabla de la pregunta 15, usuarios sin saber Java .....	69
Tabla 9.16 Tabla de la pregunta 1 del posTest, usuarios sin saber Java .....	71
Tabla 9.17 Tabla de la pregunta 2 del posTest, usuarios sin saber Java .....	72
Tabla 9.18 Tabla de la pregunta 3 del posTest, usuarios sin saber Java .....	72
Tabla 9.19 Tabla de la pregunta 4 del posTest, usuarios sin saber Java .....	73
Tabla 9.20 Tabla de la pregunta 5 del posTest, usuarios sin saber Java .....	74
Tabla 9.21 Tabla de la pregunta 6 del posTest, usuarios sin saber Java .....	75
Tabla 9.22 Tabla de la pregunta 7 del posTest, usuarios sin saber Java .....	76
Tabla 9.23 Tabla de la pregunta 8 del posTest, usuarios sin saber Java .....	77
Tabla 9.24 Tabla de la pregunta 9 del posTest, usuarios sin saber Java .....	78
Tabla 9.25 Tabla de la pregunta 10 del posTest, usuarios sin saber Java .....	79
Tabla 9.26 Tabla de la pregunta 11 del posTest, usuarios sin saber Java .....	80
Tabla 9.27 Tabla de la pregunta 12 del posTest, usuarios sin saber Java .....	81
Tabla 9.28 Tabla de la pregunta 13 del posTest, usuarios sin saber Java .....	82
Tabla 9.29 Tabla de la pregunta 14 del posTest, usuarios sin saber Java .....	83
Tabla 9.30 Tabla de la pregunta 15 del posTest, usuarios sin saber Java .....	84
Tabla 10.1 Tabla pregunta 1 preTest, participantes que sabían Java. ....	93

Tabla 10.2 Tabla pregunta 2 preTest, participantes que sabían Java. ....	94
Tabla 10.3 Tabla pregunta 3 preTest, participantes que sabían Java. ....	95
Tabla 10.4 Tabla pregunta 4 preTest, participantes que sabían Java. ....	96
Tabla 10.5 Tabla pregunta 5 preTest, participantes que sabían Java. ....	97
Tabla 10.6 Tabla pregunta 6 preTest, participantes que sabían Java. ....	98
Tabla 10.7 Tabla pregunta 7 preTest, participantes que sabían Java. ....	99
Tabla 10.8 Tabla pregunta 8 preTest, participantes que sabían Java. ....	100
Tabla 10.9 Tabla pregunta 9 preTest, participantes que sabían Java. ....	101
Tabla 10.10 Tabla pregunta 10 preTest, participantes que sabían Java. ....	102
Tabla 10.11 Tabla pregunta 11 preTest, participantes que sabían Java. ....	103
Tabla 10.12 Tabla pregunta 12 preTest, participantes que sabían Java. ....	103
Tabla 10.13 Tabla pregunta 13 preTest, participantes que sabían Java. ....	104
Tabla 10.14 Tabla pregunta 14 preTest, participantes que sabían Java. ....	105
Tabla 10.15 Tabla pregunta 15 preTest, participantes que sabían Java. ....	106
Tabla 10.16 Tabla pregunta 1 posTest, participantes que sabían Java. ....	107
Tabla 10.17 Tabla pregunta 2 posTest, participantes que sabían Java. ....	108
Tabla 10.18 Tabla pregunta 3 posTest, participantes que sabían Java. ....	109
Tabla 10.19 Tabla pregunta 4 posTest, participantes que sabían Java. ....	109
Tabla 10.20 Tabla pregunta 5 posTest, participantes que sabían Java. ....	110
Tabla 10.21 Tabla pregunta 6 posTest, participantes que sabían Java. ....	111
Tabla 10.22 Tabla pregunta 7 posTest, participantes que sabían Java. ....	112
Tabla 10.23 Tabla pregunta 8 posTest, participantes que sabían Java. ....	112
Tabla 10.24 Tabla pregunta 9 posTest, participantes que sabían Java. ....	113
Tabla 10.25 Tabla pregunta 10 posTest, participantes que sabían Java. ....	114
Tabla 10.26 Tabla pregunta 11 posTest, participantes que sabían Java. ....	115
Tabla 10.27 Tabla pregunta 12 posTest, participantes que sabían Java. ....	116
Tabla 10.28 Tabla pregunta 13 posTest, participantes que sabían Java. ....	117
Tabla 10.29 Tabla pregunta 14 posTest, participantes que sabían Java. ....	118
Tabla 10.30 Tabla pregunta 15 posTest, participantes que sabían Java. ....	119

# 1 Introducción

A la hora de aprender algún lenguaje de programación se hace difícil entender algunos conceptos explicados de forma teórica [1]. Es por lo que los profesores han buscado diversas formas de entregar dicho conocimiento utilizando herramientas como dibujos, videos, imágenes, entre otros. ¿Pero, qué tal si mediante un video juego se puedan explicar y reforzar dichos contenidos mientras que a la vez se está divirtiéndose?

Los video juegos se han vuelto una gran atracción por el público; tanto niños, adolescentes y adultos disfrutan divertirse con diversos juegos dependiendo de sus gustos. Los estudios impartidos hasta el momento [2] indican que se tiene una reacción positiva al momento de mezclar un juego con la educación. Considerando esto, los softwares educativos tienen grandes beneficios para el aprendizaje del estudiante, entre estos beneficios se pueden destacar los siguientes; permite simular procesos complejos, permiten transmitir gran volumen de información en un menor tiempo, constituye una nueva, atractiva, dinámica y rica fuente de conocimientos [3].

Una de las artistas de la programación para computadores, es la Orientación a Objetos, es de esta forma, que al momento de enseñar programación orientada a objetos se identifica que los alumnos tienden a generar dificultades debido a diversos problemas tanto en la teoría como en la práctica; tal como se ve en [1], algunos problemas que tienen los estudiantes con la orientación a objetos es el cambio del paradigma estructurado [4] al paradigma orientado a objetos, esto se puede dar debido a que la forma de presentar la teoría no es la adecuada ante el cambio, el nivel de abstracción de los conceptos y la cantidad de carga teórica [1].

Por lo antes mencionado y respondiendo a la pregunta antes señalada se pretende realizar un software educativo que busca reforzar y enseñar lenguaje de programación en Java. Para lo anterior se realizará uso de entornos 3D.

## 1.1 Objetivos

A continuación, se presentan el objetivo general y los objetivos específicos del proyecto abordado.

### 1.1.1 Objetivo General

Desarrollar un software educativo que permita reforzar contenido del lenguaje de programación Java mediante la interacción del usuario con el mundo 3D.

### 1.1.2 Objetivos Específicos

Los objetivos específicos que se plantean en este proyecto son:

- Estudiar herramientas similares, considerando su impacto en los usuarios y las técnicas utilizadas por éstas.

- Estudiar distintas formas de enseñanza para el lenguaje Java que den un apoyo en la creación del software.
- Desarrollar las escenas del video juego que contenga la historia que permitirá reforzar el lenguaje Java.
- Establecer tutoriales que serán capaces de enseñar y/o guiar al usuario ante una posible solución del problema asignado para ese nivel.
- Validar que el juego cumpla con el objetivo de reforzar el contenido Java, mediante pruebas de usuarios.

### 1.1.3 Objetivos de Aprendizaje

- Enseñar el lenguaje Java con los siguientes contenidos: método main, comentarios, tipos de datos primitivos, arreglos, métodos, instanciación de objetos; interactuando con el mundo virtual mediante la programación.
- Fomentar la correcta comprensión de lectura, mediante la necesidad de seguir instrucciones en cada nivel el juego.

## 1.1 Metodología

La metodología de proceso de desarrollo de software que se utilizará será incremental debido a que se contara con poco personal por lo cual esta metodología es útil para eso. En cada iteración en encontrará el análisis, diseño, código y pruebas que conlleva a la realización del proyecto, la gran mayoría de las iteraciones corresponde a un nivel del software educativo.

Por cada incremento se entregará un producto operacional, tanto del informe como del software educativo.

## 1.2 Plan de Trabajo

Para una mayor comprensión del plan de trabajo propuesto, se presenta la carta Gantt con sus respectivas iteraciones, mostrando los tiempos que conlleva realizar cada una de estas, en la Figura 1.1 se puede ver un resumen de la carta Gantt la cual se encuentra más detallada en el Anexo A.

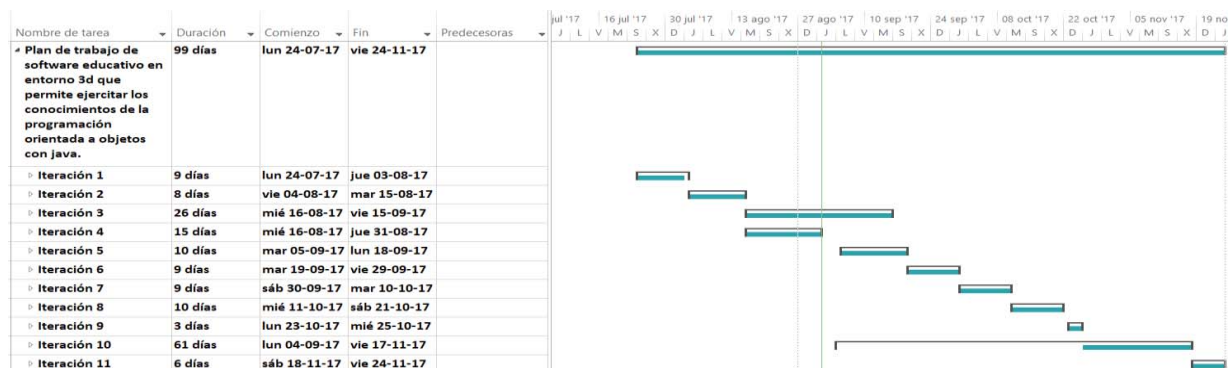


Figura 1.1 Resumen de la carta Gantt



## **2 Estado del Arte**

Los softwares educativos brindan una amplia gama de materiales para enseñar tanto matemáticas, lenguaje, ciencia social como también materiales universitarios, podemos encontrar algunos de estos en [5]. En la universidad buscan enseñar contenidos que ayudan a los estudiantes en su formación como profesionales, estas parten enseñando desde lo más básico hasta lo más complejo. En algunas instituciones que imparten informática se comienza enseñando lenguaje de programación con enfoque estructurado, lo cual proporciona una programación más sencilla y fácil de implementar, pero surge un problema al momento de pasar de este paradigma de programación estructurada a la programación orientada a objetos. Tal como se aprecia en [1] se da a conocer que el cambio de una programación estructura a una orientada a objetos lleva a los estudiantes a seguir pensando de forma estructurada, dificultando la idea de que una programación ya más llevado a lo real como es la orientación a objetos. En dicha investigación se menciona que el entorno de programación igual dificulta la manera de aprender dichos contenidos, esto debido a la complejidad de la herramienta que está diseñado para un ámbito más profesional que una forma más didáctica para que el alumno aprenda o pueda ensayar lo aprendido en las clases de programación.

### **2.1 Softwares Educativos Orientados a Objetos**

Tomando en cuenta lo mencionado anteriormente se pueden encontrar algunos softwares educativos didácticos capaces de poner en práctica la orientación a objetos, los cuales son revisados en este apartado.

#### **2.1.1 Scratch**

Es un juego que pretende potenciar el pensamiento computacional desde niños hasta adultos. Principalmente es utilizado para niños, a los cuales se pretende incentivar el pensamiento computacional. Con Scratch puedes programar tus propias historias interactivas, juegos y animaciones [6].

Se basa en utilizar objetos a los que se les puede dar alguna interacción, estas interacciones se encuentran predefinidas y se puede poner en uso arrastrando y soltando en el panel de ejecución, en la Figura 2.1, se muestra un ejemplo de Scratch.

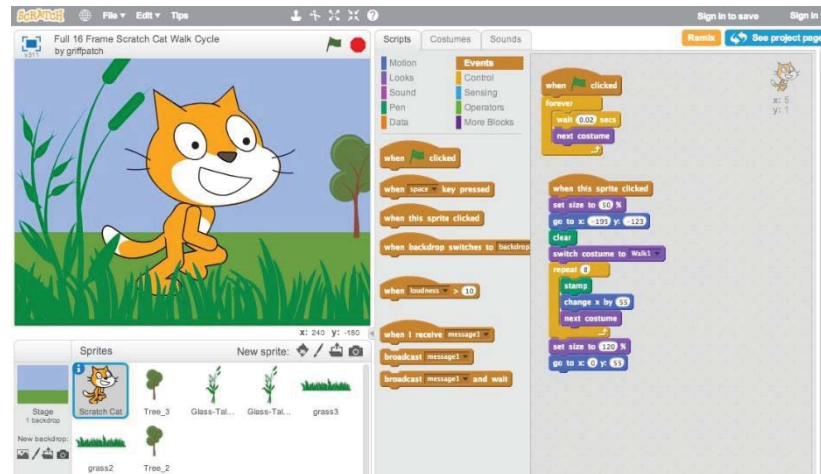


Figura 2.1 Implementación de Scratch [6]

### 2.1.1.1 Estudio

Los estudios realizados fueron en una Institución educativa ubicada en la ciudad de Aguachica, Cesar (Colombia) con estudiantes de los grados Noveno de Nivel Básica Secundaria apoyados por docentes de la asignatura Tecnología e Informática [7] afirman que se presentó una gran motivación de los alumnos al utilizar esta herramienta, también menciona que no solo potencio las áreas informáticas sino que también áreas como física y matemáticas se vieron favorecidas al ser aplicadas con este software.

### 2.1.1.2 Similares

- BeetleBlocks: tiene la mayoría de las mismas categorías que Scratch. Es un entorno de programación por bloques para generar formas 3D, que incorpora la posibilidad de exportarlas a formatos adecuados para su fabricación en 3D [8]
- ScratchX: Es una plataforma para probar extensiones experimentales de JavaScript Scratch [9].
- Mblock: es un entorno de programación gráfica basado en Scratch 2.0, para introducir la robótica de forma sencilla y enseñar a programar robots basados en Arduino [10].
- Snap!: es una modificación de Scratch. Incluye muchas características que no están disponibles en Scratch sin muchas modificaciones importantes en el código fuente [11]. La mayor diferencia es que puede crear clases.
- Aprendo a programar: son cursos de programación basado en la metodología Scratch [12].

### 2.1.2 Alice

Es un software educativo cuya finalidad es ser el primer contacto para un estudiante de programación orientada a objetos. Se basa en arrastrar y soltar los elementos gráficos del sistema, para crear objetos o interacciones que se quieran realizar, por lo que el estudiante no necesita saber todas las reglas de programación para realizar programas. Por lo tanto, Alice es



utilizada como material de apoyo para los profesores quienes introducen la programación orientada a objetos, ya que esta herramienta no cuenta con sistemas de tutoriales para enseñar los conceptos de la programación a objetos [13], en la Figura 2.2, se muestra un ejemplo de Alice.

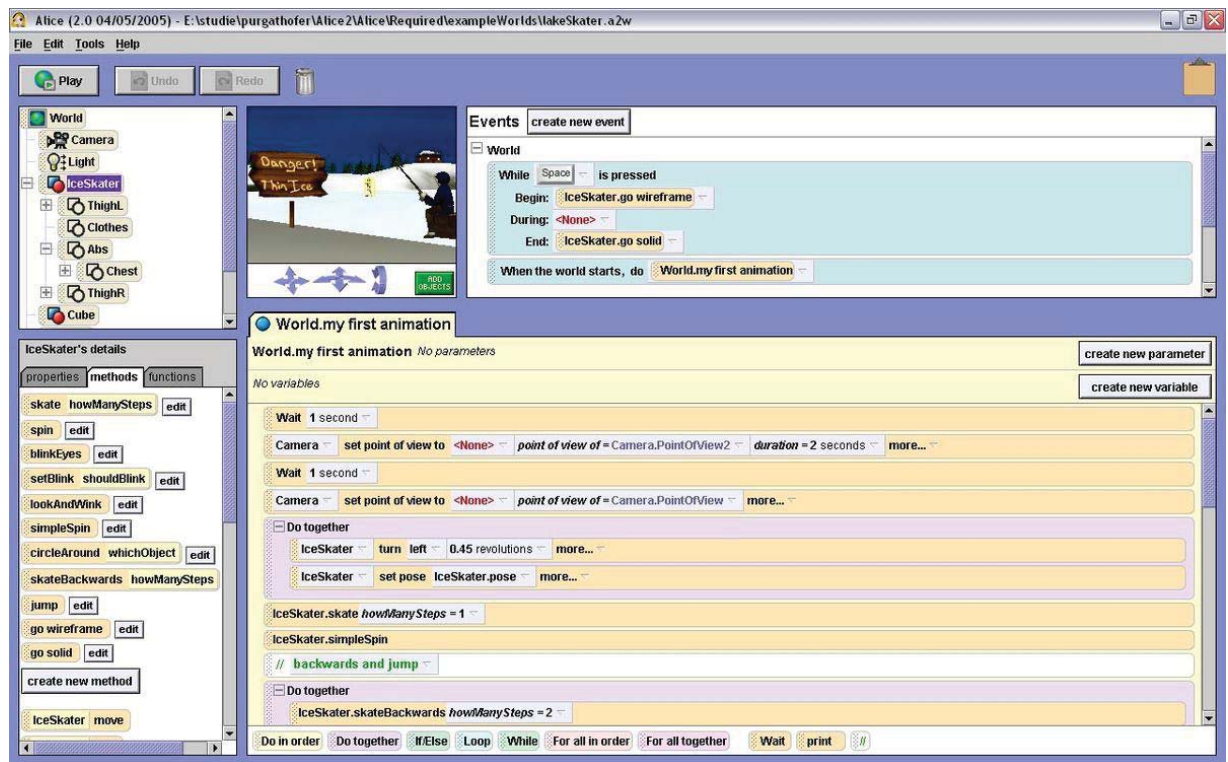


Figura 2.2 Implementación de Alice [13]

### 2.1.2.1 Estudio

Experiencias aplicadas a alumnos en la enseñanza universitaria [14] dan a conocer una respuesta positiva por parte de los estudiando dejando en evidencia que implantando un mundo 3D para la enseñanza de dichos contenidos es más fácil de comprender que de una manera más tradicional.

### 2.1.2.2 Similares

Rebeca: Es una versión traducida, adaptable a cualquier idioma y basada en el sistema Alice. Permite, de una forma sencilla, crear animaciones para contar historias, videojuegos interactivos o videos para compartir en Internet [15].

### 2.1.3 Greenfoot

Greenfoot nos permite crear videojuegos desarrollándolos en un mundo gráfico en 2D, más que todo es un entorno de desarrollo en Java con fines educativos. Al igual que los juegos antes mencionados tenemos la interacción con los objetos y le damos alguna interacción, lo que hace más novedosa a esta herramienta es que contiene un editor de código, por lo que, se podrá

desarrollar en lenguaje java sin la necesidad de arrastrar y soltar el código, como lo hacían las herramientas antes mencionadas [16], en la Figura 2.2, se muestra un ejemplo de Greenfoot.

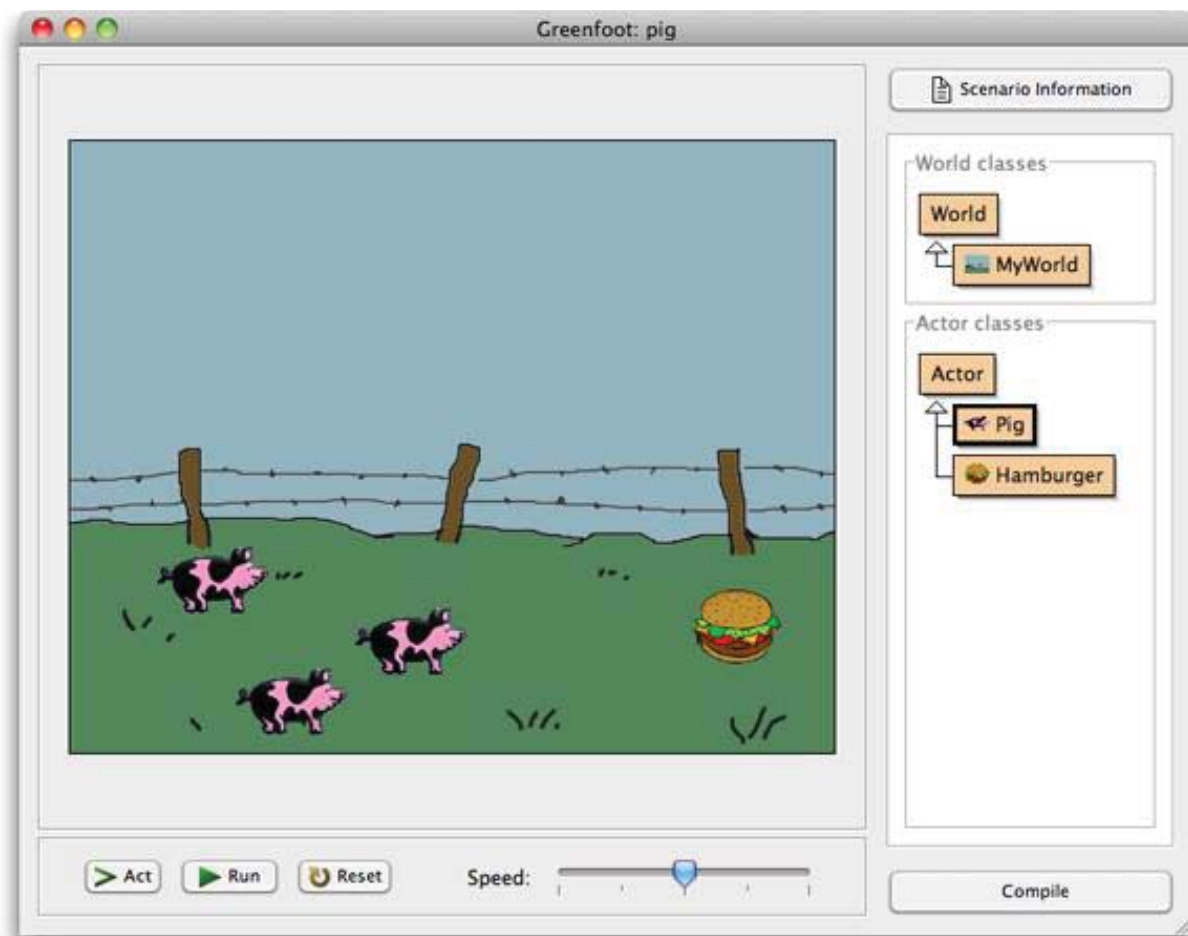


Figura 2.3 Implementación Greenfoot [16]

### 2.1.3.1 Estudios

Estudios realizados con esta herramienta muestran una respuesta satisfactoria ante los problemas propuestos [17]. Se concluye que es una buena herramienta para utilizar de apoyo en la enseñanza de orientación a objetos en el lenguaje Java. Sin embargo, para la utilización como medio de enseñanza para los alumnos no es apropiada, ya que no cuenta con sistemas que aporten algún material de apoyo con respecto a el lenguaje java o a la orientación a objetos, por lo que hace necesaria el aprendizaje previo.

### 2.1.4 Comparación

Por lo tanto, estos softwares ayudan al docente a exponer los contenidos de manera interactiva, cada uno de estos de una forma distinta. A continuación, se muestra una tabla de comparación entre los 3 tipos de software educativos expuestos en este informe.

Tabla 2.1 Comparación entre los softwares educativos actuales (elaboración propia)

	SCRATCH	ALICE	GREENFOOT
<b>USO DE TUTORIALES O SIMILARES DENTRO DEL SOFTWARE EDUCATIVO</b>	x	x	x
Entorno 3D	x	✓	x
Entorno 2D	✓	x	✓
<b>EDITOR DE CÓDIGO</b>	x	x	✓
<b>SISTEMA DE ARRASTRAR Y SOLTAR</b>	✓	✓	x
<b>CÓDIGO PREDEFINIDO</b>	✓	✓	✓

## 3 Marco Teórico

Para conocer el ambiente por el cual los alumnos aprenden *POO*, se realizará una investigación de las diferentes formas de enseñar este lenguaje y sus dificultades de aprendizaje. Con el fin de que el software educativo a crear sea acorde a los beneficios que aportan las diferentes formas de enseñar y sin entrar en alguna etapa que pueda dificultar su aprendizaje.

### 3.1 Formas de Enseñar Orientación a Objetos

La enseñanza en los institutos que dan clases de programación comienza con una programación estructurada y luego comienzan a enseñar orientación a objetos, pero a medida que va pasando el tiempo esta forma ha ido cambiando [18]. Algunos institutos comienzan enseñando programación orientada a objetos y se olvidan de enseñar la programación orientada a objetos. A continuación, se presentarán estas nuevas y antiguas formas de enseñar *OO*.

#### 3.1.1 Cursos de *POO*

Al inicio de un curso de programación se enseña los conceptos de *OO* mediante diapositivas, textos, libros, entre otros. Los profesores se enfocan en que el alumno pueda resolver los problemas propuestos. Se mencionarán algunas consideraciones pedagógicas [19].

##### 3.1.1.1 Aprendizaje por Recepción

El docente se encarga de entregarle al alumno toda la información necesaria para que el alumno se interiorice con la materia. Por lo tanto, el alumno solo toma nota de lo que el docente le está exponiendo, lo cual será suficiente para completar su aprendizaje [20].

##### 3.1.1.2 Aprendizaje por Descubrimiento

El estudiante adquiere el conocimiento por sus propios medios, apoyándose en el material entregado por el docente. Bruner [21] considera que el aprendizaje va motivado por la curiosidad. Por lo tanto, el profesor no debe explicar los contenidos demasiado claros, sino que debe proporcionar el material adecuado para estimular al alumno mediante distintas estrategias.

Las ventajas de este aprendizaje ayudan a que el estudiante pueda pensar por sí mismo, estimula la autoestima y la seguridad, da soluciones creativas ante los problemas, aprende como aprender por sí mismo por lo que está más activo [21].

Las desventajas de este aprendizaje es el tiempo entregado para el autoaprendizaje, el cual en la mayoría de los casos es mucho; por parte de los alumnos la motivación inicial no la adquieren; el incentivo para este tipo de aprendizaje es poco; alumnos impulsivos dan respuestas equivocadas con este tipo de aprendizaje [22].

### **3.1.1.3 Aprendizaje Colaborativo**

Los estudiantes se ayudan entre sí, provocando un grupo de estudio, donde cada estudiante es responsable de aprender de sus compañeros y de enseñarles a ellos también. Para llevar a cabo este tipo de aprendizaje se utilizan una variedad de actividades donde el estudiante pueda desarrollar sus habilidades con otros miembros y aprender de las habilidades de los demás. El profesor estará observando a los estudiantes e intervendrá si lo encuentra necesario, entregándole sugerencias o información [23].

Las ventajas de este aprendizaje es aumentar el desarrollo interpersonal, el estudiante se involucra con su propio aprendizaje y contribuye al aprendizaje del grupo, aumenta la autoestima [24].

Las desventajas de este aprendizaje se pueden dar en que el estudiante no se sienta cómodo expresándose frente a otros, esto llevado por su baja autoestima o por el miedo al fracaso, también la falta de experiencia ante este tipo de actividades puede llevar al fracaso de la actividad [25].

### **3.1.1.4 Aprendizaje Basado en Problemas**

Este aprendizaje en la investigación y reflexión que realizan los alumnos para llegar ante la solución del problema planteado por el profesor. Este método propone que los estudiantes sean quienes adquieran el conocimiento y lo aplique en la resolución del problema, sin la necesidad de que el docente tenga que pasar el contenido [26].

Las ventajas de este aprendizaje, permite que el alumno relacione la información nueva con la que ya posee, es versátil, fomenta la autonomía, resulta motivador y ameno, fomenta la habilidad para identificar, analizar y resolver problemas, ejercita la competencia digital [27].

Las desventajas de este aprendizaje; el ritmo de avance es lento y se cubre menor materia, la preferencia de los alumnos es a trabajar individualmente, la preparación adecuada toma mucho tiempo [28].

## **3.1.2 Video Tutoriales**

El término tutorial se utiliza para hacer referencia a un curso breve que indican paso a paso las instrucciones que deben seguirse y ejecutarse para el desarrollo de cualquier actividad. Son más accesibles por medio de internet, la gran mayoría de estos tutoriales se encuentran gratis por esta vía. La realización de estos tutoriales no solo la ejercen los docentes, sino que también los alumnos o cualquier tipo de usuario [29].

Como recurso de ayuda o guía, se mejora y agiliza el proceso de aprendizaje, puesto que los procedimientos pueden seguirse paso a paso y de forma visual [29].

### **3.1.2.1 Como Crear un Video Tutorial**

Los pasos a seguir para realizar un buen video tutorial son los siguientes [30]:

1. Definir los objetivos que se pretender enseñar; viendo su grado de efectividad, viendo como lo va integrar, si es necesario un conocimiento previo.
2. Analizar las tecnologías necesarias para la elaboración del video tutorial.
3. Realizar un guion, preparar un guion ayuda a realizar una grabación más eficaz.
4. Planificar la grabación, preparando el entorno donde se grabará el video.
5. Grabar el video de forma clara y con convicción.
6. Editar el video para que sea más eficaz.
7. Revisar y publicar.

### **3.1.2.2 Requisitos Para su Efectividad**

Para obtener un buen video tutorial se recomienda que cumpla con las siguientes condiciones:

- Breve, entre 5 a 6 minutos.
- Que sea atractivo para el usuario y obtenga la atención de este.
- Comprensible.
- Reforzar el contenido con alguna imagen visual acorde a lo propuesto.

### **3.1.3 Mediante Cuentos**

Una de las formas que se implementan para enseñar orientación a objetos es por medio de cuentos [31]. Permite construir historias donde los personajes son los objetos, las acciones son los métodos, las características son los atributos, dando el modelo de clases [32]. Con este sistema se puede abarcar tres procesos en el desarrollo del software, que son: Análisis orientado a objetos, diseño orientado a objetos y la programación orientada a objetos.

#### **3.1.3.1 Funcionamiento**

El académico presenta el cuento en cuestión como por ejemplo “Los Tres Cerditos”, el alumno identifica los personajes con sus respectivas características y acciones, como se puede apreciar en la Tabla 3.1.

Tabla 3.1 Identificación problemática (elaboración propia)

Personajes	Atributos	Método
Casa	Material, resistencia, color, dueño	-
Cerdos	Personalidad, casa, instrumento	Cantar, bailar, correr, caminar, huir, gritar, reír, burlar, escapar, tocar instrumento
Lobo	Personalidad	Soplar, gritar, hablar, espiar, empujar, engañar, mentir, fingir, reír

Luego se genera una tabla de actividades donde se evaluará; cuando (Evento), quien (Clase) y que (Método); como muestra la Tabla 3.2.

Tabla 3.2 Identificador de actividades (elaboración propia)

Cuando	Quien	Que
Derribar casa	Lobo	Soplar

Con estos datos se podrá crear el diagrama de clases del cuento, como se muestra en la Figura 3.1. Ya para finalizar podremos poner en práctica los elementos identificados por medio de la programación.

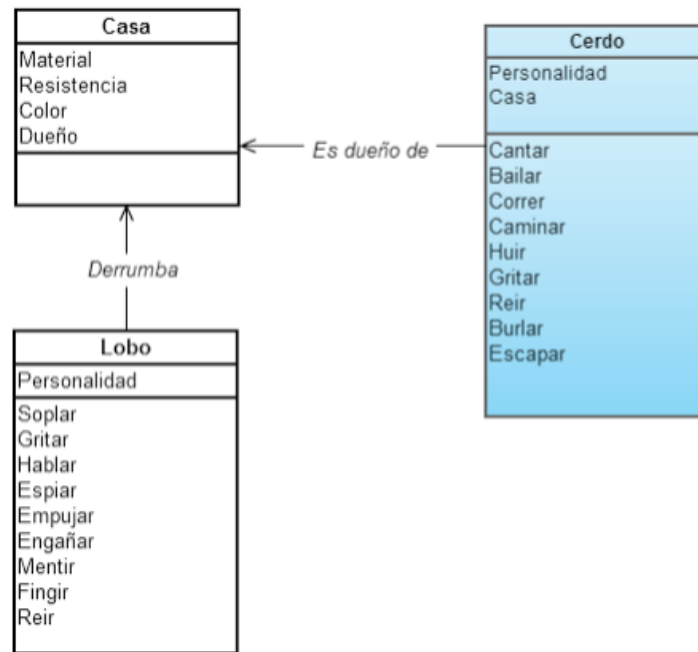


Figura 3.1 Diagrama de clases [31]

### 3.1.3.2 Ventajas

Los alumnos comprenden de mejor manera el funcionamiento del sistema propuesto, dando a conocer sus clases, métodos, atributos, relaciones, entre otros.

Los cuentos sirven para identificar eventos, los estados de los objetos (personajes) y revelan acciones (procesos) importantes en el modelo funcional.

También logran evitar problemas de diseño y de programación [32].

### 3.1.4 Software Educativo

El software educativo está diseñado para educar al usuario, siendo una herramienta pedagógica o de enseñanza que ayuda a la adquisición de conocimientos y al desarrollo de habilidades [33].

Los softwares educativos se pueden dividir en dos, por un lado, los que están diseñados para ser un apoyo para el docente y los que le enseñaran directamente al alumno, le cual puede aprender por su propia cuenta.

Estas herramientas facilitan la enseñanza, mostrando un entorno diferente, en el cual la enseñanza de algún concepto puede ser complicado de exponer. A la vez los alumnos demuestran mayor motivación, ya que se exponen de manera creativa y divertidas los conceptos enseñados.



## 3.2 Dificultades de Enseñar *POO*

Generalmente a la hora de enseñar *POO* los alumnos tienen a complicarse con este concepto, debido a distintos factores [1]. Se mostrará a continuación algunos de estos factores que serán independientes del modo de enseñar el docente.

### 3.2.1 Entorno de Programación

Según estudios el entorno de programación dificulta su aprendizaje debido que estos se encuentran a un idioma diferente al que los alumnos saben [1]. Generalmente estos entornos se encuentran en el idioma inglés, por lo que algunos de los alumnos no llegan a dominarlo completamente por lo que a la hora de que el compilador entregue el mensaje de error, los alumnos que no saben dicho idioma no podrán saber de manera clara el error que están obteniendo [1].

### 3.2.2 Cambio de Paradigma

En algunos institutos que llevan a cabo la programación a objetos, parten estudiando primero con lenguajes de programación estructurados [1], los cuales le ayudan al alumno a comprender los conceptos básicos de la programación, pero cambiar de un lenguaje de programación estructurado a un lenguaje de programación orientado a objetos en muchas ocasiones lleva a la confusión por parte de los alumnos.

Algunas instituciones están dejando de enseñar desde un comienzo la programación estructurada y están comenzando a enseñar *POO* desde el comienzo de la enseñanza. Diciendo que la lógica de la programación al enseñar *POO* directamente va implícita.

## 3.3 Gamificación

La Gamificación es una técnica de aprendizaje que traslada la mecánica de los juegos al ámbito educativo-profesional con el fin de conseguir mejores resultados ya sea para absorber mejor algunos conocimientos, mejorar alguna habilidad, o bien recompensar acciones concretas, entre otros muchos objetivos, lo que hace más fácil la interiorización de conocimientos de una forma más divertida, y así generar una experiencia positiva en el usuario, La idea de la gamificación no es crear un juego, sino valernos de los sistemas de puntuación, recompensas, objetivos que normalmente componen a los mismos. [34].

### 3.3.1 Mecánicas de la Gamificación

Permiten que los alumnos adquieran un compromiso para superar los distintos retos a los que se somete a los alumnos [35], De entre las mecánicas las que más se destacan son las que se puede apreciar en la Figura 3.2.



Figura 3.2 Mecánicas de la Gamificación [34]

### 3.3.2 Dinámicas de Juego de la Gamificación

Las dinámicas tienen como objetivo la motivación del usuario para jugar y seguir adelante en las actividades del juego. En la Figura 3.3 se puede ver las dinámicas destacadas.



Figura 3.3 Dinámicas de juego más destacadas [34]

## 4 Tecnologías

Existen varias plataformas para el desarrollo de video juegos, sin embargo, para este proyecto se necesitará una plataforma capaz de desarrollar juegos en 3D y que a la vez pueda ser desarrollado mediante script. Se consideraron dos tipos de plataformas capaces de desarrollar modelos 3D jMonkey y Unity entre las cuales se decidió utilizar Unity por su documentación y por la experiencia obtenida con anterioridad.

### 4.1 Unity

Unity es un motor de desarrollo para la creación de juegos y contenidos 3D interactivos, soporta varias plataformas las cuales están smartphone, consola o jugador de PC [36]. Contiene un editor de texto lo cual permite la creación de scripting que le las funcionalidades al video juego esto permite a los usuarios a tener más flexibilidad en la creación del juego ya que no están tan limitados en lo que le ofrezca la plataforma; utiliza dos lenguajes de programación para los scripts los cuales son C# y JavaScript, se puede usar Visual Studio o MonoDevelop para la creación del script [37]. A su vez contiene un editor visual muy útil y completo, donde se podrá hacer juegos, tanto en entorno 3D como 2D. Para conseguir recursos para el video juego, cuenta con Asset Store, donde se encontrar elementos de paga o gratuitos.

Unity cuenta con varias versiones están la versión personal que es la gratuita, que solo cuenta con las características básicas del motor; la versión plus y profesional que son de paga [38].

#### 4.1.1 Blender

Como herramienta complementaria a Unity, se utilizará Blender, el cual para este proyecto apoyó en el desarrollo 3D, dando mejoramientos en la animación con su sistema de huesos [39].

Blender es un software libre para diseño y animación 3D muy completo, sus características son [40]:

- Modelado
- Esculpido
- Texturizado
- Texturizado UV
- Pintar UV sobre los modelos
- Materiales
- Sistema de nodos para las texturas y materiales para mayor complejidad y profesionalismo
- Multitextura
- Texturizados aplicados a diversos elementos
- Sistema de huesos
- Sistema de partículas
- Simulador de océanos

- Animación
- Animación no linear
- Desarrollo de juegos en el sistema
- Composición
- Motor de Render
- Edición de video
- Modificadores
- Tracking de cámara
- Una gran cantidad de Addons para aumentar el potencial del programa

## 4.2 Photoshop

Se utilizará Photoshop para la creación del logo y el comic de la historia. Debido que es la herramienta en la que tiene experiencia la técnica de bachillerato en artes, que implementara el comic en el proyecto.

Photoshop es un editor de gráficos. Utilizado principalmente para el retoque de fotografías y gráficos. Traducido al español significa “taller de fotos” [41].

## 4.3 JDK

El JDK se usará como compilador en el entorno 3D para que los estudiantes puedan aprender POO con el lenguaje de programación Java.

JDK traducido al español es, Herramientas de desarrollo para Java. Contiene un conjunto de herramientas (programas y librerías) que permiten desarrollar (compilar, ejecutar, generar documentación, etc.) programas en lenguaje Java [42].

## 4.4 LMMS

LMMS permitirá crear la música ambiental para el video juego.

Linux MultiMedia Studio (LMMS) es una herramienta de creación musical que nos permitirá producir música desde cero, desde una interfaz intuitiva, aunque muy completa y llena de prestaciones, con soporte para teclados MIDI y demás periféricos especializados [43].

## 5 Dominio

El dominio en el que se encontrara el software educativo corresponde al lenguaje de programación en Java. Debido a que es un lenguaje de programación orientado a objetos, es uno de los lenguajes más usados a nivel mundial según TIOBE [44], en la Figura 5.1 se puede ver el ranking de lenguajes de programación más utilizados. Además, en la Universidad donde se implementarán las pruebas del software se utiliza el lenguaje Java como lenguaje para enseñar *POO*, las pruebas se realizarán en la Pontificia Universidad Católica de Valparaíso (PUCV) en la escuela de Ingeniería en informática.

Aug 2017	Aug 2016	Change	Programming Language	Ratings	Change
1	1		Java	12.961%	-6.05%
2	2		C	6.477%	-4.83%
3	3		C++	5.550%	-0.25%
4	4		C#	4.195%	-0.71%
5	5		Python	3.692%	-0.71%
6	8	▲	Visual Basic .NET	2.569%	+0.05%
7	6	▼	PHP	2.293%	-0.88%
8	7	▼	JavaScript	2.098%	-0.61%
9	9		Perl	1.995%	-0.52%
10	12	▲	Ruby	1.965%	-0.31%

Figura 5.1 Ranking de los lenguajes de programación más usados.

### 5.1 Temas de Aprendizaje

Se pretende enseñar los siguientes conceptos: Estructura, atributos, arreglos, métodos y Objetos. No se abordarán más temas de *POO* debido a que el tiempo acordado para el desarrollo del software es corto, por lo que no se alcanza a desarrollar más niveles que aborden el tema de *OO*.

## 6 Problemática Propuesta

Se desarrollará un software educativo que busca reforzar y enseñar lenguaje de programación en Java.

Sera un juego en el entorno 3D, el cual consiste en una historia en donde al usuario se le dará una problemática concorde a la historia y deberá ser respondida mediante código de programación en el lenguaje Java, al resolver de manera correcta el ejercicio, el jugador podrá ver en tiempo real la ejecución del código en los personajes asignados en el entorno 3D.

Cada nivel reforzara y/o enseñara programación en lenguaje Java partiendo desde lo más básico hasta llegar a instanciar un objeto.

Las problemáticas serán de construir, comprender y completar código en java dependiendo del nivel en que se encuentre el jugador.

El sistema también contara con tutoriales capaces de enseñar y/o reforzar al usuario programación en el lenguaje Java.

Por lo tanto, este software permitirá que el usuario pueda aprender y/o reforzar el lenguaje de programación java de manera interactiva.

### 6.1 Historia

Esta historia contara con 4 personajes, donde su principal personaje es el insecto denominado chinita. Esto personajes están compuesto por:

- Científico (hombre)
- Asistente del científico (mujer)
- Monstruo
- Chinita que será la guía del video juego (Ashy)

La historia comienza con la obsesión de un científico de crear al ser humano perfecto. Sin embargo, en su experimentación el científico en vez de crear lo propuesto, crea a un monstruo verde.

Al comienzo de la historia podremos apreciar como el científico está experimentando y sufre un accidente en el cual confunde los químicos y crea un monstruo por error, nuestra protagonista se encuentra en el laboratorio y observa lo acontecido. El científico en vista de que creo algo semejante, busca destruir al monstruo, pero en esa búsqueda se da cuenta de que no es posible la eliminación de este, por lo que consigue en ese intento dormir al monstruo, por lo cual lo logra encerrar. Posteriormente el científico se concentrará en crear una bala que pueda destruir al monstruo, cuando por fin lo logra decide ir a buscar el arma para poder hacer uso de la bala. En este transcurso llega la asistente del científico y descubre al monstruo encerrado, la asistente asustada grita de terror, por lo que el monstruo despierta y logra escapar. La asistente consigue esconderse del monstruo por lo que este se escapa hacia la ciudad. La chinita que siempre se encontró en el recinto observando los hechos, decide ayudar a la asistente indicándole

lo que debe realizar y es así como la chinita en el transcurso del juego ayudara a los personajes (hombre, mujer, monstruo) a realizar sus respectivas misiones.

El objetivo del monstruo será encontrar el recipiente con la fórmula para poder multiplicarse y así crear a los mini monstruos. También tendrá que asesinar a cualquier persona que se encuentre en su camino, en la Tabla 6.1 se muestran las acciones que realizara el monstruo.

Las misiones de la mujer están involucradas con encontrar la bala en el laboratorio y luego salir buscando al científico por la ciudad, durante ese transcurso se tendrá que ir escondiendo de los mini monstruos y del monstruo, en la Tabla 6.1 se muestran las acciones que realizara la mujer.

Para el hombre sus misiones serán ir a buscar la bala, durante el transcurso del camino que tiene que recorrer desde donde consiguió el arma hasta ir a buscar la bala, se encontrara con mini monstruos, por lo que, el hombre tendrá que destruirlos con una simple arma, ya que estos mini monstruos no son indestructibles como el monstruo original. Al llegar a su laboratorio se dará cuenta de que la bala no está, por lo que, tiene que comenzar a buscar a la mujer quien es la responsable de tener la bala, en la Tabla 6.1 se muestran las acciones que realizara el hombre.

Tabla 6.1 Acciones que realizaran los personajes (elaboración propia)

Monstruo	Mujer	Hombre
Encontrar el recipiente con fórmula para multiplicarse	Encontrar bala especial	Buscar bala
Crear mini monstruos	Buscar a científico (hombre)	Matar monstruos

Las misiones del hombre y la de la mujer terminaran cuando encuentre se encuentren y luego el usuario podrá jugar de la manera tradicional sin código de por medio.

## 6.2 Niveles

El software educativo contara con sistema de niveles los cuales presentaran los distintos conceptos de *POO*, por medio de misiones.

### 6.2.1 Introducción

Para el nivel de introducción se contará la historia propuesta anteriormente para que el usuario pueda identificarse con la historia, los personajes y las acciones de cada personaje. Utilizando así la forma de enseñar mediante cuentos.

La historia se contará por medio del relato comic, en la Figura 6.1, se puede observar un ejemplo de cómo será el comic dentro del video juego.



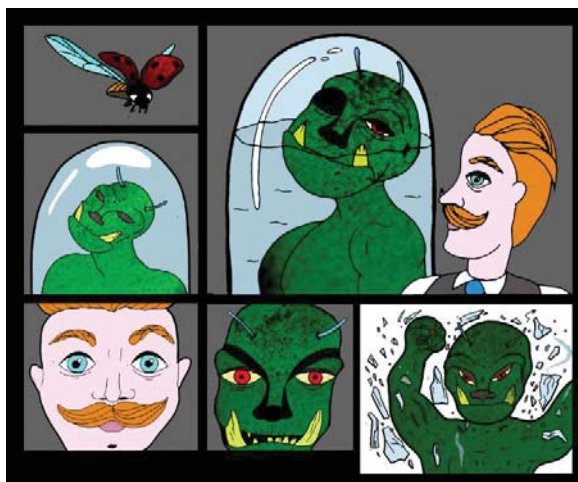


Figura 6.1 Demostración del comic [45]

## 6.2.2 Nivel 1

En el nivel 1 se le pedirá al usuario que cree la clase principal, la cual el nombre de esta será el nombre de la ciudad del mundo en 3D. En caso de que el usuario no sepa lo que es una clase tendrá un tutorial donde le explicara los contenidos de esta. Por lo tanto, aquí tenemos un tipo de aprendizaje basado en problemas. También se puede encontrar una forma de enseñar por tutoriales, no es por medio de video tutoriales, pero sigue la dinámica de estos.

Una vez desarrollado la tarea asignada se mostrará la ciudad del mundo en 3D, con el nombre propuesto por el usuario.

## 6.2.3 Nivel 2

En el nivel 2 se comenzará mostrando a los personajes (hombre, mujer, monstruo), para que el usuario pueda elegir que personaje utilizara durante el transcurso del juego. Esto hará que el usuario pueda tener más de una opción de juego. En la figura 6.2, se muestra una captura de la elección de los personajes.

Luego al igual que el nivel 1 contara con la problemática asignada para este nivel y para el personaje elegido. La misión de este nivel tendrá que ver con los atributos de la clase del personaje.

Finalmente, el usuario podrá ver en tiempo real la ejecución del programa en el personaje.

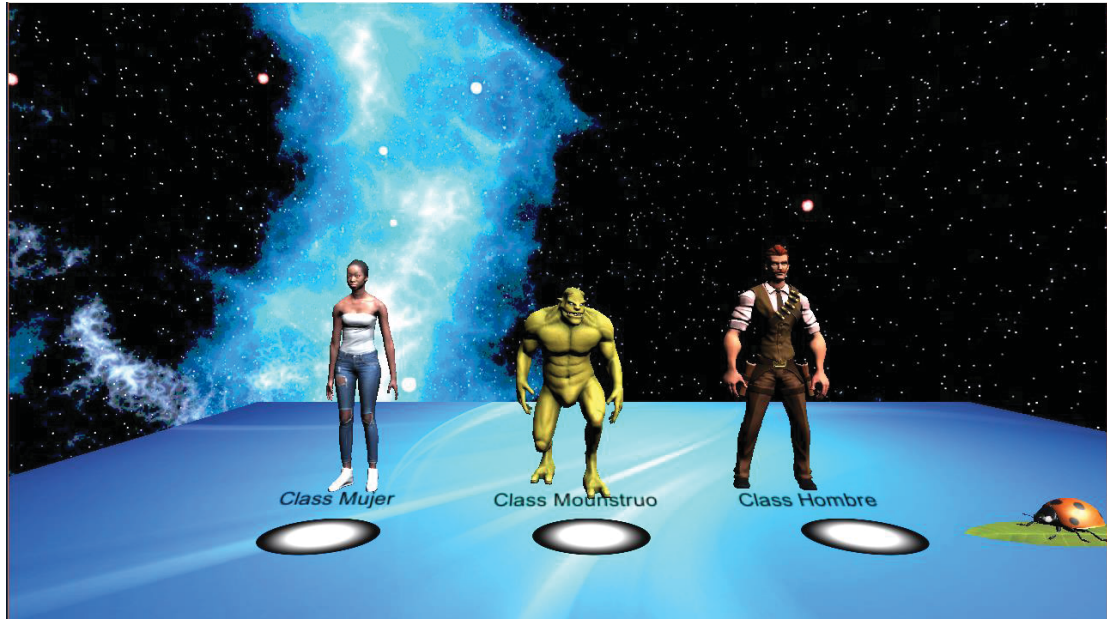


Figura 6.2 Demostración elección de personajes (elaboración propia)

#### 6.2.4 Nivel 3

Al igual que el nivel 1 y 2 contara con la problemática asignada para este nivel y para el personaje elegido. La misión de este nivel tendrá que ver con arreglos.

También contara con un sistema de tutoría para el aprendizaje del usuario.

Luego el usuario podrá ver en tiempo real la ejecución del programa en el personaje.

#### 6.2.5 Nivel 4

La problemática de este nivel será con respecto a los métodos que tendrá el personaje. También contara con un sistema de tutoría para el aprendizaje del usuario.

En esta misión no se verá en tiempo real la ejecución del problema debido que los métodos serán ejecutados una vez se vea como se instancia un objeto.

#### 6.2.6 Nivel 5

Aquí se verá como instanciar el objeto por lo que aquí ya se podrá entrar a fondo lo que es clases y objetos.

Contará con un sistema de tutoría el cual enseñará la diferenciación como muestra la Figura 6.3 y la Figura 6.4

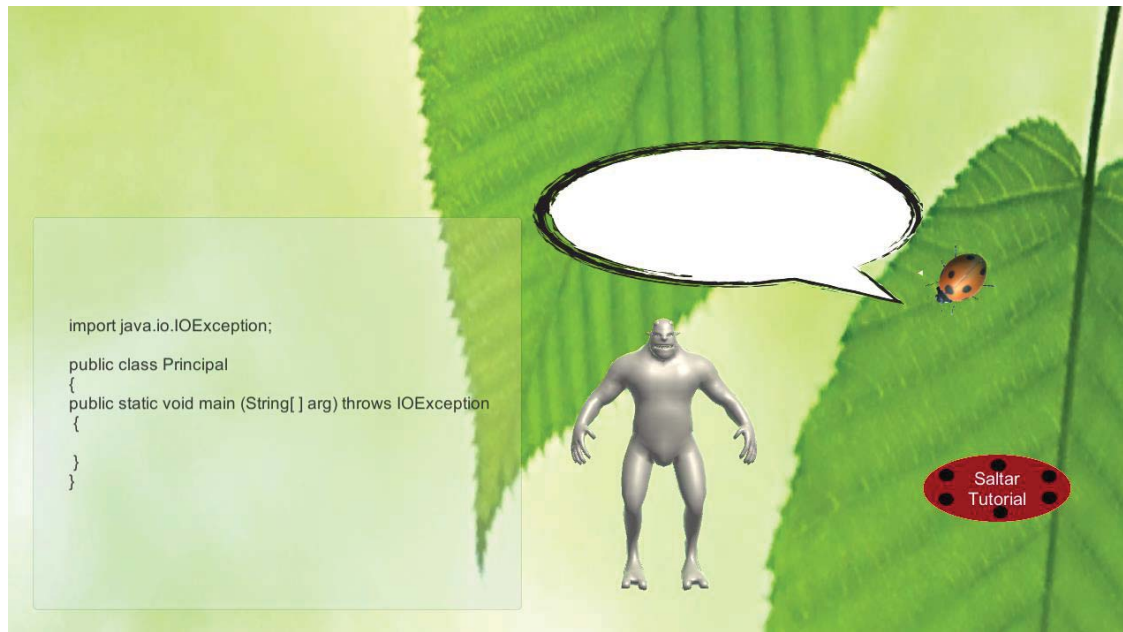


Figura 6.3 Prototipo de tutorial para objetos (elaboración propia)

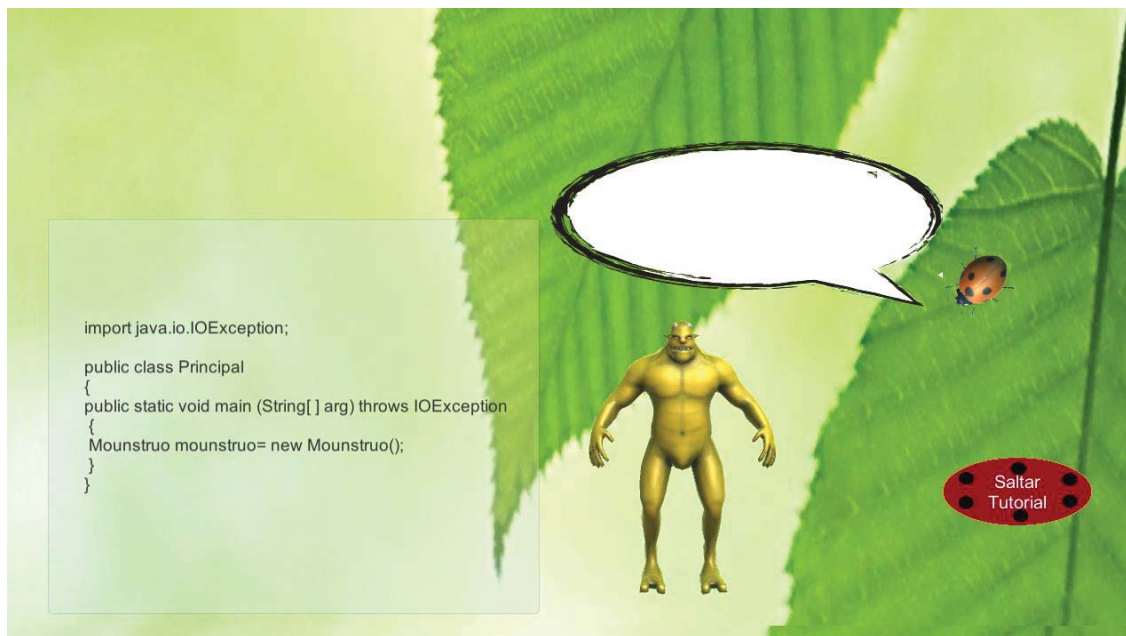


Figura 6.4 Prototipo tutorial para objetos 2 (elaboración propia)

## 6.2.7 Nivel 6

Aquí se mostrará el juego finalizado por lo que el usuario en esta instancia podrá jugar sin la necesidad de utilizar código, sino que con las configuraciones que el mismo le asigno durante el transcurso del juego.

### 6.2.7.1 Reglas del Juego

Para hacer el video juego mas interesantes, existen las siguientes reglas:

- Para que la mujer gane el juego tendrá que encontrar al hombre, evitando ser asesinada por el monstruo o los mini monstruos.
- Para que el hombre gane el juego tendrá que matar al monstruo con la bala especial.
- Para que el monstruo gane el juego tendrá que matar a la mujer antes de que se encuentre con el hombre.
- Si el personaje elegido es el hombre o la mujer y este es asesinado por un monstruo o mini monstruo se reiniciará el nivel 6.
- Si el personaje elegido es el monstruo y este es asesinado por el hombre, se reiniciará el juego en el nivel 6.
- Si se reinicia el nivel 6, el sistema de puntajes se vuelve a 0.

## 6.3 Puntaje

Cada nivel contara con un sistema de puntajes que servirán cuando el usuario llegue al nivel 6, en dicho nivel el usuario quedara más cerca de su próxima misión dependiente de cuando puntaje haya acumulado. Ejemplo: si el usuario eligió al hombre y el primer objetivo de este es encontrar a la mujer, este partirá a unas pocas cuadras de donde se encuentra la mujer si es que acumulo muchos puntos de lo contrario si acumulo poco se encontrara más lejos de donde está la mujer.

La forma de acumular el puntaje es mediante la realización del código, cada vez que salga un mensaje de error se le ira descontando puntos. Por cada nivel se le dará un puntaje de 1000 puntos, si realiza la tarea de manera correcta acumulara los mil puntos, pero en caso de que le salgan mensajes de errores acumulara los 1000 menos la cantidad de errores. Por lo tanto, el puntaje máximo que se puede conseguir para el nivel 6 son 5000 puntos. Se utiliza la acumulación de puntaje para darle gamificación al videojuego.

## 6.4 Requerimientos

Los requerimientos del software educativo presentado se encuentran a continuación como requerimientos funcionales y no funcionales.

### 6.4.1 Requerimientos Funcionales

A continuación, se mostrarán los requerimientos funcionales del software educativo.

Tabla 6.6.2 Requerimiento funcional 1

<b>ID: RF-01</b>
<b>Título: Historia</b>
<b>Descripción:</b>
<p>Se necesita que el usuario conozca la historia, mediante una explicación visual y posiblemente audible. Se debe incluir:</p> <ul style="list-style-type: none"><li>- Narración de la historia.</li><li>- Imágenes de los personajes interactuando en la historia.</li></ul>
<b>de Negocio</b>
<b>1. Las imágenes tienen que ir acorde con la narración de la historia.</b>
<b>cciones:</b>
<ul style="list-style-type: none"><li>- El usuario no podrá saltarse la narración de la historia debido que es el fundamento del videojuego.</li></ul>
<b>Interacciones</b>
<p>Es el comienzo del videojuego por lo que no lleva interacciones con otros requerimientos.</p>

Tabla 6.6.3 Requerimiento funcional 2

<b>ID: RF-02</b>
<b>Título: Misiones</b>
<b>Descripción:</b>
<p>El juego contara con un sistema de misiones, le cual le indicara al usuario lo que tiene que hacer en dicho nivel. Para esto debe contar con:</p> <ul style="list-style-type: none"> <li>- Una redacción clara, para que el usuario pueda entender de manera coherente las misiones que tiene que realizar.</li> <li>- Si el usuario tiene dudas sobre la resolución del problema, contara con un sistema de tutoriales que le enseñaran al usuario la materia que se esta llevando a cabo en esa misión.</li> </ul>
<b>de Negocio</b>
1. Coherencia con la historia del juego.
<b>cciones:</b>
<ul style="list-style-type: none"> <li>- El usuario tiene que resolver correctamente la misión asignada para ir al siguiente nivel.</li> </ul>
<b>Interacciones</b>
<b>Este requerimiento va ligado al requerimiento RF-01, que permite unir la historia con las misiones para el usuario.</b>

Tabla 6.6.4 Requerimiento funcional 3

<b>ID: RF-03</b>
<b>Título: Interpretación del código en el lenguaje de programación Java</b>
<b>Descripción:</b>
<p>Se necesita que el videojuego pueda compilar en lenguaje Java, ya que el juego está desarrollado para que los estudiantes comprendan la orientación a objetos mediante el lenguaje Java. Para esto se debe incluir:</p> <ul style="list-style-type: none"> <li>- JDK se implante en el videojuego.</li> <li>- La interpretación sea la adecuada a las instrucciones indicadas en el videojuego.</li> </ul>
<b>de Negocio</b>
1. La interpretación sea acorde a la misión asignada para el usuario.
<b>cciones:</b>
<ul style="list-style-type: none"> <li>- El usuario tiene que resolver correctamente la misión asignada para ir al siguiente nivel.</li> <li>- El usuario debe realizar el código de la manera que se es indicado, de lo contrario no podrá pasar de nivel.</li> </ul>
<b>Interacciones</b>

**Este requerimiento va ligado al requerimiento RF-02, que permite que el usuario sepa cómo resolver el problema propuesto.**


Tabla 6.6.5 Requerimiento funcional 4

<b>ID: RF-04</b>
<b>Título: Entorno 3D</b>
<b>Descripción:</b>
<p>El videojuego estará desarrollado en un entorno 3D, esto facilitará la comprensión orientado a objetos para el usuario, dejando en evidencia los tipos de objetos a utilizar en el entorno 3D. Para esto se necesita:</p> <ul style="list-style-type: none"> <li>- Que cada personaje sea personalizado en el entorno 3D.</li> <li>- Tener un mundo donde los personajes puedan interactuar entre si.</li> </ul>
<b>de Negocio</b>
1. El mundo sea acorde a la historia.
<b>cciones:</b>
<ul style="list-style-type: none"> <li>- El usuario podrá interactuar con el mundo 3D a medida que vaya avanzando en las misiones.</li> </ul>
<b>Interacciones</b>
<b>Este requerimiento va ligado al requerimiento RF-01, RF-02,RF-03, ya que es el entorno en que está enfocado el juego.</b>

Tabla 6.6.6 Requerimiento funcional 5

<b>ID: RF-05</b>
<b>Título: Personajes</b>
<b>Descripción:</b>
<p>El videojuego contara con personajes que harán real la historia, estos serán 3 personajes, los cuales tendrán distintos propósitos dentro de la historia. Para realizar los personajes se necesitará:</p> <ul style="list-style-type: none"> <li>- Los personajes serán una mujer, un hombre y un monstruo.</li> <li>- Para el personaje de la mujer esta realizara las siguientes acciones: Encontrar bala, correr, esconderse, encontrarse con el hombre.</li> <li>- Para el personaje del hombre este realizara las siguientes acciones: Buscar bala, disparar, buscar mujer, matar monstruo.</li> <li>- Para el personaje del monstruo este realizara las siguientes acciones: Matar personas, clonarse, huir.</li> </ul>
<b>de Negocio</b>
1. Dependiendo del personaje que elija podrá utilizar las funciones antes mencionadas.



cciones:
<ul style="list-style-type: none"> <li>- El usuario solo podrá elegir uno de estos personajes para utilizar durante el juego.</li> <li>- Cada acción que realice el personaje será contada como un método dentro del juego.</li> </ul>
Interacciones
Este requerimiento va ligado al requerimiento RF-04, RF-03, RF-02, RF-01 debido a que cada una de estas interacciones definirán la jugabilidad del personaje.
Anexos


## 6.4.2 Requerimientos No Funcionales

A continuación, se mostrarán los requerimientos funcionales del software educativo.

Tabla 6.6.7 Requerimiento no funcional 1

<b>ID: RNF-01</b>
<b>Título: Audio</b>
Descripción:
<p>Se incluirá audios para los niveles del juego, para que el usuario este más activo dentro de este. Para esto se debe incluir:</p> <ul style="list-style-type: none"> <li>-Música ambiental por cada nivel.</li> <li>-Las explicaciones serán explicadas por una voz en off dentro del juego.</li> </ul>
de Negocio



cciones:
<ul style="list-style-type: none"> <li>- Si hay una explicación en ejecución la música ambiental será más baja para que se pueda escuchar de forma clara la dicha explicación.</li> </ul>

Tabla 6.6.8 Requerimiento no funcional 2

<b>ID: RNF-02</b>
<b>Título: Mensajes</b>
<b>Descripción:</b>
<p>Cada etapa que requiera completar código contara con mensajes de aprobación o rechazo, indicándole al usuario que lo hizo correctamente o se equivocó. Para esto se debe considerar:</p> <ul style="list-style-type: none"> <li>- El mensaje de error dirá cuál es el motivo de su equivocación, puede ser de compilación, realizo una acción que no era la indicada o le faltaron cosas por completar.</li> <li>- Cada mensaje de aprobación será incentivador para el usuario.</li> </ul>
<b>de Negocio</b>
1. Los mensajes de errores dependerán del nivel en el cual se encuentra el usuario.
cciones:
<ul style="list-style-type: none"> <li>- Siempre tendrá que mostrarse algún mensaje, para que el usuario sepa en que condiciones esta.</li> </ul>
<b>Interacciones</b>
<b>Este requerimiento va ligado al requerimiento RF-02, permitiendo saber cual es el mensaje correspondiente.</b>

## 6.5 Evaluación

El modo de evaluación será mediante un pre test que evalué los conocimientos que tiene el estudiante antes de probar Ashy.AIRescate(). Posteriormente el estudiante probara el software, en este proceso se dividirá al curso en dos partes, el primer grupo de curso probara el software sin la puntuación, música, mensajes positivos y el otro grupo de curso lo probara con puntuación, música, mensajes positivos. Terminado el proceso de prueba del software se hará una prueba post test que permita evaluar a los alumnos y medir la diferencia de conocimientos tanto en el pre test como en la diferenciación de los alumnos que probaron el software con y sin puntuación, música, mensajes positivos.

## 7 Diseño

Se presentan los siguientes diagramas para la comprensión del desarrollo del video juego: diagrama de actividad, diagrama de clases, casos de uso.

### 7.1 Diagrama de Actividad

En el presente diagrama de actividad se representa las actividades que puede realizar el usuario dentro del video juego.

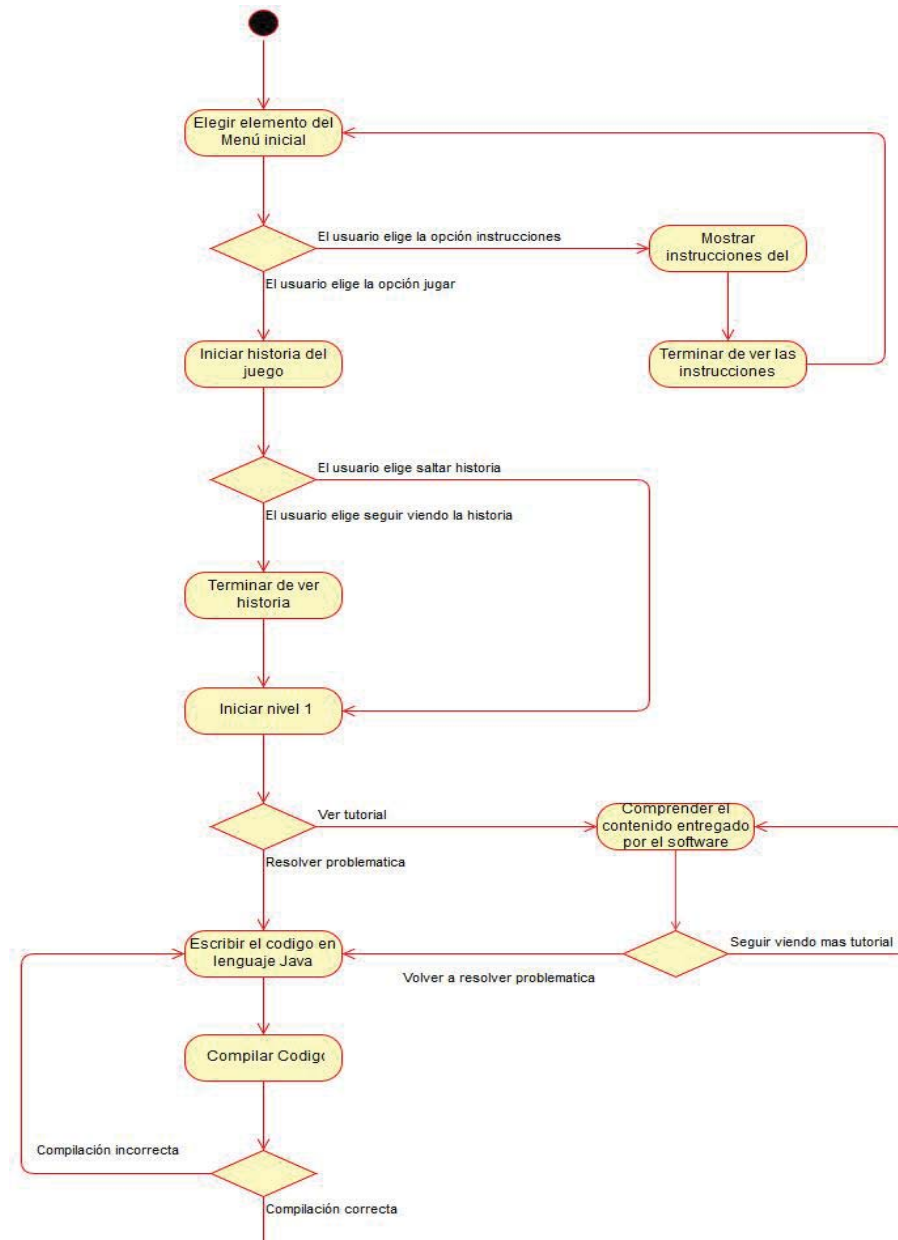


Figura 7.1 Inicio del diagrama de actividad

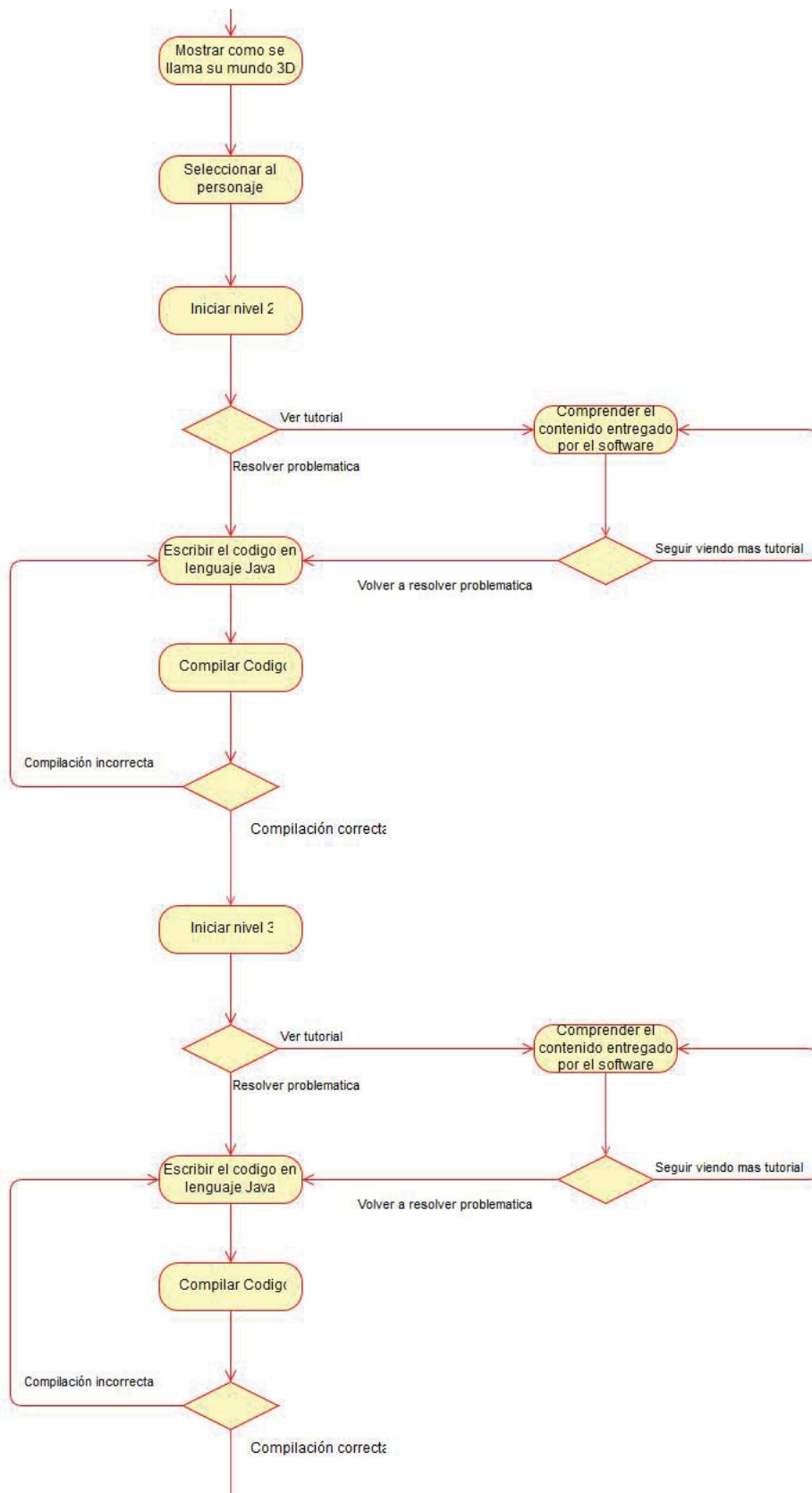


Figura 7.2 Continuación del diagrama de flujo parte 1

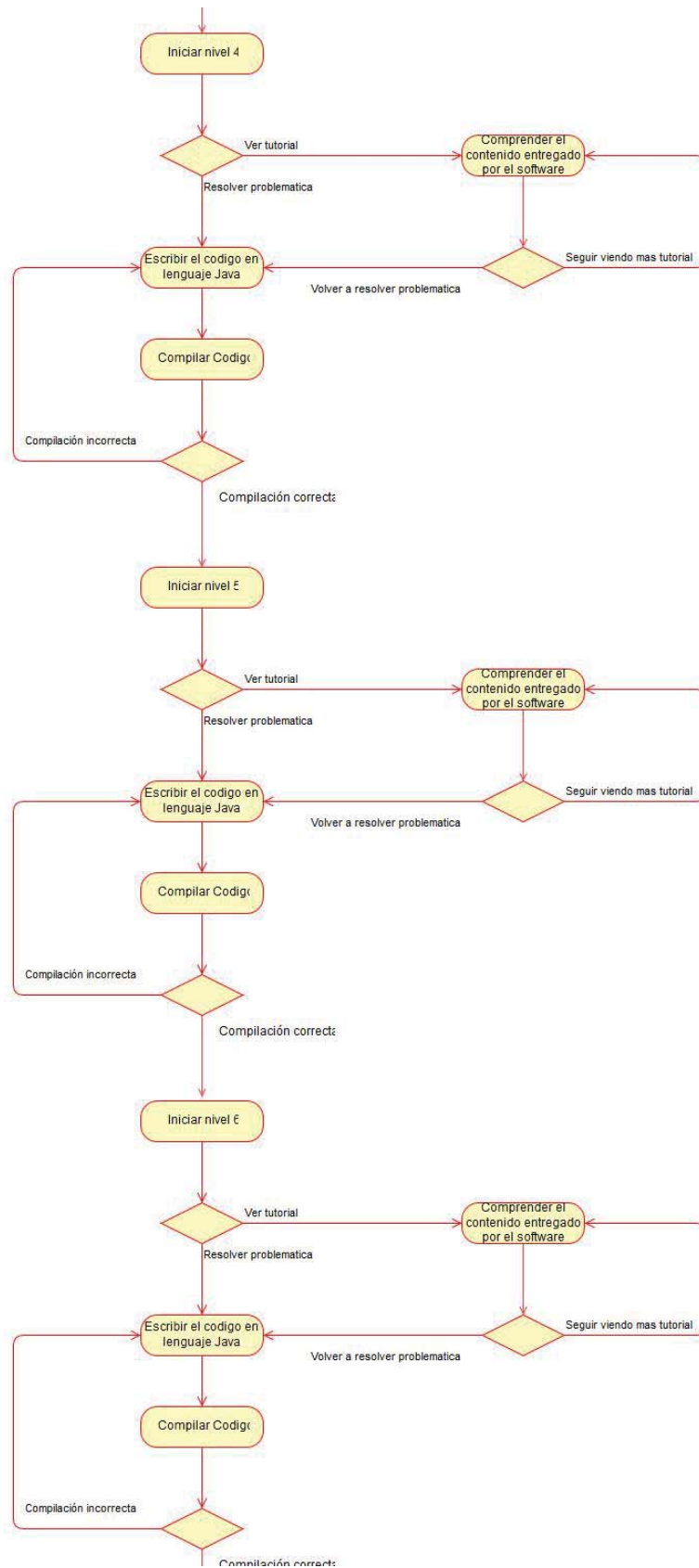


Figura 7.3 Continuación del diagrama de flujo parte 2

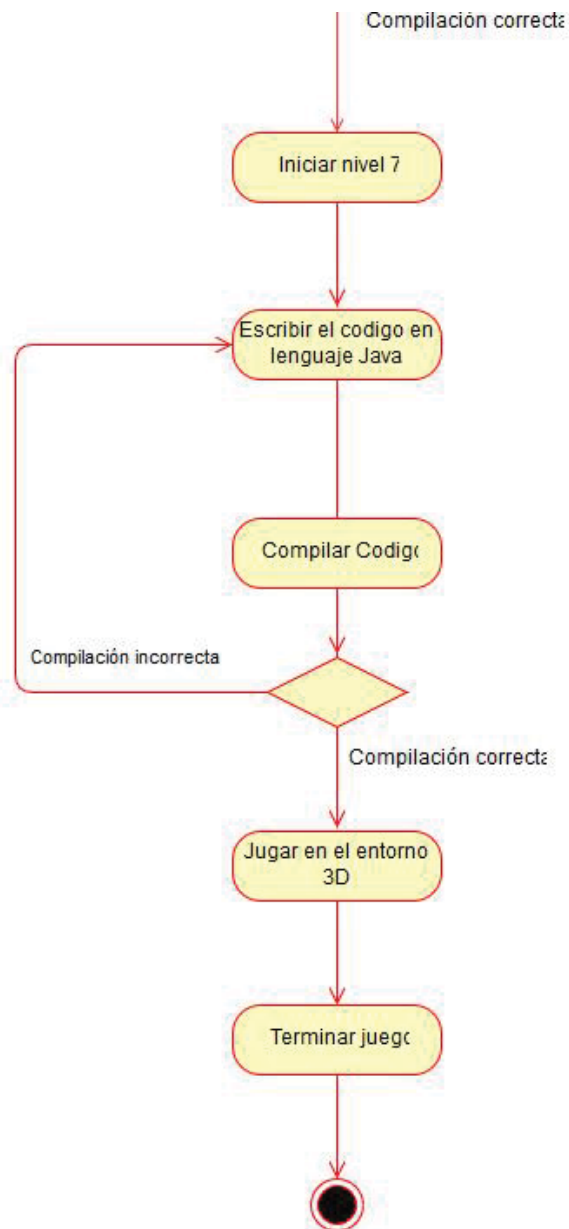


Figura 7.4 Finalización del diagrama de actividad.

## 7.2 Diagrama de Clases

A continuación, se presenta el diagrama de clases el cual incluye todas las clases creadas en el proyecto junto con sus asociaciones.

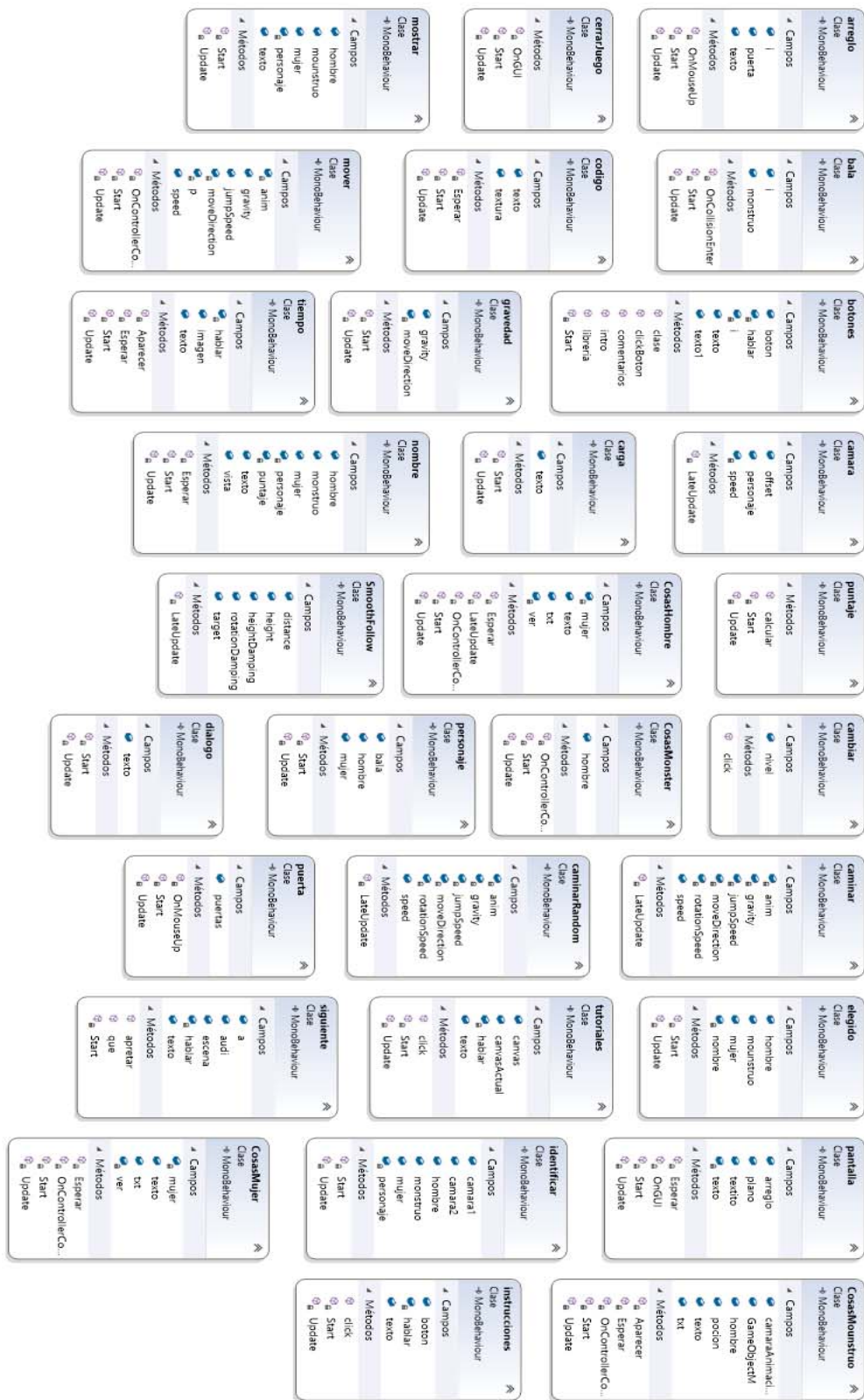


Figura 7.5 Clases que no contienen asociaciones



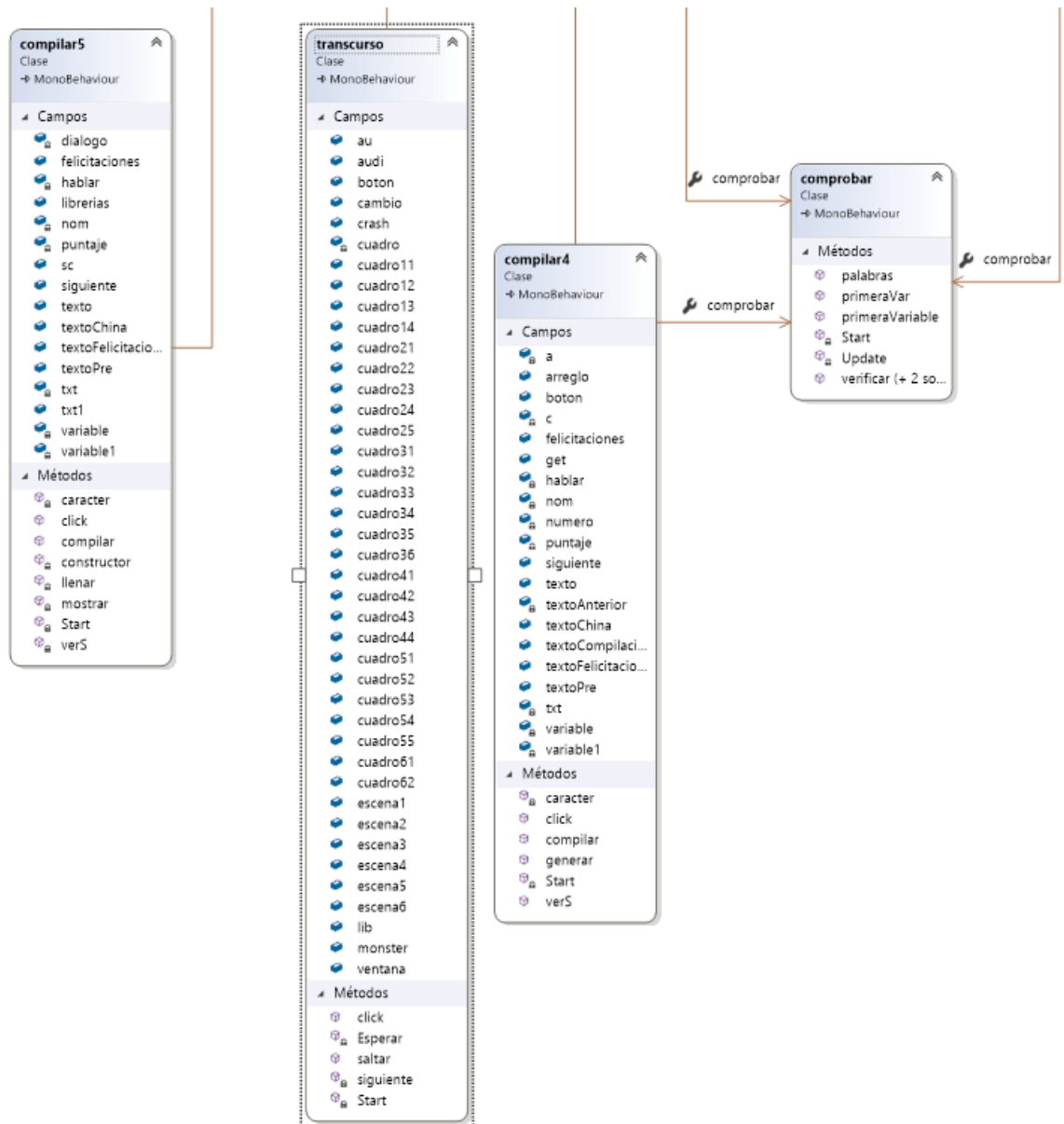


Figura 7.7 Diagrama de clases con sus asociaciones parte 2



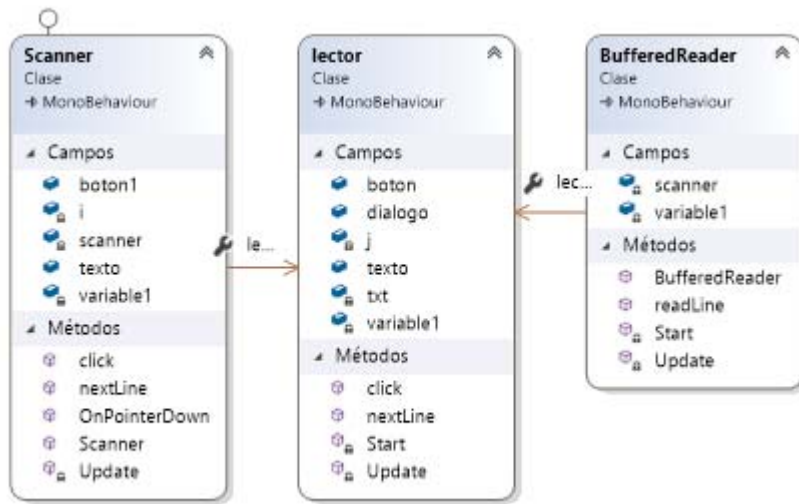


Figura 7.8 Diagrama de clases junto con sus asociaciones

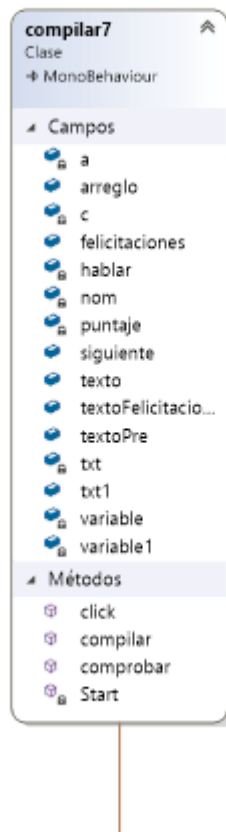


Figura 7.9 Diagrama de clases junto con sus asociaciones parte 1

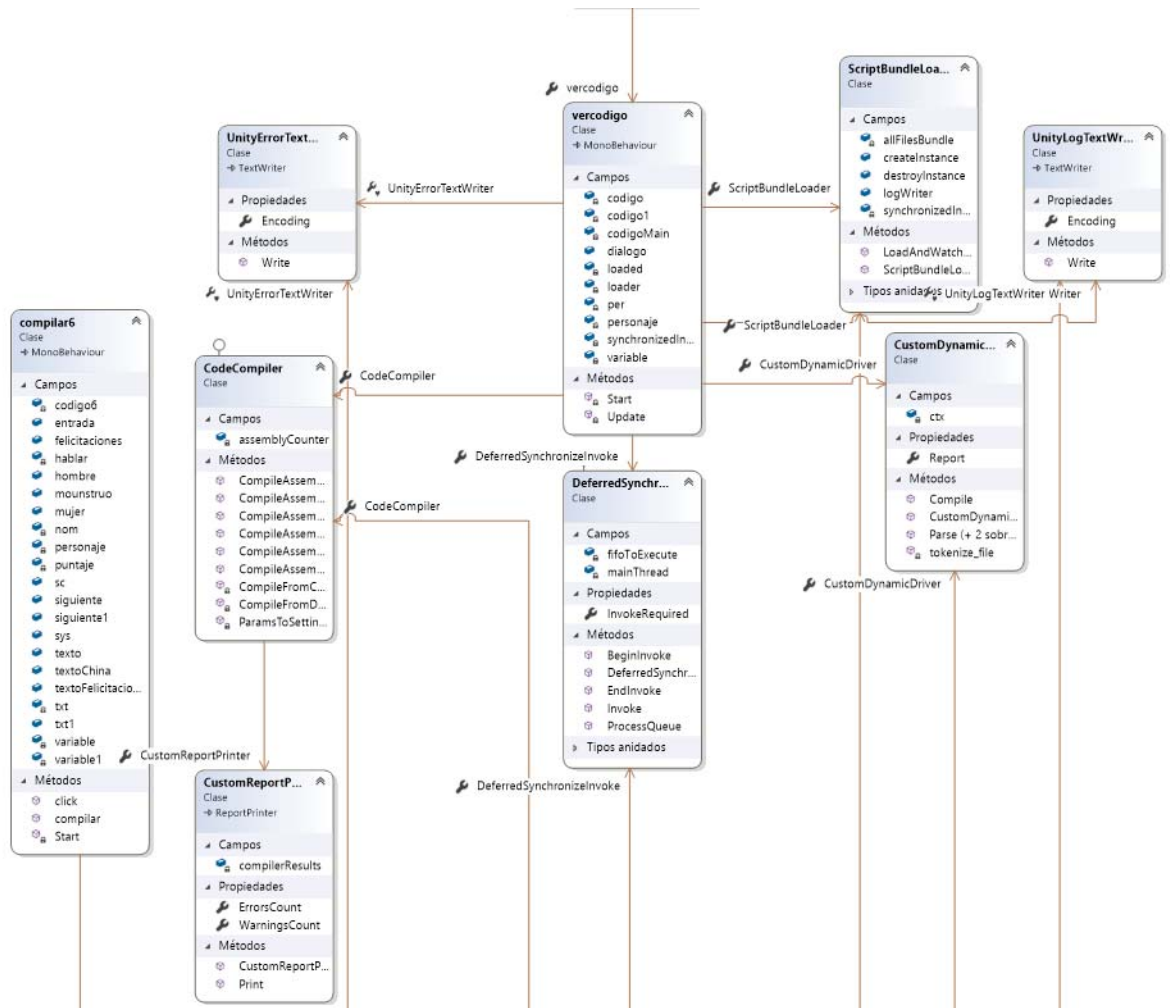


Figura 7.10 Diagrama de clases junto con sus asociaciones parte 2

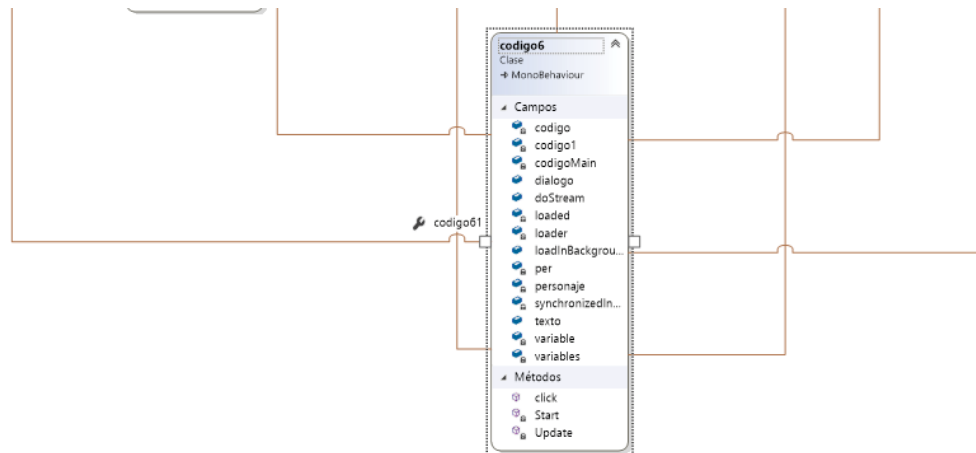


Figura 7.11 Diagrama de clases junto con sus asociaciones parte 3

### 7.3 Diagrama Casos de Uso

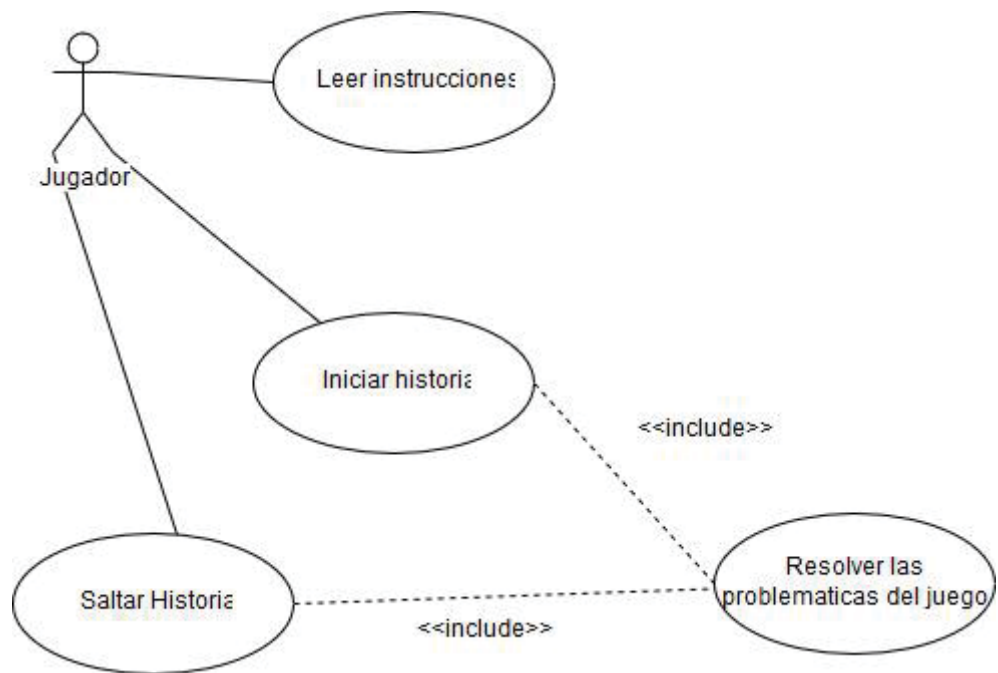


Figura 7.12 Diagrama de casos de uso

## 8 Pruebas de usuario

Se aplicaron pruebas de usuario a alumnos que cursan *POO*, cursos superiores y a alumnos que están cursando la asignatura anterior de *POO* en la universidad Pontificia Universidad Católica de Valparaíso (PUCV).

### 8.1 PreTest

Las preguntas utilizadas en el PreTest fueron las siguientes:

1-¿Cuál es la descripción que crees que define mejor el concepto 'clase' en la programación orientada a objetos?

- ☐ Es un concepto similar al de 'array'
- ☐ Es un tipo particular de variable
- ☐ Es un modelo o plantilla a partir de la cual creamos objetos
- ☐ Es una categoría de datos ordenada secuencialmente

2-Los tipos de modificadores de visibilidad en Java son

- ☐ Publico, Privado y mixto
- ☐ public,private,protected y por default
- ☐ import, package, java.util
- ☐ public, private, protected

3-¿Qué elementos crees que definen a un objeto?

- ☐ Sus cardinalidad y su tipo
- ☐ Sus atributos y sus métodos
- ☐ La forma en que establece comunicación e intercambia mensajes
- ☐ Su interfaz y los eventos asociados

Figura 8.1 PreTest de la pregunta 1 a la 3

4-¿Qué significa instanciar una clase?

- ☐ Duplicar una clase
- ☐ Eliminar una clase
- ☐ Crear un objeto a partir de la clase
- ☐ Conectar dos clases entre sí

5-Desde una clase externa, la llamada del método "saludar()", que posee el objeto "persona" se realiza de la forma:

- ☐ saludar();
- ☐ persona->saludar();
- ☐ persona.saludar();
- ☐ ninguna de las anteriores

6-Acción u operación que realiza un objeto

- ☐ Clase
- ☐ Herencia
- ☐ Metodo
- ☐ Casting

Figura 8.2 PreTest de la pregunta 4 a la 6

7-¿Cual es la forma correcta de declarar un vector?

- ☐ `int vector [] = new int [10];`
- ☐ `int vector [] = new int [10]`
- ☐ `int vector [][] = new int [5][5];`
- ☐ `int vector = new int [10];`

8-Algunos ejemplos de nombres correctos para objetos son:

- ☐ `Reloj, ObtenerHora, Fecha`
- ☐ `MiClas, extens, int b`
- ☐ `Lápiz, Pez, Reloj`
- ☐ `abstract,assert,Hola`

9-Java soporta los siguientes tipos de datos primitivos

- ☐ `Entero, punto flotante, logico, caracter`
- ☐ `Entero,punto flotante, caracter`
- ☐ `Númeroico, carácter, logico`
- ☐ `Numerico, punto flotante, logico, string`

Figura 8.3 PreTest de la pregunta 7 a la 9

10-Una clase siempre tiene un constructor por default

- ☐ Si
- ☐ No

11-Una librería nos permite siempre

- ☐ Nombrar con mas facilidad una clase
- ☐ Utilizar métodos , clases y atributos predefinidos
- ☐ Manejar excepciones
- ☐ Importar clases del mismo package

12-¿Al crear un nuevo arreglo en java este tiene valores por defecto?

- ☐ Si y depende del tipo de dato del arreglo
- ☐ No, contiene contenido basura

Figura 8.4 PreTest de la pregunta 10 a la 12

13-Al crear un objeto a partir de la clase se esta:

- ☐ Instanciando la clase
- ☐ Agregando librerias a una clase
- ☐ Haciendo sobrecarga de metodos
- ☐ Creando una interfaz de usuario

14-La sintaxis " String[ ] nombre = new String[4]; " corresponde a

- ☐ Sobre carga de un metodo
- ☐ La sintaxis para declarar e inicializar un arreglo de String
- ☐ La sintaxis para declarar e inicializar un arreglo de enteros
- ☐ Un error de sintaxis

Figura 8.5 PreTest de la pregunta 13 a la 14

15- La definición "Métodos en una clase que tienen el mismo nombre pero con diferentes argumentos." corresponde a

- ☐ Error de compilacion
- ☐ Excepcion
- ☐ SobrEscritura de metodos
- ☐ SobreCarga de metodos

Figura 8.6 PreTest de la pregunta 15

A continuación, se presenta una tabla resumen donde se logra identificar la categoría de cada pregunta junto con su respuesta correcta.



Tabla 8.1 Tabla de identificación de contenidos del PreTest

Numero	Temática	Pregunta	Respuesta
1	Clase	1-¿Cuál es la descripción que crees que define mejor el concepto 'clase' en la programación orientada a objetos?	Es un modelo o plantilla a partir de la cual creamos objetos
2	Atributos	2-Los tipos de modificadores de visibilidad en Java son	public,private, protected, por default
3	Objetos	3-¿Qué elementos crees que definen a un objeto?	Sus atributos y sus métodos
4	Objetos	4-¿Qué significa instanciar una clase?	Crear un objeto a partir de la clase
5	Métodos	5-La llamada del método saludar(), que posee el objeto persona se realiza de la forma:	Persona.Saludar();
6	Método	6-Acción u operación que realiza un objeto	Método
7	Arreglos	7-¿Cual es la forma correcta de declarar un vector?	int vector = new int [10];
8	Atributos	8-Algunos ejemplos de nombres correctos para objetos son:	Lápiz, Pez, Reloj
9	Atributos	9-Java soporta los siguientes tipos de datos primitivos	Entero, punto flotante, lógico, carácter
10	Clase	10-Una clase siempre tiene un constructor por default	No
11	Clase	11-Una librería nos permite siempre	Utilizar métodos , clases y atributos predefinidos

12	Arreglos	12-¿Al crear un nuevo arreglo en java este tiene valores por defecto?	Si y depende del tipo de dato del arreglo
13	Objetos	13-Al crear un objeto a partir de la clase se está :	Instanciando la clase
14	Arreglos	14-La sintaxis " String[ ] nombre = new String[4]; " corresponde a	La sintaxis para declarar e inicializar un arreglo de String
15	Método	15- La definición "Métodos en una clase que tienen el mismo nombre pero con diferentes argumentos." corresponde a	SobreCarga de metodos

## 8.2 PosTest

Las preguntas realizadas en el PosTest son las siguientes:

1.La definición: "sirven para definir variables que contienen valores numéricos, lógicos y caracteres unitarios " corresponde a

- ☐ Excepciones
- ☐ Generar una instancia
- ☐ Atributos primitivos
- ☐ Datos enteros

2.La estructura principal de una aplicación en java contiene

- ☐ Comentarios
- ☐ Metodo main
- ☐ Clase
- ☐ Todas las anteriores

3.Las clases están compuestas por

- ☐ Variables de instancia
- ☐ Constructor
- ☐ Métodos
- ☐ Todas las anteriores

Figura 8.7 PosTest de la pregunta 1 a la 3

4. Una clase se puede definir como

- ☐ Una plantilla que define las variables y métodos a ser incluidas en un tipo de objeto específico
- ☐ Un concepto similar a un vector
- ☐ Una lista de objetos ordenados de forma no secuencial
- ☐ Una categoría que ordena datos primitivos

5. public, private, protected y por default son

- ☐ Tipos de excepciones
- ☐ Clases de dominio
- ☐ Modificadores de visibilidad en Java
- ☐ Un tipo particular de variable

6. Crear un objeto a partir de una clase es también llamado :

- ☐ Duplicar la clase
- ☐ Instanciar la clase
- ☐ Eliminar la clase
- ☐ Conectar la clase

Figura 8.8 PosTest de la pregunta 4 a la 6

7. Declare una variable del tipo string y asignele un contenido

Short-answer text

---

8. Declare un arreglo de caracteres

Short-answer text

---

9. El contenido por defecto para arreglos de cadenas en Java es

- ☐ 0
- ☐ Null
- ☐ Contenido basura
- ☐ 'a'

10. La sintaxis para declarar e inicializar un array será:

- ☐ Nombre\_del\_array [][] = Tipo\_de\_variable[dimensión];
- ☐ New Tipo\_de\_variable[] Nombre\_del\_array = Tipo\_de\_variable[dimensión];
- ☐ Tipo\_de\_variable Nombre\_del\_array [] = Tipo\_de\_variable[dimensión];
- ☐ Tipo\_de\_variable[] Nombre\_del\_array = new Tipo\_de\_variable[dimensión];

Figura 8.9 PostTest de la pregunta 7 a la 10

11. La definición "conjunto de instrucciones definidas dentro de una clase, que realizan una determinada tarea y a las que podemos invocar mediante un nombre " corresponde a :

- ☐ Excepción
- ☐ Variable de control
- ☐ Metodo
- ☐ Funcion Main()

12. Si un método posee el tipo de retorno "void" significa que:

- ☐ Su retorno es variable segun los parametros de entrada
- ☐ Su retorno es variable y no depende de los parametros de entrada
- ☐ No posee retorno
- ☐ Es una funcion con retorno indefinido

13. Al ejecutar la linea " MiClase a = new MiClase();" .

- ☐ Se puede instanciar un objeto a de la clase MiClase
- ☐ Siempre provoca un error en el compilador
- ☐ Siempre se provoca un error por falta de argumentos

Figura 8.10 PosTest de la pregunta 11 a la 13

14. Cuál sintaxis corresponde a la cabecera de un método que tiene por nombre cambiaLetra, modificador de visibilidad público, retorna un valor tipo entero y recibe por parámetro un carácter.

- ☐ private char cambiaLetra (char letra);
- ☐ public int saludar(int hola);
- ☐ public int cambiaLetra(char letra);
- ☐ public void cambiaLetra(char letra);

15."Instanciar" corresponde a:

- ☐ Escribir cualquier tipo de código en Java
- ☐ Reservar una porción determinada de memoria en la que se almacenarán los datos, y las direcciones de memoria de :
- ☐ Ejecutar una excepción
- ☐ Limpiar una variable

Figura 8.11 PosTest de la pregunta 14 a la 15

A continuación, se presenta una tabla resumen donde se logra identificar la categoría de cada pregunta junto con su respuesta correcta.

Tabla 8.2 Tabla que identifica las categorías de las preguntas del posTest

Numero	Temática	Pregunta	Respuesta
1	Atributos	1.La definición: "sirven para definir variables que contienen valores numéricos, lógicos y caracteres unitarios " corresponde a	Atributos primitivos
2	Clase	2.La estructura principal de una aplicación en java contiene	Todas las anteriores
3	Clase	3.Las clases están compuestas por	Todas las anteriores
4	Clase	4.Una clase se puede definir como	Una plantilla que define las variables y métodos a ser incluidas en un tipo de objeto específico
5	Atributos	5. public,private,protected y por default son	Modificadores de visibilidad en Java

6	Objetos	6. Crear un objeto a partir de una clase es también llamado :	Instanciar la clase
7	Atributos	7. Declare una variable del tipo string y asígnele un contenido	<code>String[] hola = "Hola Mundo";</code>
8	Arreglos	8. Declare un arreglo de caracteres	<code>char[] array = new char[5];</code>
9	Arreglos	9. El contenido por defecto para arreglos de cadenas en Java es	Null
10	Arreglos	10. La sintaxis para declarar e inicializar un array será:	<code>Tipo_de_variable[ Nombre_del_array= new Tipo_de_variable[dimensión];</code>
11	Metodo	11. La definición "conjunto de instrucciones definidas dentro de una clase, que realizan una determinada tarea y a las que podemos invocar mediante un nombre " corresponde a :	Metodo
12	Metodo	12. Si un método posee el tipo de retorno "void" significa que:	No posee retorno
13	Objetos	13. Al ejecutar la línea " MiClase a = new MiClase();" .	Se puede instanciar un objeto a de la clase MiClase
14	Metodo	14. Cuál sintaxis corresponde a la cabecera de un método que tiene, modificador de visibilidad público, retorna un valor tipo entero y recibe por parámetro un carácter.	<code>public int cambiaLetra(char letra);</code>
15	Objetos	15. "Instanciar" corresponde a :	Reservar una porción determinada de memoria en la que se almacenarán



			los datos, y las direcciones de memoria de sus métodos
--	--	--	--------------------------------------------------------

## 9 Pruebas Realizadas a Estudiantes sin Conocimiento de *POO*

Las pruebas de usuario se realizaron al curso de Estructura de datos a los alumnos de Ingeniería de ejecución en informática de la universidad *PUCV*. En un comienzo se pretendía tener un grupo de control y un grupo experimental, el cual consistía que el grupo de control no tuviera feedback y tutoriales, mientras que el grupo experimental si los poseía. La problemática que surgía al realizar las pruebas con este método fue que los alumnos no sabían de antemano la materia expuesta en el juego por lo que de todas formas tenía que ser tutoriales para este grupo, el feedback contaba con música, Text to speech y sistema de puntuación; con respecto al Text to speech, este funcionaba con el adaptador de Windows, por lo que los computadores del laboratorio donde se hicieron las pruebas contaba solo con el adaptador de voz en inglés, de esta forma los diálogos que decía el personaje Ashy sonaban de manera extraña,; con respecto a la puntuación se tomó en consideración de que ellos recién estaban experimentando con este nuevo lenguaje de programación, por lo que se asumió que tendrían baja puntuación o en mucho de los casos nada de puntuación; considerando todo aquello se decidió solo tener un grupo de prueba.

Los participantes fueron 20 en total, con una hora y media para la realización de éstas.

El PreTest se realizo a las 14:06, posteriormente a medida que los estudiantes terminaban de contestar el PreTest estos procedían a probar el software educativo, a las 14:25 se les pidió a los alumnos que terminaran de utilizar el software a pesar de que no hayan terminado el juego, para que realizaran el PosTest.



Figura 9.1 Los participantes haciendo las pruebas

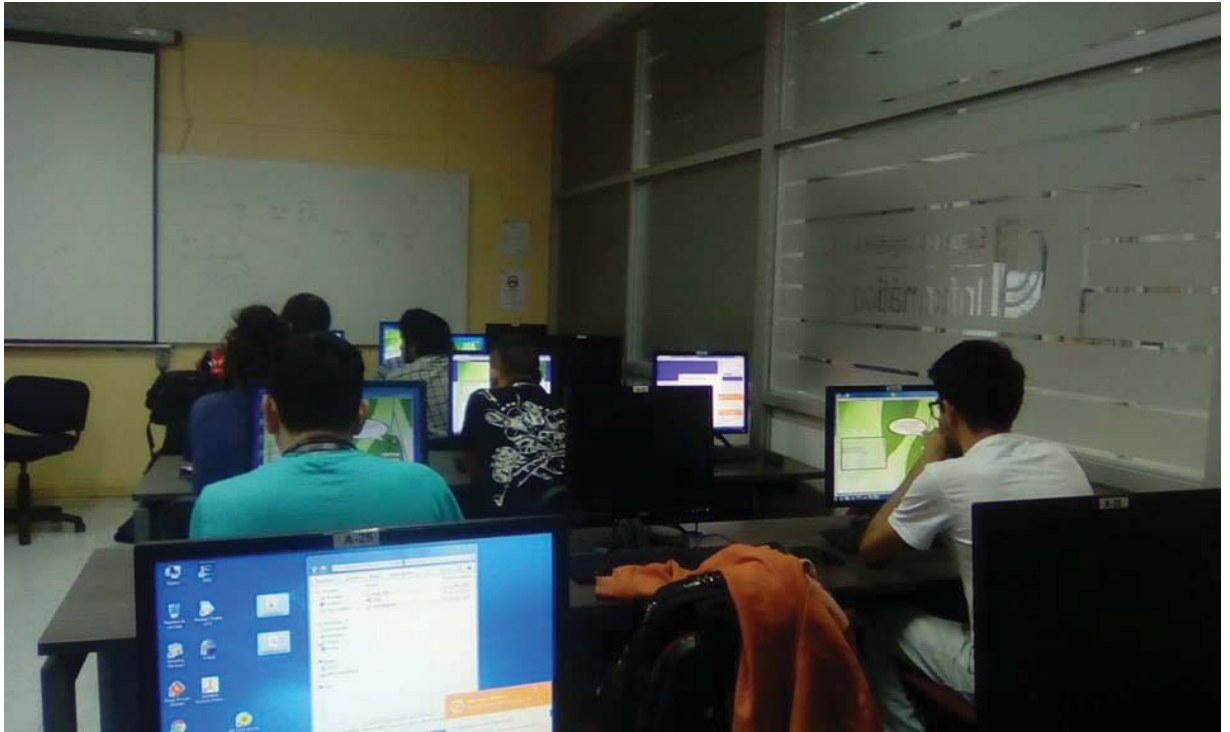


Figura 9.2 Realización de las pruebas

## 9.1 Respuestas de los Participantes por cada Pregunta PreTest

Las respuestas correctas son representa en color verde en el gráfico de torta.

Pregunta 1:

Tabla 9.1 Tabla de la pregunta 1, usuarios sin saber Java

Es un concepto similar al de 'array'	2
Es un tipo particular de variable	4
Es un modelo o plantilla a partir de la cual creamos objetos	10
Es una categoría de datos ordenada secuencialmente	4

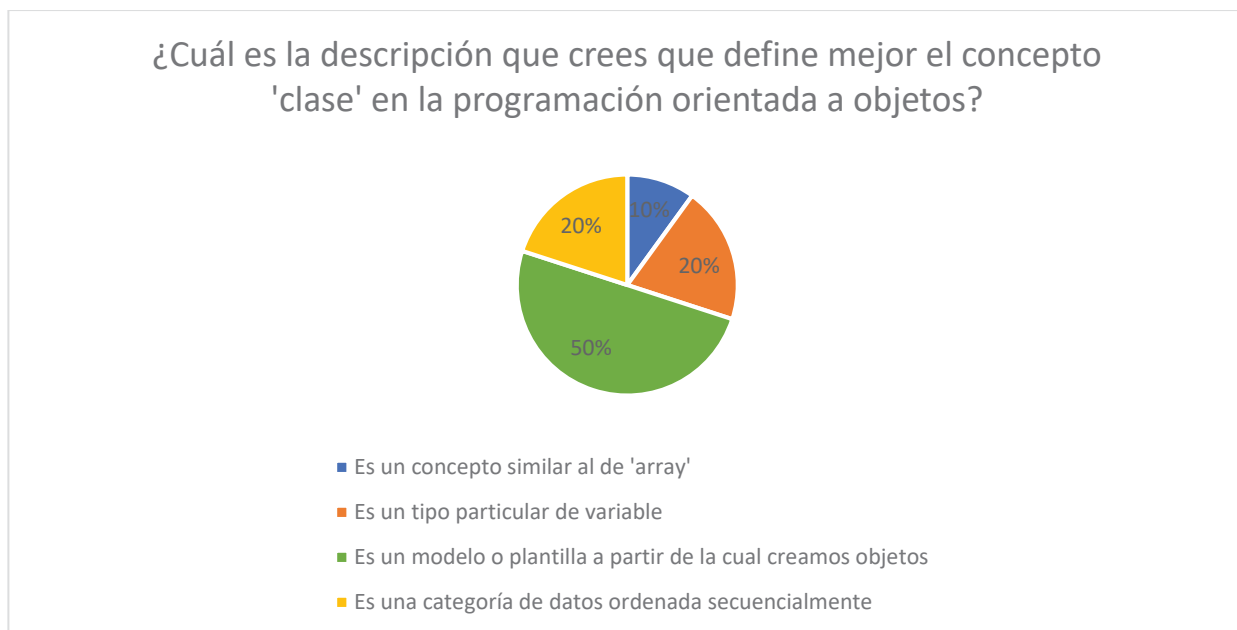


Figura 9.3 Grafico de la pregunta 1, usuarios sin saber Java

Podemos observar que un 50% del curso logro responder de forma correcta la pregunta que involucra clases, debido a que el curso no había visto POO con anterioridad, deja sorprendentes resultados ya que el 50% supo la respuesta correcta sin ningún material de ante mano.

Pregunta 2:

Tabla 9.2 Tabla de la pregunta 2, usuarios sin saber Java

Publico, Privado y mixto	3
public,private,protected y por default	7
import, package, java.util	2
public, private, protected	8

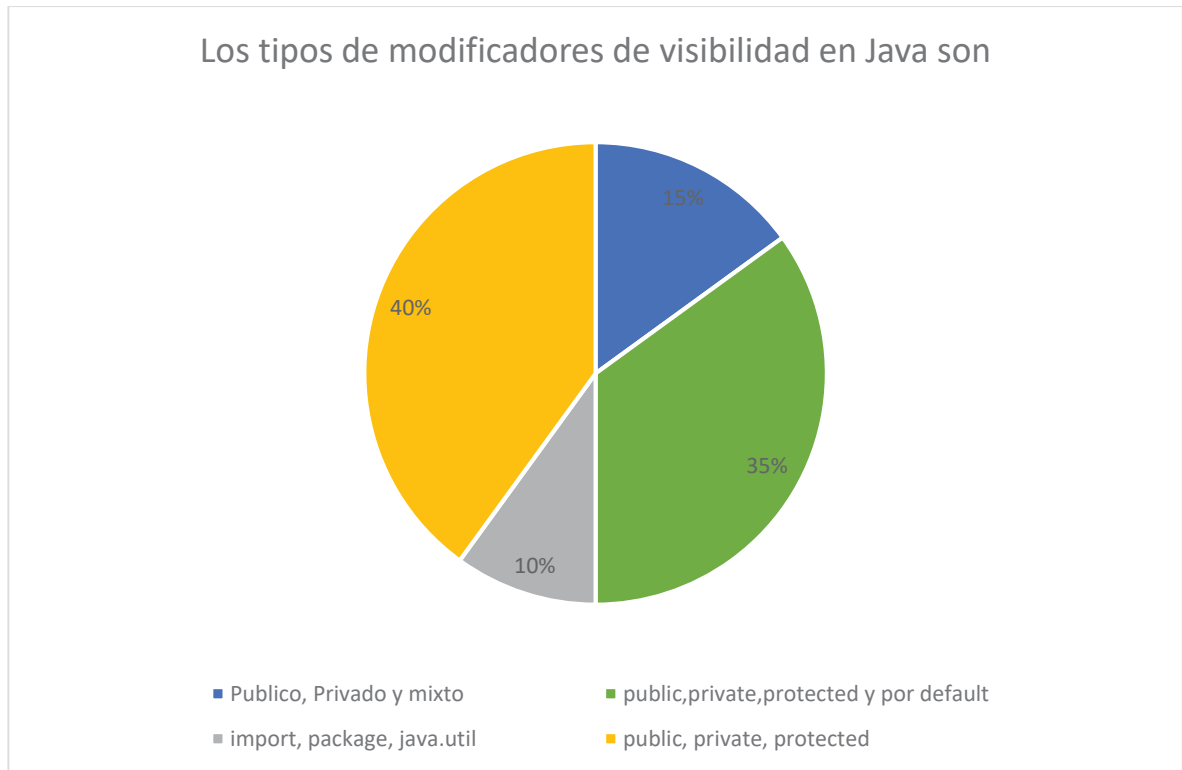


Figura 9.4 Grafico de la pregunta 2, usuarios sin saber Java

Podemos observar que el 35% de los participantes respondió de forma correcta este bajo porcentaje se puede atribuir a que es primera vez que ven un concepto similar a este.

Pregunta 3:

Tabla 9.3 Tabla de la pregunta 3, usuarios sin saber Java

Sus cardinalidad y su tipo	4
Sus atributos y sus métodos	13
La forma en que establece comunicación e intercambia mensajes	1
Su interfaz y los eventos asociados	2

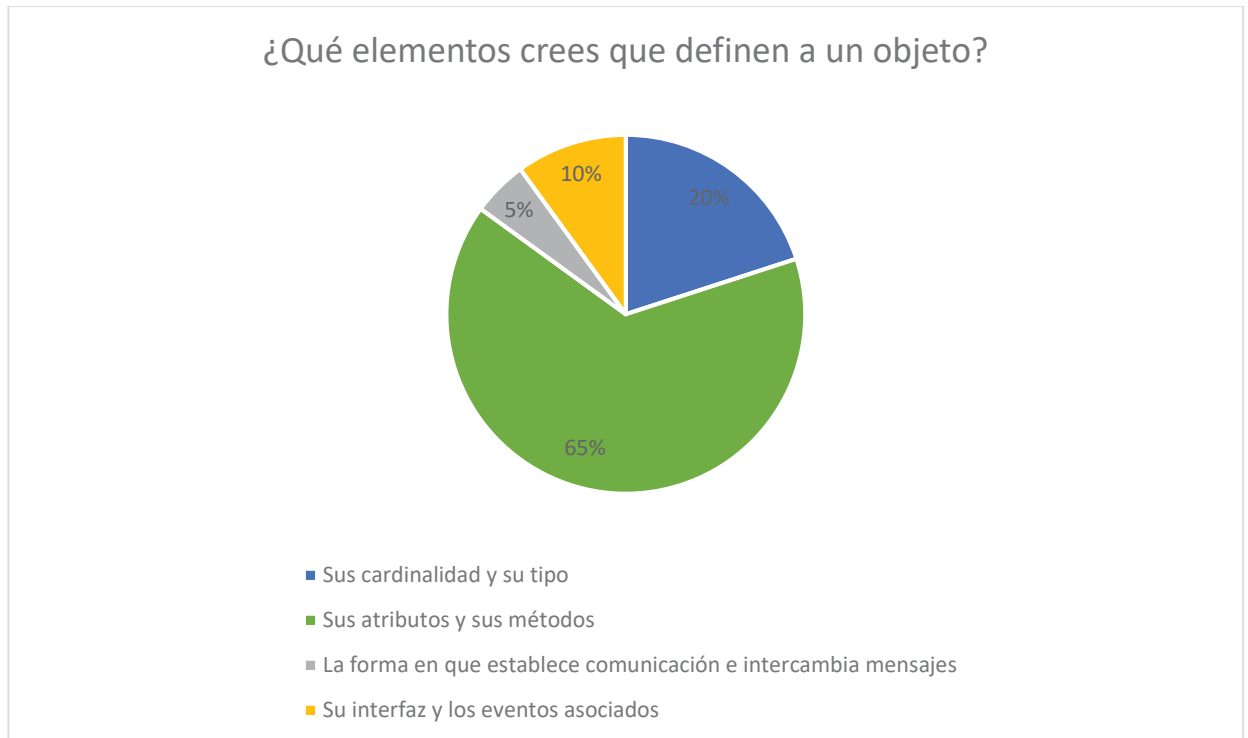


Figura 9.5 Grafico de la pregunta 3, usuarios sin saber Java

Esta pregunta también sorprende ya que mas del 50% responde de forma correcta a un concepto que no han visto con anterioridad.

Pregunta 4:

Tabla 9.4 Tabla de la pregunta 4, usuarios sin saber Java

Duplicar una clase	0
Eliminar una clase	0
Crear un objeto a partir de la clase	15
Conectar dos clases entre sí	5

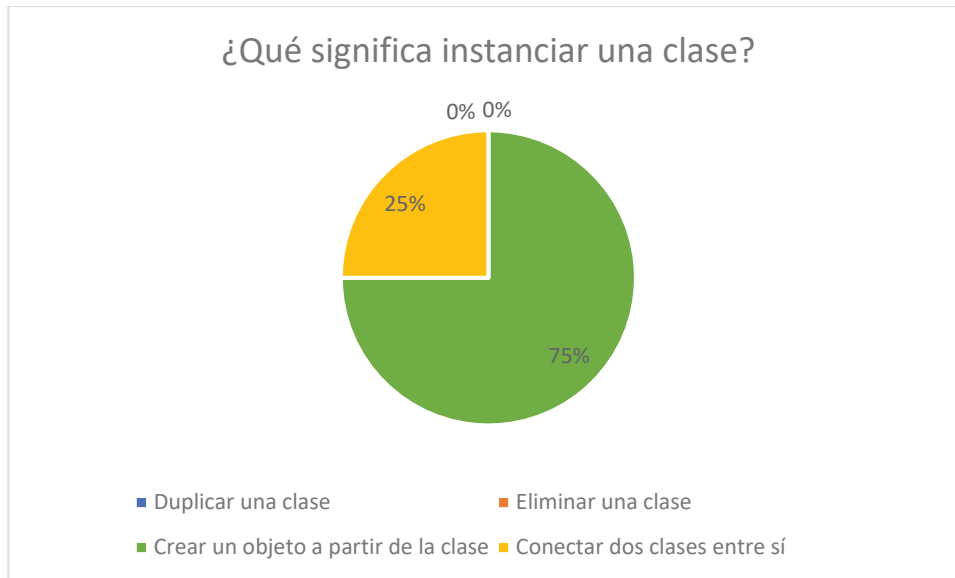


Figura 9.6 Grafico de la pregunta 4, usuarios sin saber Java

Por lo que se puede apreciar el 75% de los participantes dieron una respuesta correcta, dando a conocer que habían experimentado antes con algunos conceptos de la POO.

Pregunta 5:

Tabla 9.5 Tabla de la pregunta 5, usuarios sin saber Java

saludar();	3
persona->saludar();	3
persona.saludar();	11
ninguna de las anteriores	3

Desde una clase externa, la llamada del método "saludar()", que posee el objeto "persona" se realiza de la forma:

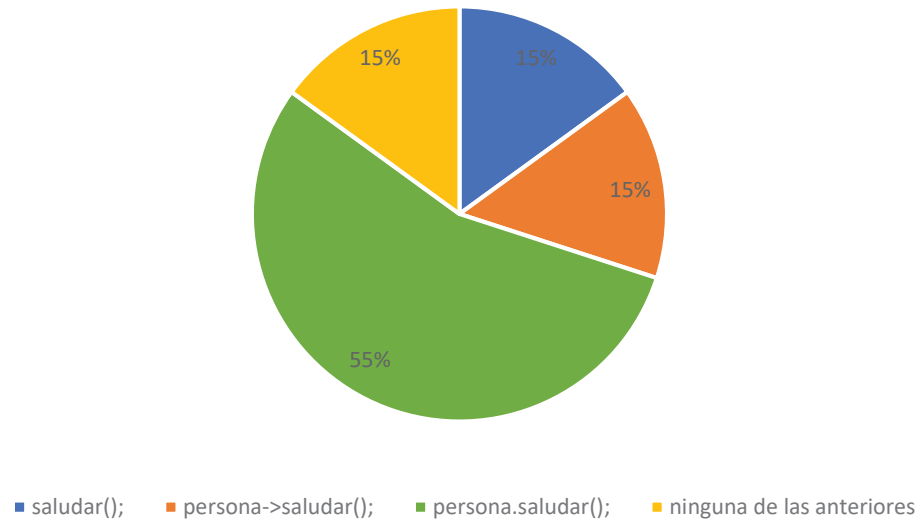


Figura 9.7 Grafico de la pregunta 5, usuarios sin saber Java

Se observa que el 55% de los estudiantes respondió de forma correcta mientras que el resto de los participantes se dividió entre el resto de las respuesta con un 15% en cada una de ellas.

Pregunta 6:

Tabla 9.6 Tabla de la pregunta 6, usuarios sin saber Java

Clase	1
Herencia	3
Método	11
Casting	5





Figura 9.8 Grafico de la pregunta 6, usuarios sin saber Java

Nuevamente podemos interpretar que el 55% de los participantes había tenido alguna cercanía con el tema tratado.

Pregunta 7:

Tabla 9.7 Tabla de la pregunta 7, usuarios sin saber Java

<code>int vector [] = new int [10];</code>	17
<code>int vector [] = new int [10]</code>	0
<code>int vector [][] = new int [5][5];</code>	0
<code>int vector = new int [10];</code>	3

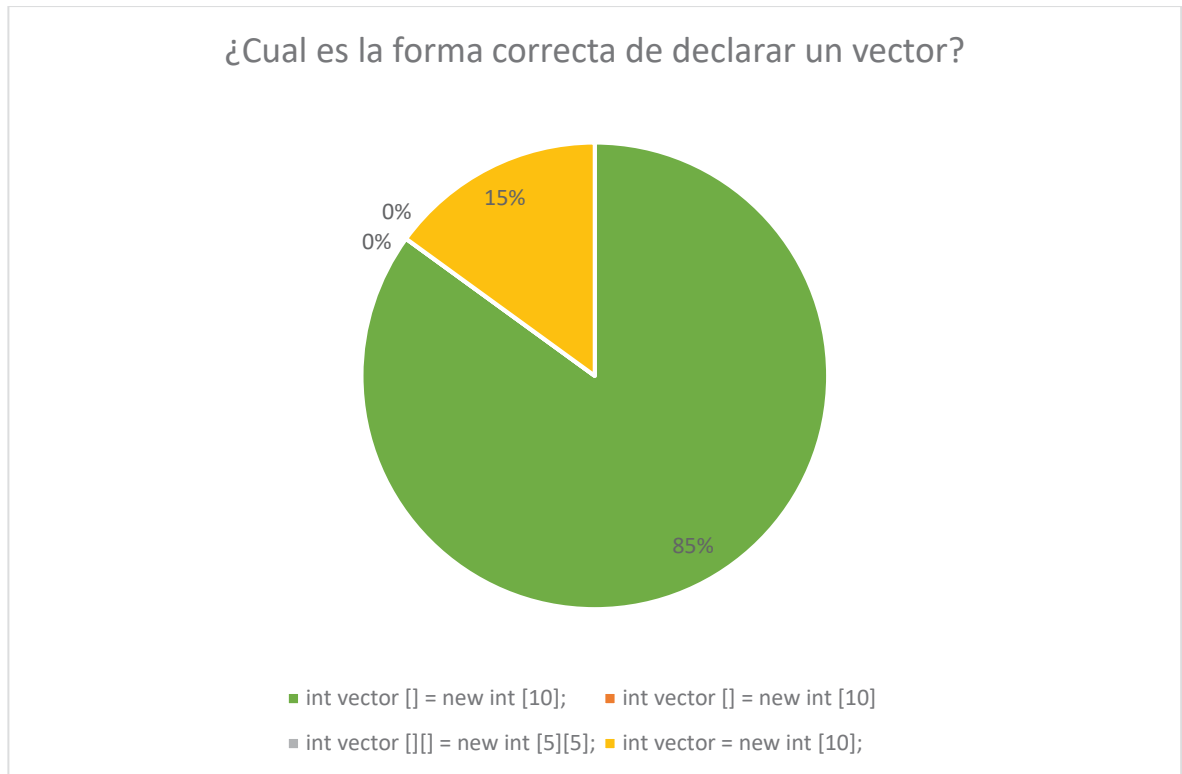


Figura 9.9 Grafico de la pregunta 7, usuarios sin saber Java

Se demuestra que 85% de los participantes pudo responder de forma correcta la pregunta propuesta, se concluye que esto debe ser por la similitud del lenguaje C con Java al crear arreglos.

Pregunta 8:

Tabla 9.8 Tabla de la pregunta 8, usuarios sin saber Java

MiClas, extens, int b	1
Lápiz, Pez, Reloj	10
abstract,assert,Hola	2

Algunos ejemplos de nombres correctos para objetos son:

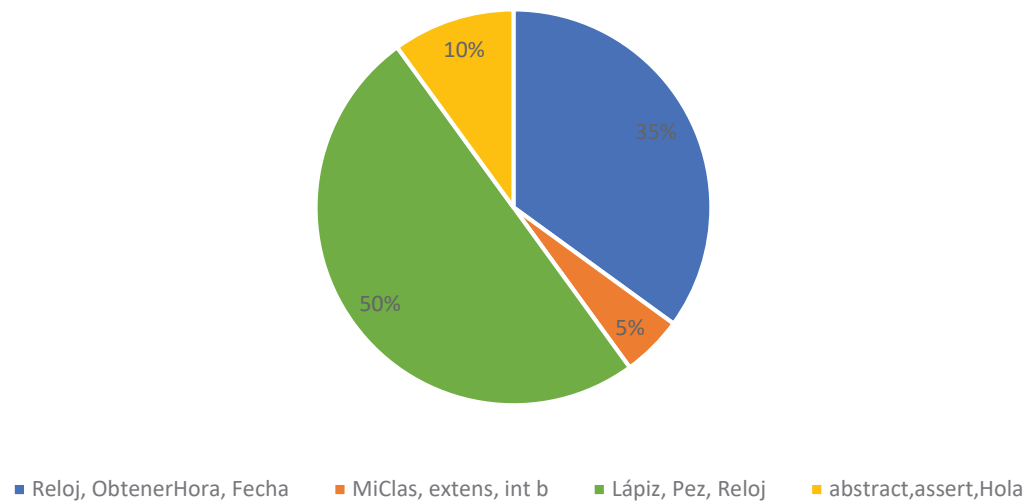


Figura 9.10 Grafico de la pregunta 8, usuarios sin saber Java

Esta era una pregunta compleja ya que daba a pensar de que tiene dos respuestas correctas, por lo que podemos apreciar que el 35% de los participantes puso la respuesta que daba a la confusión mientras que el 50% pudo responder la respuesta correcta.

Pregunta 9:

Tabla 9.9 Tabla de la pregunta 9, usuarios sin saber Java

Entero,punto flotante, caracter	1
Númérico, carácter, logico	2
Numerico, punto flotante, logico, string	10

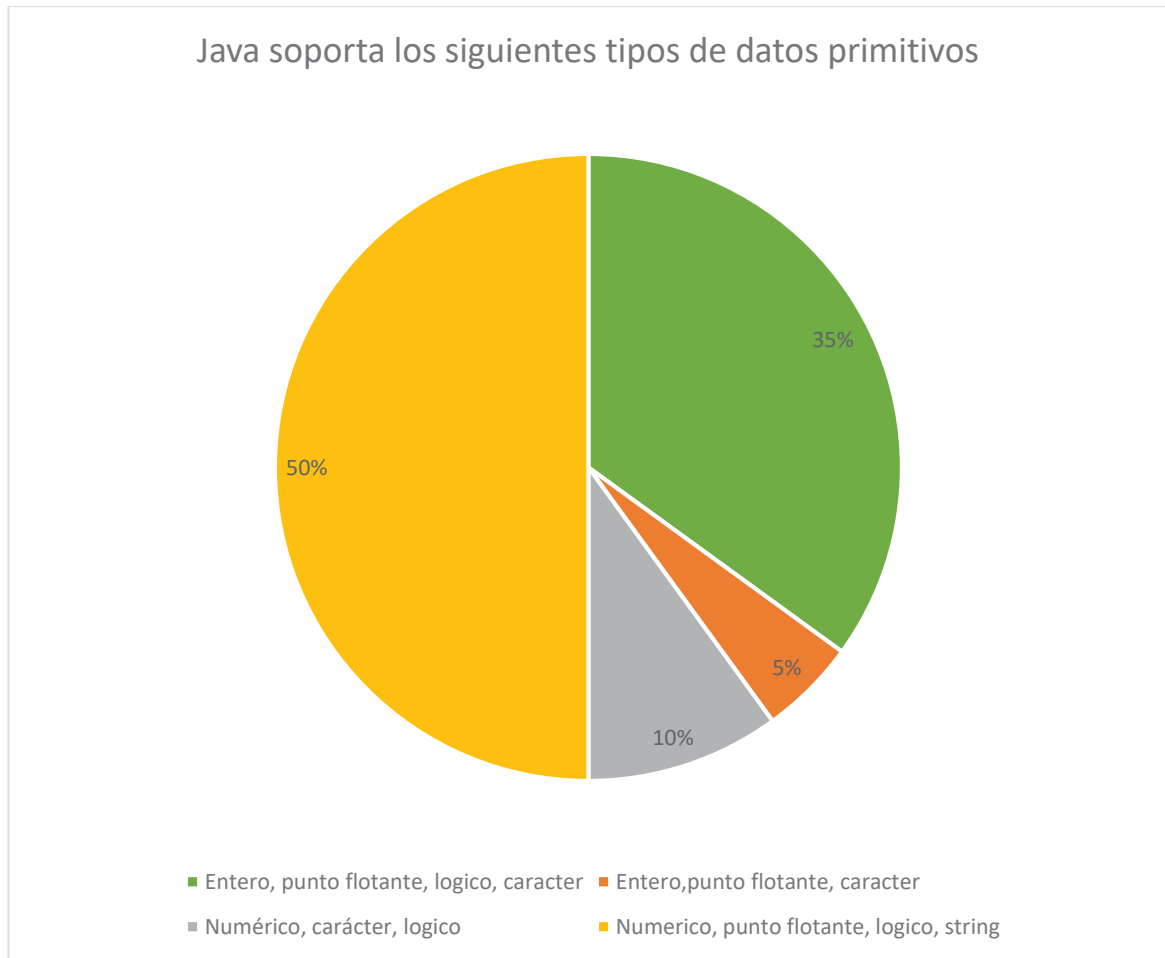


Figura 9.11 Grafico de la pregunta 9, usuarios sin saber Java

Podemos ver que un 35% de los participante respondió de manera correcta mientras que un 50% opto por al respuesta que incluía el string dentro de las respuestas, esto se puede dar ya que el lenguaje de programación en C no tiene distinción dentro del lenguaje.

Pregunta 10:

Tabla 9.10 Tabla de la pregunta 10, usuarios sin saber Java

Si	11
No	8
Sin responder	1

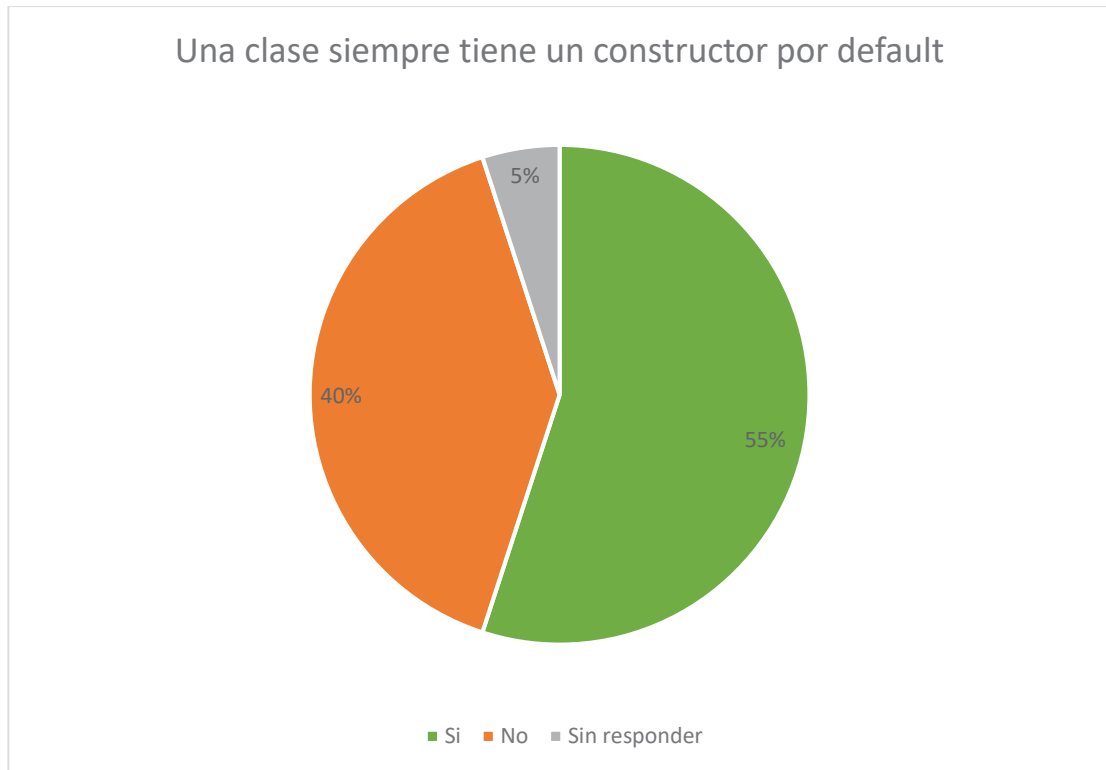


Figura 9.12 Grafico de la pregunta 10, usuarios sin saber Java

Nuevamente el 55% de los participantes tuvo la respuesta correcta con respecto a las temáticas que tienen que ver con clases.

Pregunta 11:

Tabla 9.11 Tabla de la pregunta 11, usuarios sin saber Java

Nombrar con mas facilidad una clase	2
Utilizar métodos , clases y atributos predefinidos	16
Manejar excepciones	0
Importar clases del mismo package	2

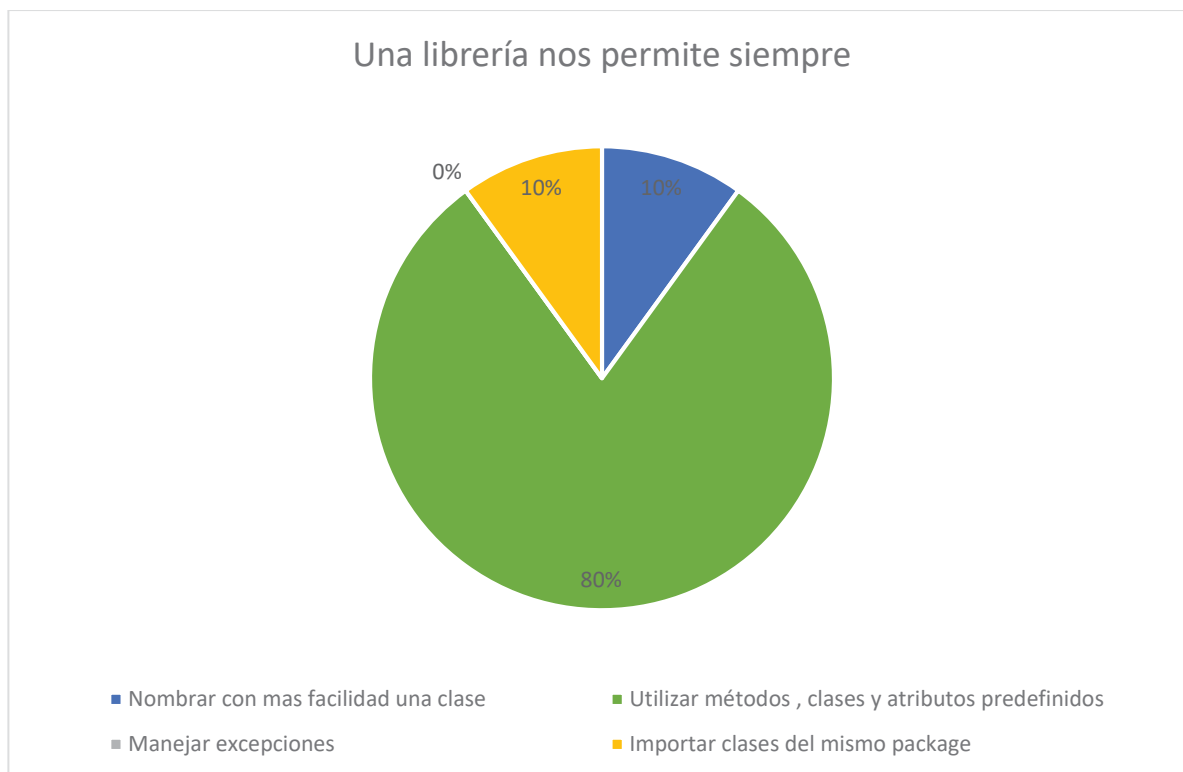


Figura 9.13 Grafico de la pregunta 11, usuarios sin saber Java

El 80% de los participantes respondió de forma correcta, podemos suponer que esto se debe a que el lenguaje de programación en C utiliza bibliotecas que son similares a las librería en Java, por lo que los participantes se guiaron por esa similitud.

Pregunta 12:

Tabla 9.12 Tabla de la pregunta 12, usuarios sin saber Java

Si y depende del tipo de dato del arreglo	8
No, contiene contenido basura	12

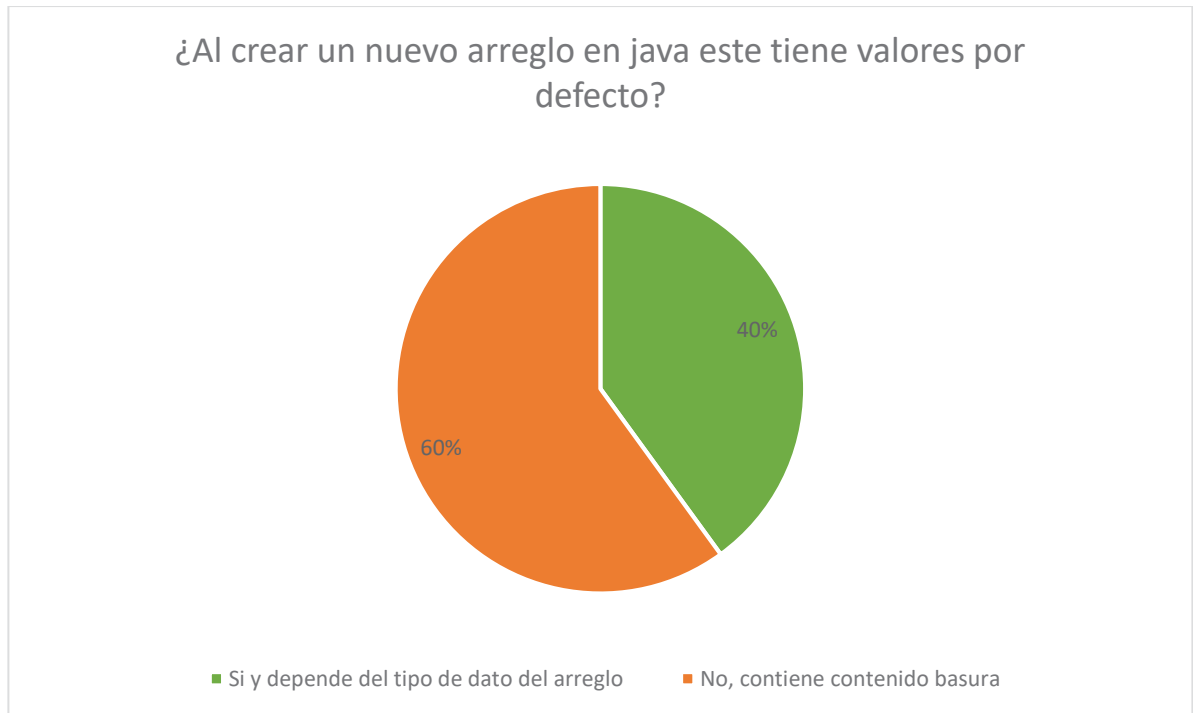


Figura 9.14 Grafico de la pregunta 12, usuarios sin saber Java

El 40% de los participantes tuvo buena esta respuesta versus un 60% que lo tuvo erróneo, se presume que en el lenguaje de programación en C, creaba datos basura, por lo que los participantes consideran que en el lenguaje de programación en Java puede llegar a ser similar.

Pregunta 13:

Tabla 9.13 Tabla de la pregunta 13, usuarios sin saber Java

Instanciando la clase	17
Agregando librerías a una clase	1
Haciendo sobrecarga de métodos	0
Creando una interfaz de usuario	2

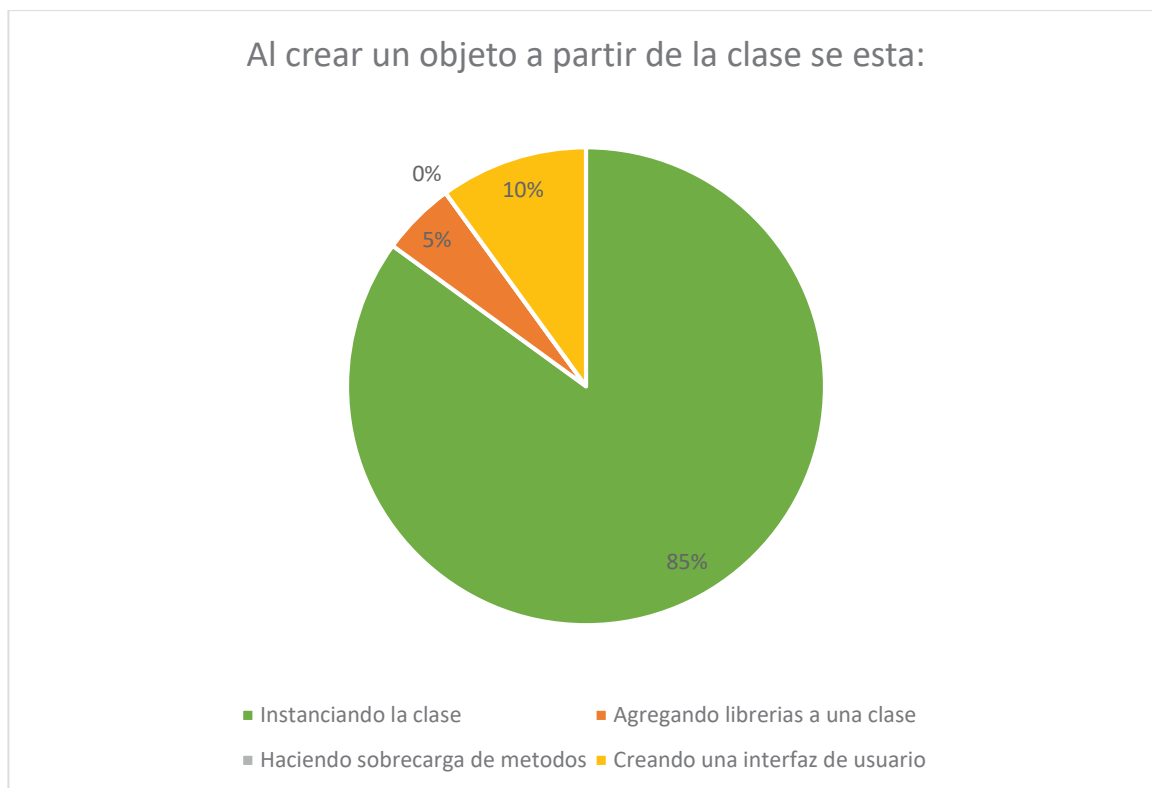


Figura 9.15 Grafico de la pregunta 13, usuarios sin saber Java

Se toma en consideración que dentro de las preguntas anteriores se realizo una pregunta muy similar a esta por lo que los participantes se dejaron llevar por ambas, dando como resultado que el 85% de los estudiantes respondiera de forma correcta.

Pregunta 14:

Tabla 9.14 Tabla de la pregunta 14, usuarios sin saber Java

Sobre carga de un método	1
La sintaxis para declarar e inicializar un arreglo de String	13
La sintaxis para declarar e inicializar un arreglo de enteros	0
Un error de sintaxis	5
Sin responder	1



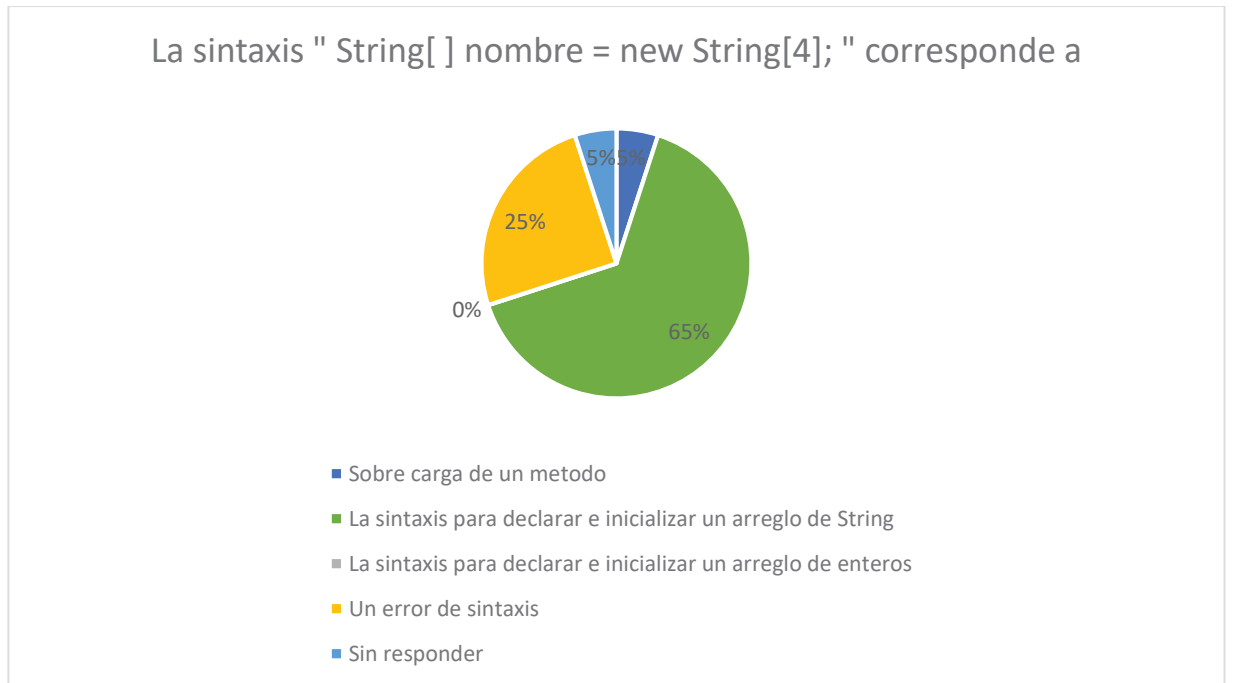


Figura 9.16 Grafico de la pregunta 14, usuarios sin saber Java

Al igual que la pregunta anterior, se considera que dentro de el test hubo una pregunta similar, por lo que, los estudiantes lo tomaron en consideración, dando como resultado que le 65% de los participantes tuviera la respuesta correcta.

Pregunta 15:

Tabla 9.15 Tabla de la pregunta 15, usuarios sin saber Java

Error de compilación	1
Excepción	3
SobrEscritura de métodos	12
SobreCarga de métodos	3
Sin responder	1

La definición "Métodos en una clase que tienen el mismo nombre pero con diferentes argumentos." corresponde a

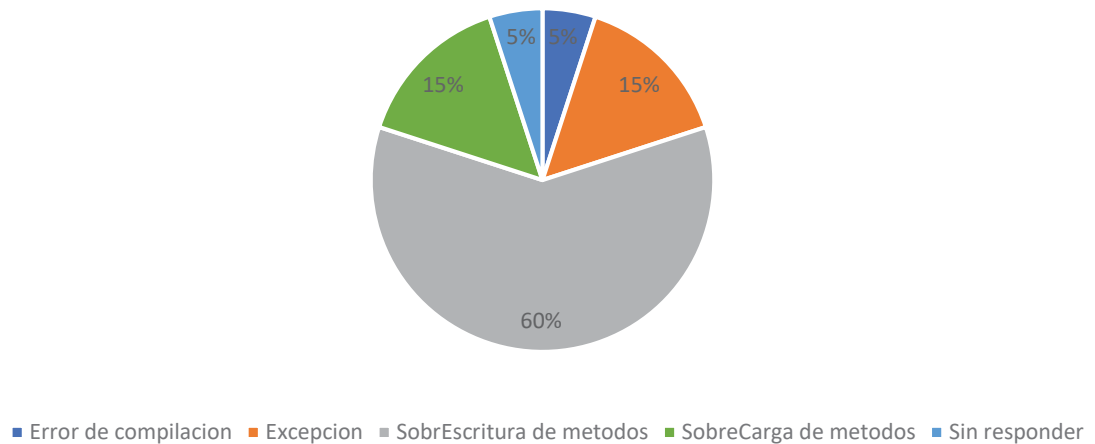


Figura 9.17 Grafico de la pregunta 15, usuarios sin saber Java

Este es un concepto complejo de entender y que no se trata dentro del software educativo, tampoco es un concepto básico, por lo que, se esperaba un bajo porcentaje de respuestas correctas, podemos observar que el 60% de los participantes opto por la opción de SobreEscritura, mientras que el 15% opto por la respuesta correcta que es SobreCarga.

### 9.1.1 Conclusiones de las Respuestas

A pesar de que los estudiantes tenían poco conocimiento de los contenidos expuestos en el PreTest los alumnos presentaron un buen manejo de los conceptos evaluados, por lo que se espera que el porcentaje que no pudo responder de forma correcta, logre mejorar al realizar el testeo del software educativo llamado Ashy.alRescate().

## 9.2 Respuestas de los Participantes por cada Pregunta PosTest

Las respuestas correctas serán interpretas de color verde dentro del grafico de torta.

Pregunta 1:

Tabla 9.16 Tabla de la pregunta 1 del posTest, usuarios sin saber Java

Excepciones	0
Generar una instancia	4
Atributos primitivos	14
Datos enteros	2
Datos	0

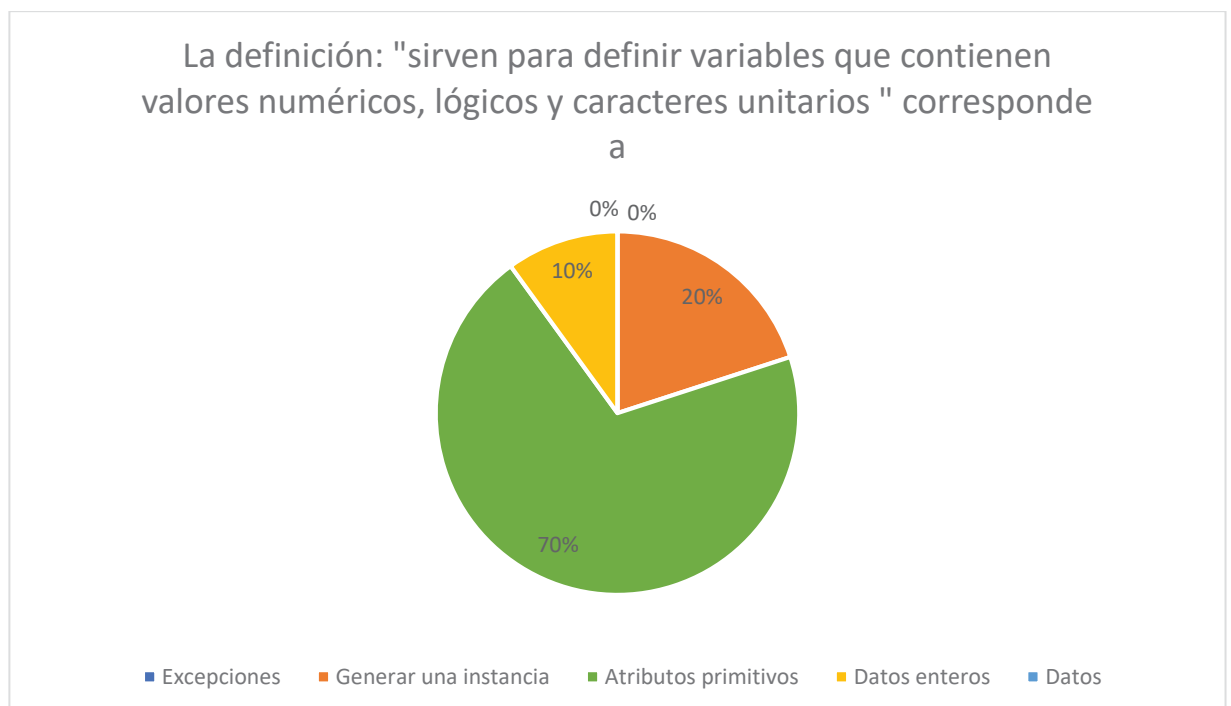


Figura 9.18 Grafico de la pregunta 1 del posTest, usuarios sin saber Java

Podemos notar una gran mejora en esta pregunta ya que se les hizo una muy similar en el PreTest dejando en evidencia de que sabían muy poco respecto al tema ya que el 35% de los participantes respondió de forma correcta en el PreTest, por lo que, podemos apreciar que ahora el 70% de los participantes les quedo claro cuales son los Atributos primitivos para el lenguaje de programación en Java.

Pregunta 2:

Tabla 9.17 Tabla de la pregunta 2 del posTest, usuarios sin saber Java

Comentarios	0
Método main	1
Clase	3
Todas las anteriores	16

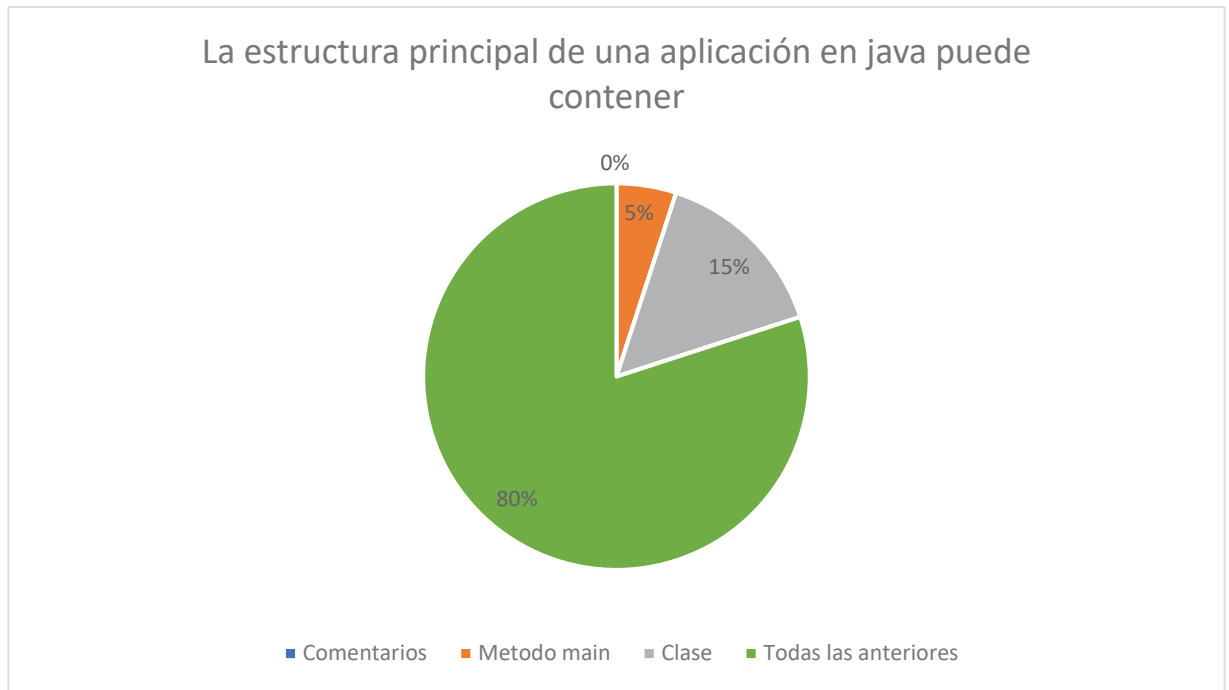


Figura 9.19 Grafico de la pregunta 2 del posTest, usuarios sin saber Java

A diferencia del PreTest donde las respuestas correctas no superaban el 65% exceptuando una pregunta, ahora notamos que ya el 80% de los participantes puede contestar de forma correcta.

Pregunta 3:

Tabla 9.18 Tabla de la pregunta 3 del posTest, usuarios sin saber Java

Variables de instancia	2
Constructor	0
Métodos	2
Todas las anteriores	16

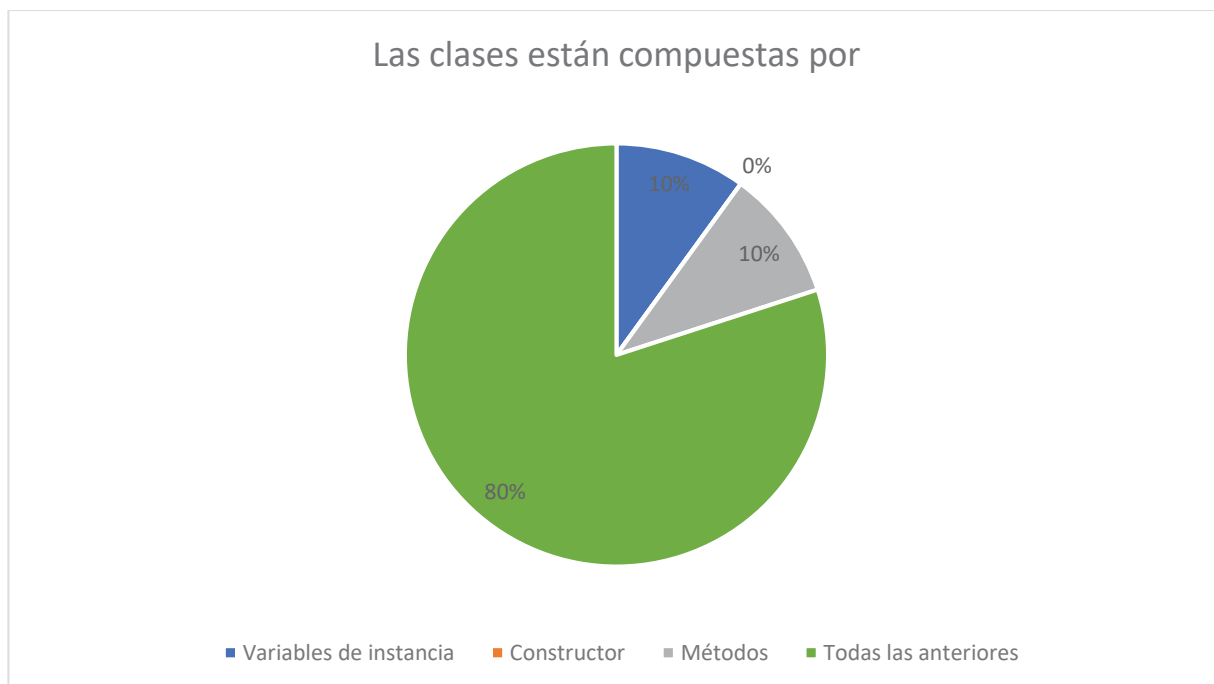


Figura 9.20 Grafico de la pregunta 3 del posTest, usuarios sin saber Java

Se demuestra una mejora en las respuestas de los participantes, ya que el 80% de los participantes logra responder de forma correcta preguntas sobre la temática Clases.

Pregunta 4:

Tabla 9.19 Tabla de la pregunta 4 del posTest, usuarios sin saber Java

Una plantilla que define las variables y métodos a ser incluidas en un tipo de objeto específico	18
Un concepto similar a un vector	0
Una lista de objetos ordenados de forma no secuencial	2
Una categoría que ordena datos primitivos	0

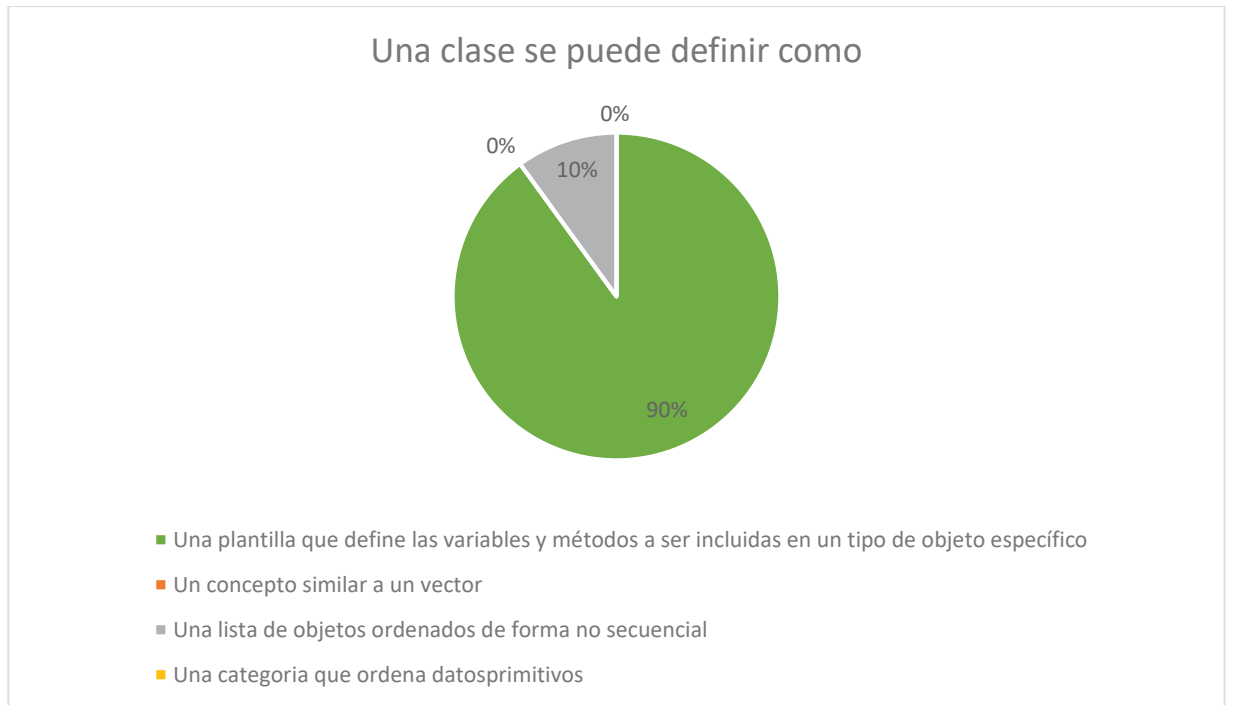


Figura 9.21 Grafico de la pregunta 4 del posTest, usuarios sin saber Java

Las respuestas correctas con respecto a esta pregunta ha sido del 90%, por lo que se están mostrando mejoras, con respecto a la claridad de algunos conceptos.

Pregunta 5:

Tabla 9.20 Tabla de la pregunta 5 del posTest, usuarios sin saber Java

Tipos de excepciones	3
Clases de dominio	10
Modificadores de visibilidad en Java	6
Un tipo particular de variable	1

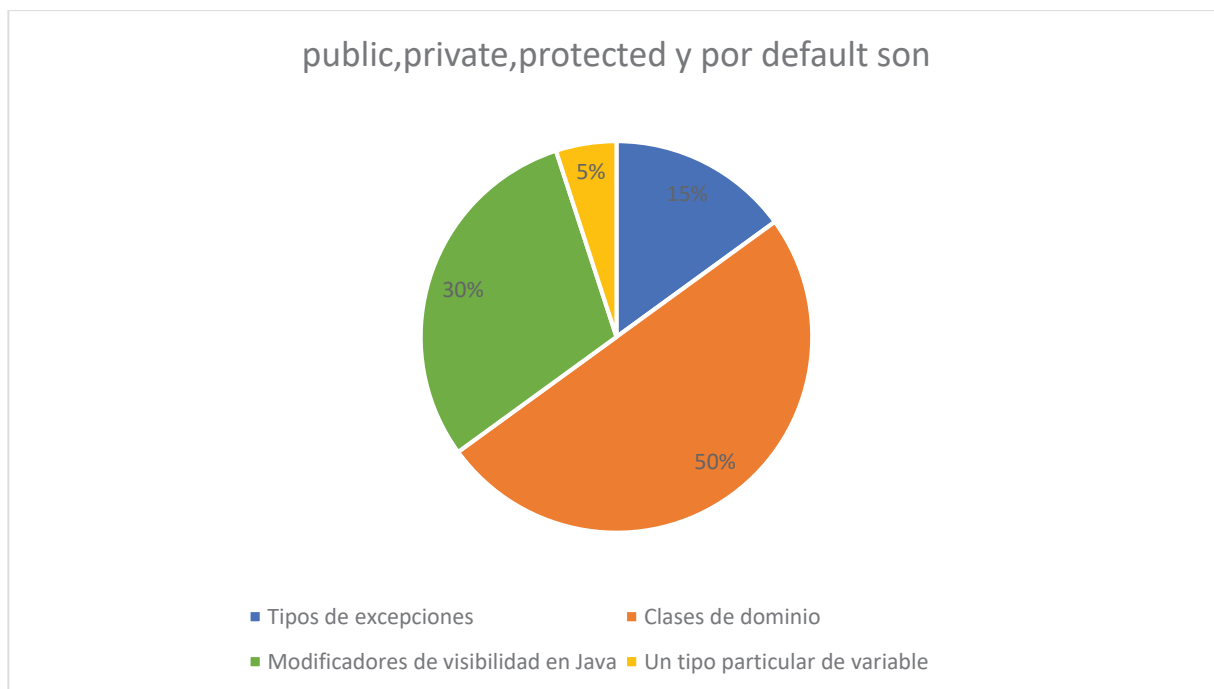


Figura 9.22 Grafico de la pregunta 5 del posTest, usuarios sin saber Java

En esta pregunta se esperaba que los participantes no respondieran de forma correcta, ya que el software educativo no contiene tutoriales que enseñe los modificadores de visibilidad en Java por lo que los estudiantes no podían saber la respuesta por medio del software educativo, obteniendo de esta manera un 30% de respuestas correctas.

Pregunta 6:

Tabla 9.21 Tabla de la pregunta 6 del posTest, usuarios sin saber Java

Duplicar la clase	1
Instanciar la clase	17
Eliminar la clase	0
Conectar la clase	2

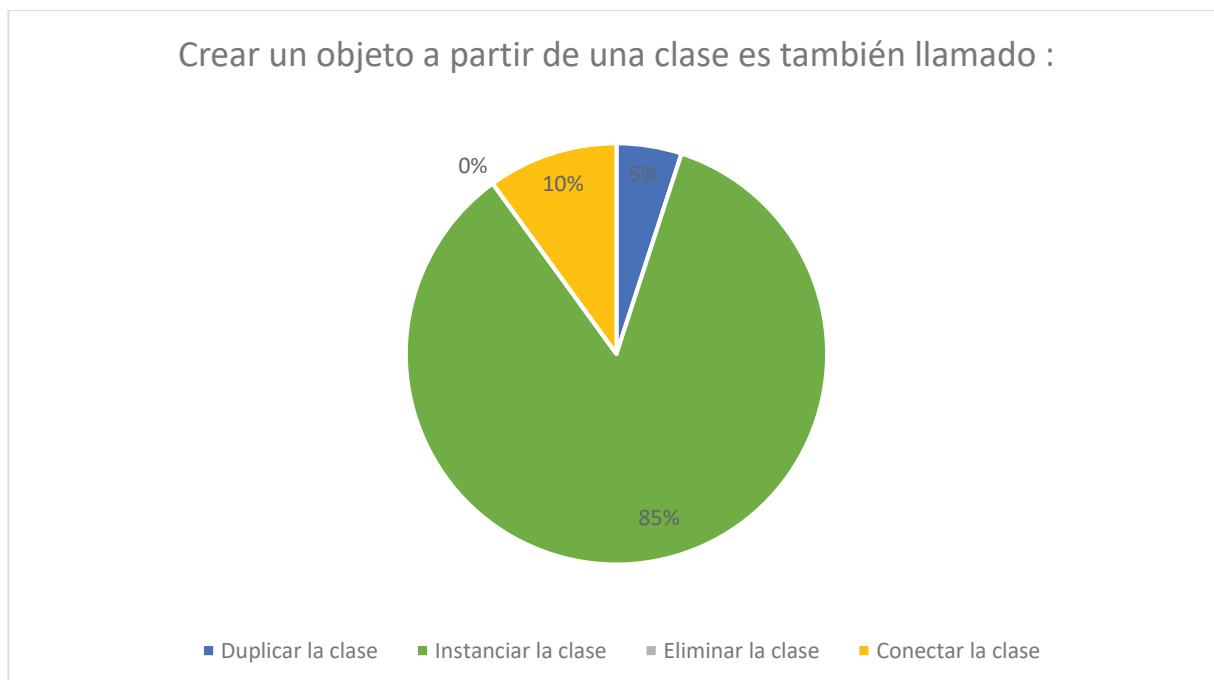


Figura 9.23 Grafico de la pregunta 6 del posTest, usuarios sin saber Java

Con la diferencia de la respuestas obtenidas en el PreTest ante una pregunta similar a esta, se obtiene el mismo porcentaje de respuestas correctas, por lo que se puede suponer que los alumnos que respondieron de forma incorrecta no entendieron de forma correcta los tutoriales propuestos en el software educativo.

Pregunta 7:

Tabla 9.22 Tabla de la pregunta 7 del posTest, usuarios sin saber Java

Respuestas correcta	14
Respuestas incorrectas	5
Sin responder	1



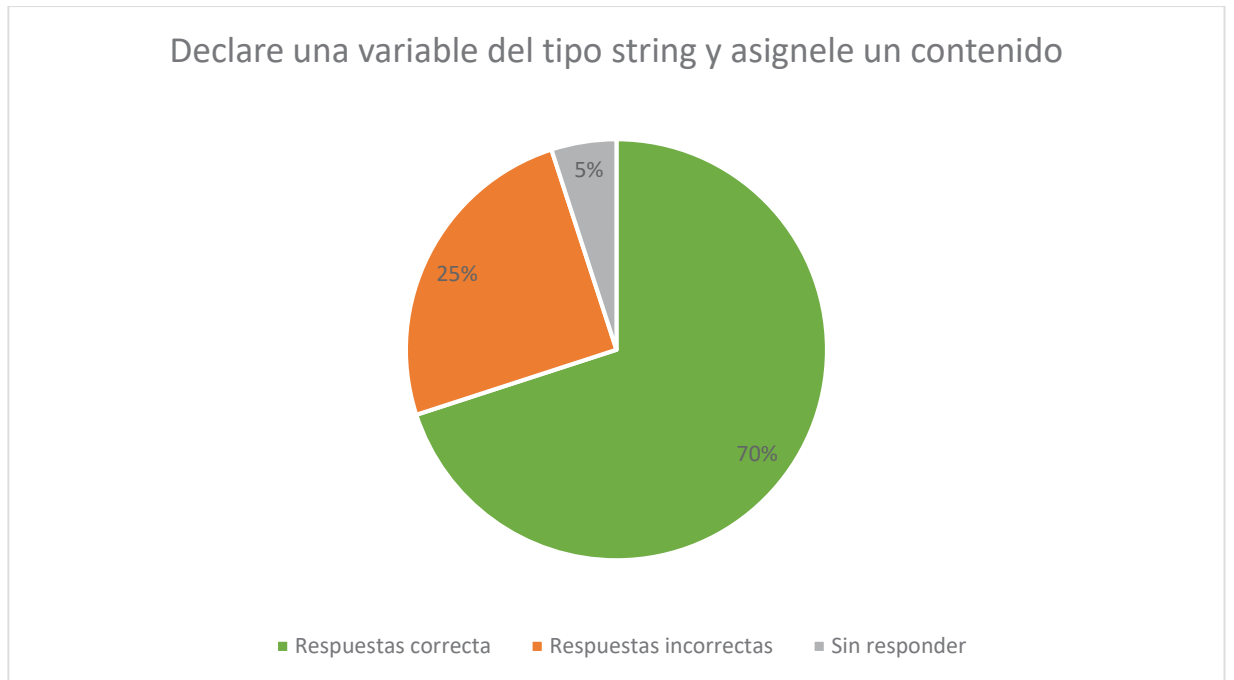


Figura 9.24 Grafico de la pregunta 7 del posTest, usuarios sin saber Java

Esta respuesta era escrita por lo analizando las respuestas de los estudiantes en el Test, los usuarios que respondieron de forma errónea se debió por falta de datos, exceptuando un usuario que al parecer no entendió la pregunta o se confundió, ya que realizo un char en vez de un String. También podemos apreciar que un usuario rechazo la propuesta de realizar esta pregunta, por lo que, no podemos saber de forma correcta la respuesta de este.

Pregunta 8:

Tabla 9.23 Tabla de la pregunta 8 del posTest, usuarios sin saber Java

Respuestas correcta	14
Respuestas incorrectas	6

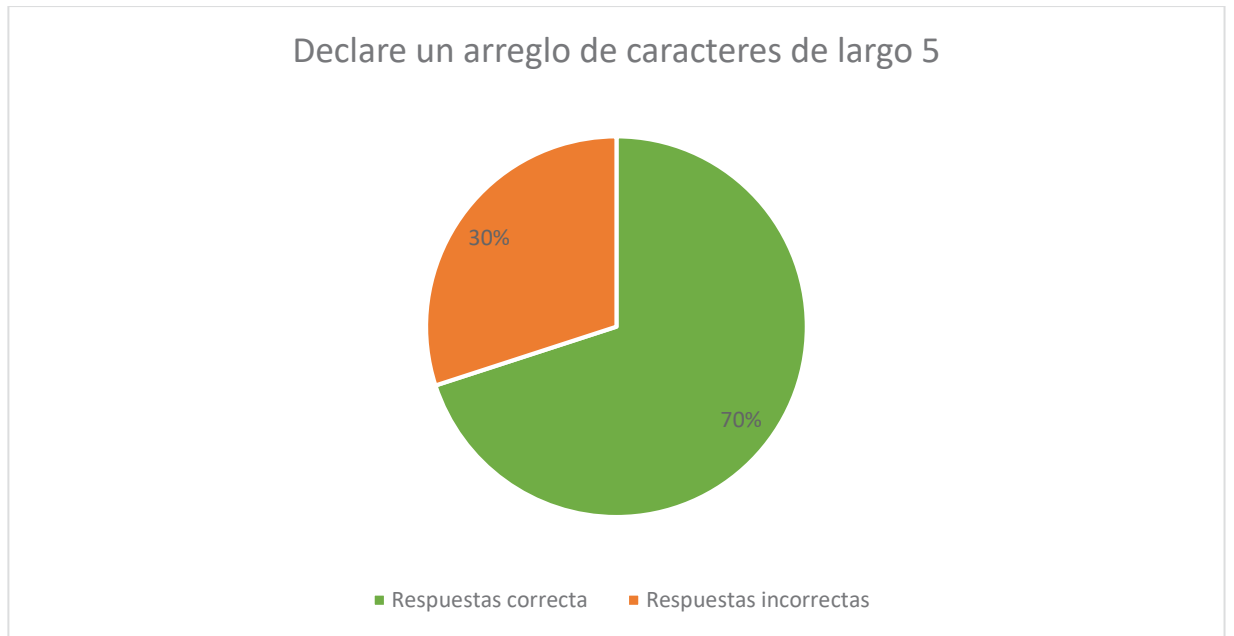


Figura 9.25 Grafico de la pregunta 8 del posTest, usuarios sin saber Java

Esta pregunta es similar a lo que tenían que realizar los participantes dentro del software, por lo que se esperaba un mayor porcentaje de respuestas correctas, analizando las respuestas erróneas, se aprecia que 2 participantes hicieron un arreglo de tipo int en vez de tipo char, mientras que el resto le faltaban datos al arreglo exceptuando a un participante que al parecer confundió la declaración de un String con la declaración de un arreglo.

Pregunta 9:

Tabla 9.24 Tabla de la pregunta 9 del posTest, usuarios sin saber Java

0	2
Null	10
Contenido basura	7
'a'	0
Sin responder	1

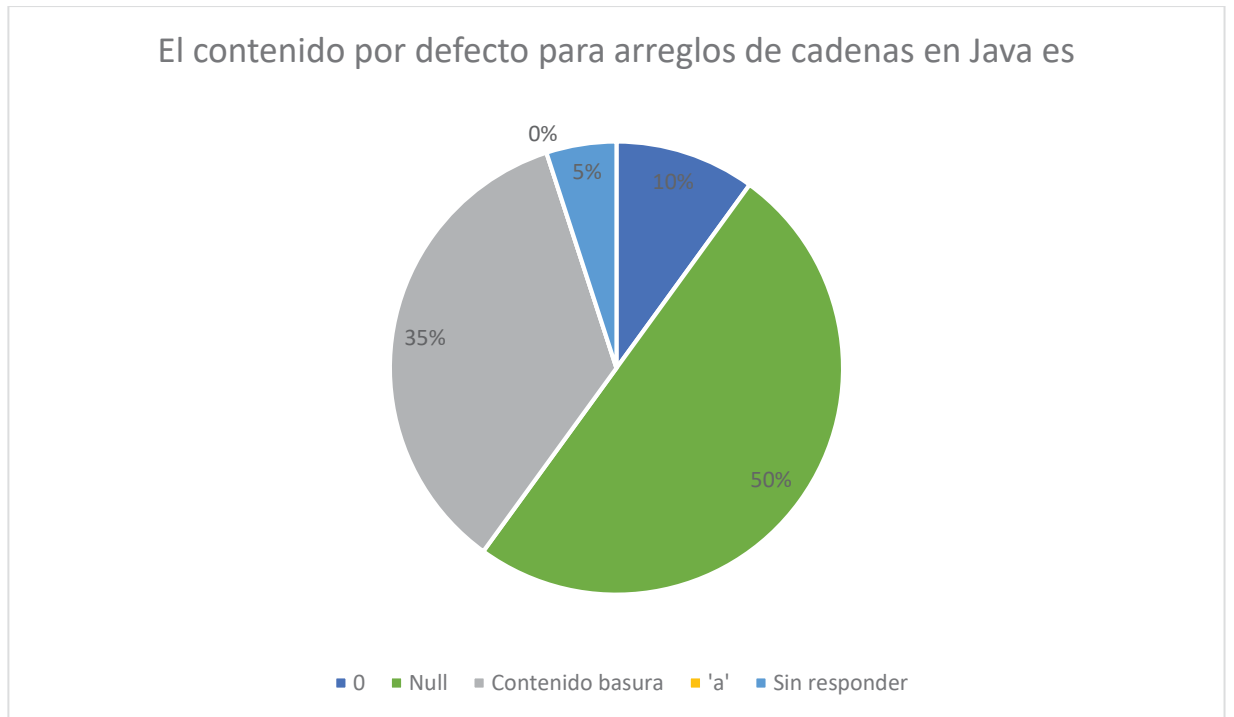


Figura 9.26 Grafico de la pregunta 9 del posTest, usuarios sin saber Java

Podemos notar una mejora ante esta pregunta que el PreTest se realizo de forma similar, teniendo como resultado que un 50% pensaba que contenía contenido basura, mientras que después de la realización del test del software, aclararon sus dudas el 50%

Pregunta 10:

Tabla 9.25 Tabla de la pregunta 10 del posTest, usuarios sin saber Java

Nombre_del_array [ ][ ] = Tipo_de_variable[dimensión];	0
New Tipo_de_variable[ ] Nombre_del_array = Tipo_de_variable[dimensión];	1
Tipo_de_variable Nombre_del_array [] = Tipo_de_variable[dimensión];	3
Tipo_de_variable[ ] Nombre_del_array = new Tipo_de_variable[dimensión];	16

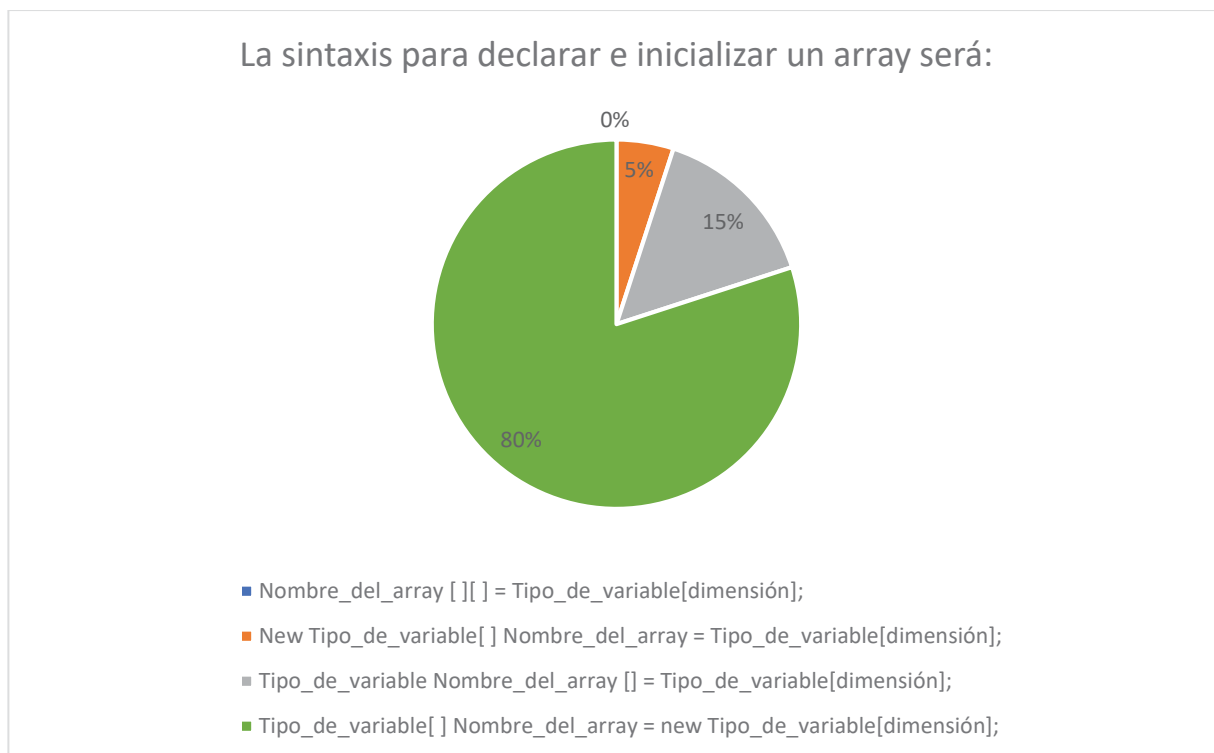


Figura 9.27 Grafico de la pregunta 10 del posTest, usuarios sin saber Java

Se puede observar que el 80% respondió esta pregunta de manera correcta por lo que se da como hipótesis de que el 20% que respondió erróneo es el mismo que respondió de manera incorrecta la declaración de un arreglo tipo char anteriormente.

Pregunta 11:

Tabla 9.26 Tabla de la pregunta 11 del posTest, usuarios sin saber Java

Excepción	0
Variable de control	2
Método	18
Funcion Main()	0

La definición "conjunto de instrucciones definidas dentro de una clase, que realizan una determinada tarea y a las que podemos invocar mediante un nombre " corresponde a :

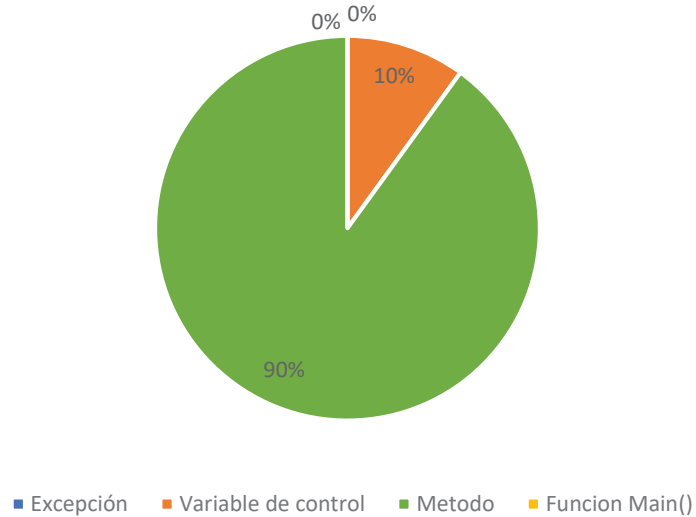


Figura 9.28 Grafico de la pregunta 11 del posTest, usuarios sin saber Java

Casi el 100% de los participantes responde de manera correcta esta pregunta, dejando en evidencia de que se explico de correctamente este tema en el software educativo pero que se puede mejorar.

Pregunta 12:

Tabla 9.27 Tabla de la pregunta 12 del posTest, usuarios sin saber Java

Su retorno es variable según los parámetros de entrada	1
Su retorno es variable y no depende de los parámetros de entrada	2
No posee retorno	17
Es una función con retorno indefinido	0

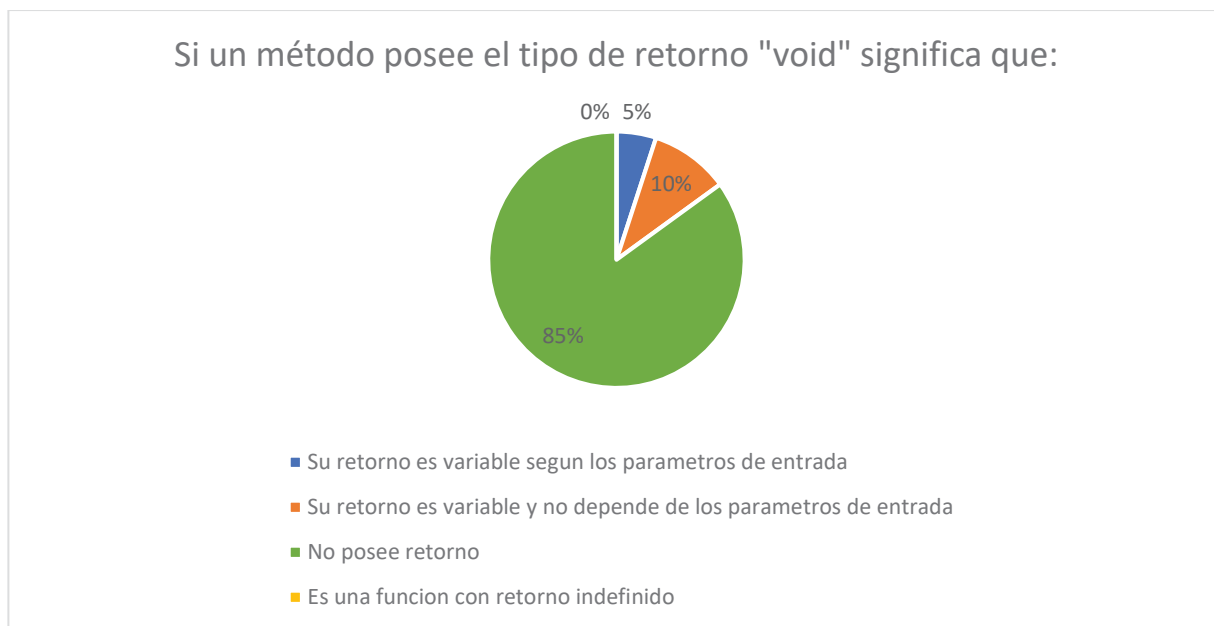


Figura 9.29 Grafico de la pregunta 13 del posTest, usuarios sin saber Java

Como conclusión se debería mejorar la forma de enseñar este concepto ya que es un concepto fácil de entender, sin embargo hubieron participantes que tomaban como respuesta la equivocada.

Pregunta 13:

Tabla 9.28 Tabla de la pregunta 13 del posTest, usuarios sin saber Java

Se puede instanciar un objeto a de la clase MiClase	15
Siempre provoca un error en el compilador	2
Siempre se provoca un error por falta de argumentos	2
Sin responder	1

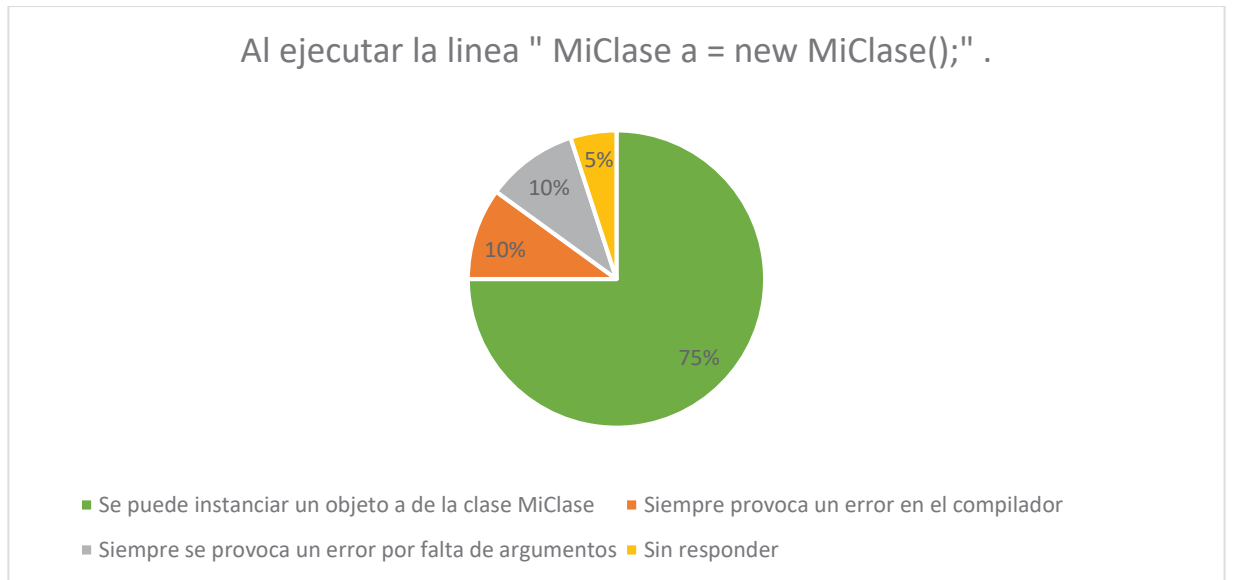


Figura 9.30 Grafico de la pregunta 14 del posTest, usuarios sin saber Java

Nuevamente nos encontramos con que un estudiante se quiso abstener de responder, por lo que los resultados no son 100% correcto, dejando dudas de lo que podría haber contestado dicho estudiante. Por otro lado nos encontramos con que el 75% de los participantes respondió de manera correcta. Como hipótesis se puede suponer que los estudiantes que respondieron de manera incorrecta se debe a que no alcanzaron a llegar al nivel donde que tenían que instanciar una clase.

#### Pregunta 14:

Tabla 9.29 Tabla de la pregunta 14 del posTest, usuarios sin saber Java

private char cambiaLetra (char letra);	1
public int saludar(int hola);	1
public int cambiaLetra(char letra);	18
public void cambiaLetra(char letra);	0

Cuál sintaxis corresponde a la cabecera de un método que tiene por nombre cambiaLetra, modificador de visibilidad público, retorna un valor tipo entero y recibe por parámetro un carácter.

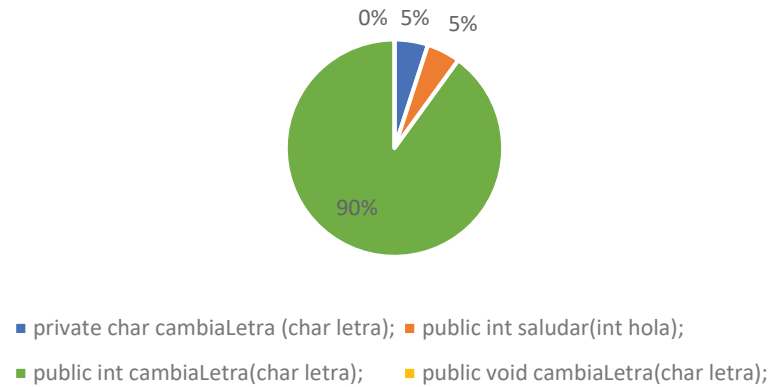


Figura 9.31 Grafico de la pregunta 15 del posTest, usuarios sin saber Java

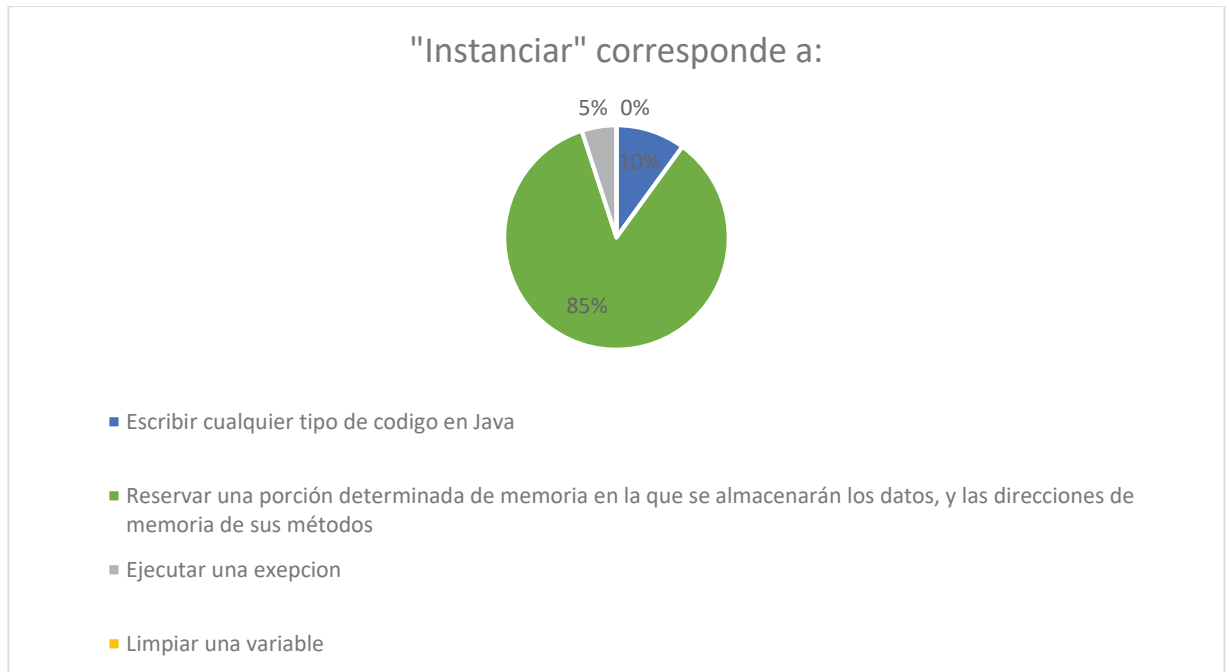
El 90% de los participantes responde de manera correcta dejando en suposición de que un alumno respondió de manera aleatoria, mientras que el otro alumno que respondió de manera incorrecta no tenía clara los conceptos de visibilidad, como se señaló anteriormente.

Pregunta 15:

Tabla 9.30 Tabla de la pregunta 15 del posTest, usuarios sin saber Java

Escribir cualquier tipo de código en Java	2
Reservar una porción determinada de memoria en la que se almacenarán los datos, y las direcciones de memoria de sus métodos	17
Ejecutar una exepcion	1
Limpiar una variable	0





Se aprecia que el 85% de los estudiantes pudo responder de manera correcta lo que es instanciar, por lo que el concepto dentro del juego está bien explicado, pero puede mejorar.

### 9.2.1 Conclusiones de las Respuestas

Como conclusión las respuestas dadas por los usuarios han demostrado mejoras y dejado en claro que los conceptos no tratados dentro del software son en los que se ha visto mayor margen de errores. Por otro lado, el tiempo que se tuvo para realizar la prueba de software no fue suficientes para algunos alumnos, por lo que, influencia en las respuestas entregadas por los usuarios, ya que hay conceptos que no pudieron aprender dentro del software por no haber llegado al nivel correspondiente.

## 9.3 Comparación PreTest y PosTest por Temáticas

### 9.3.1 Temática: Clases

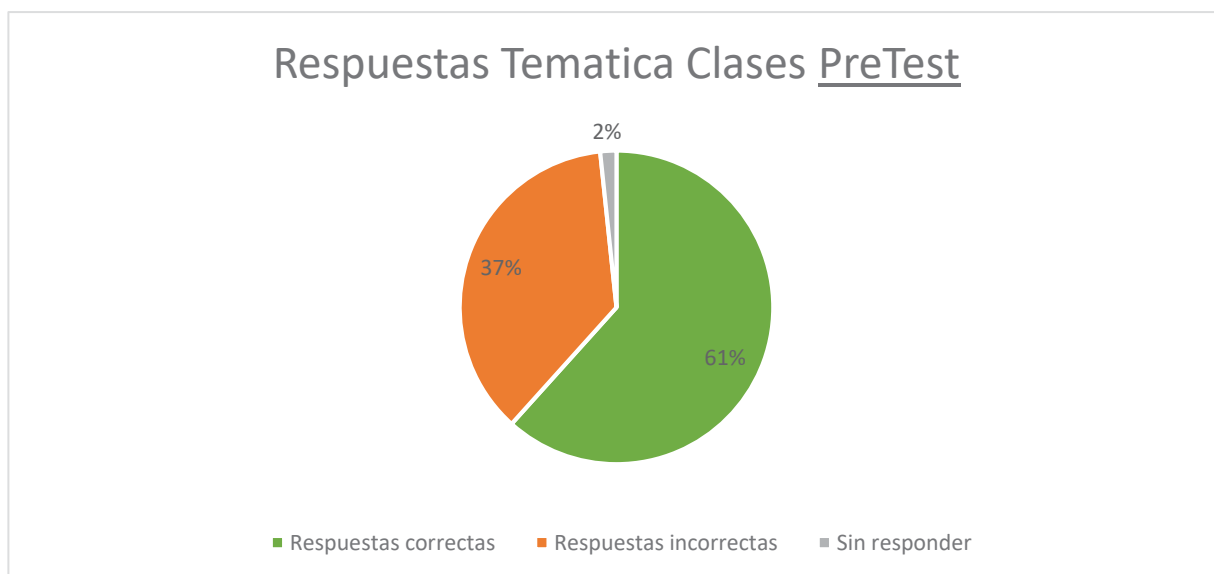


Figura 9.32 Grafico temática clases preTest, participantes sin saber Java

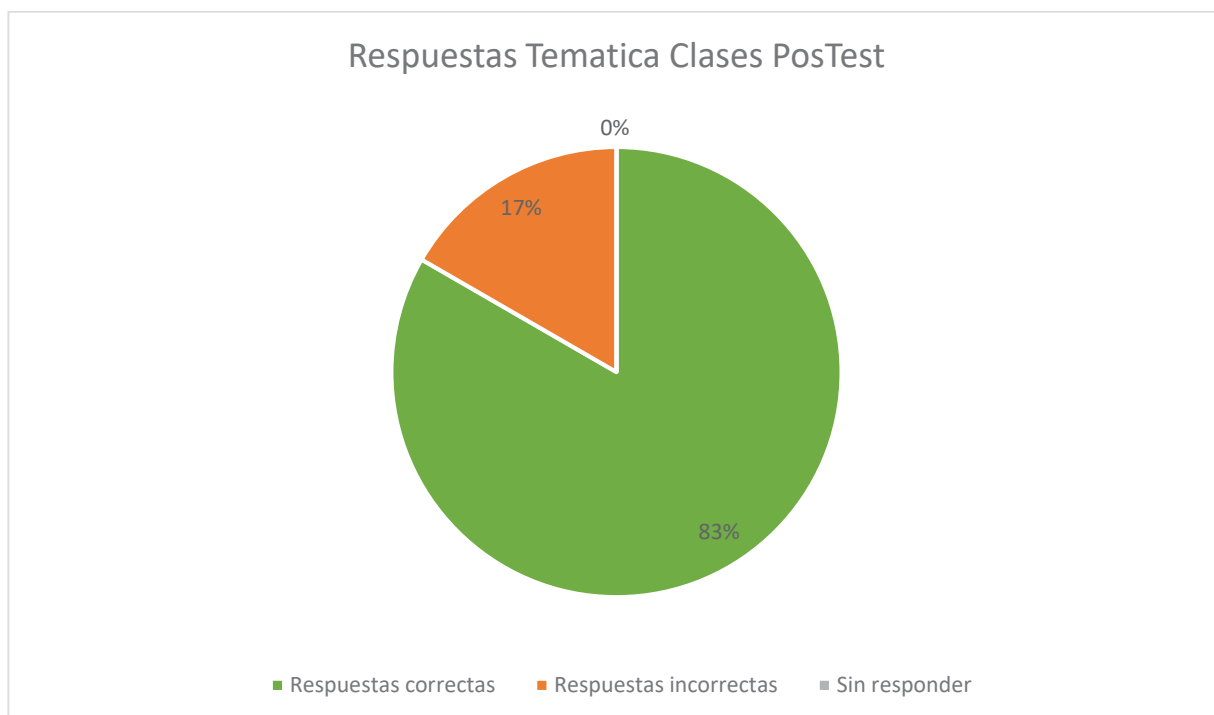


Figura 9.33 Grafico temática clases posTest, participantes sin saber Java

A pesar de que se esperaba que la mayoría del curso respondiera de manera errónea el PreTest debido a que no tenían conocimientos previos de la materia propuesta, de igual manera se logró conseguir que más del 50% respondiera de manera correcta. Haciendo una comparación

entre los dos test se ha reflejado una mejora del 22% después de la prueba del software educativo, por lo que, se concluye que el software educativo cumple su función de enseñar contenidos de Java para la Programación Orientada a Objetos.

### 9.3.2 Temática: Arreglos

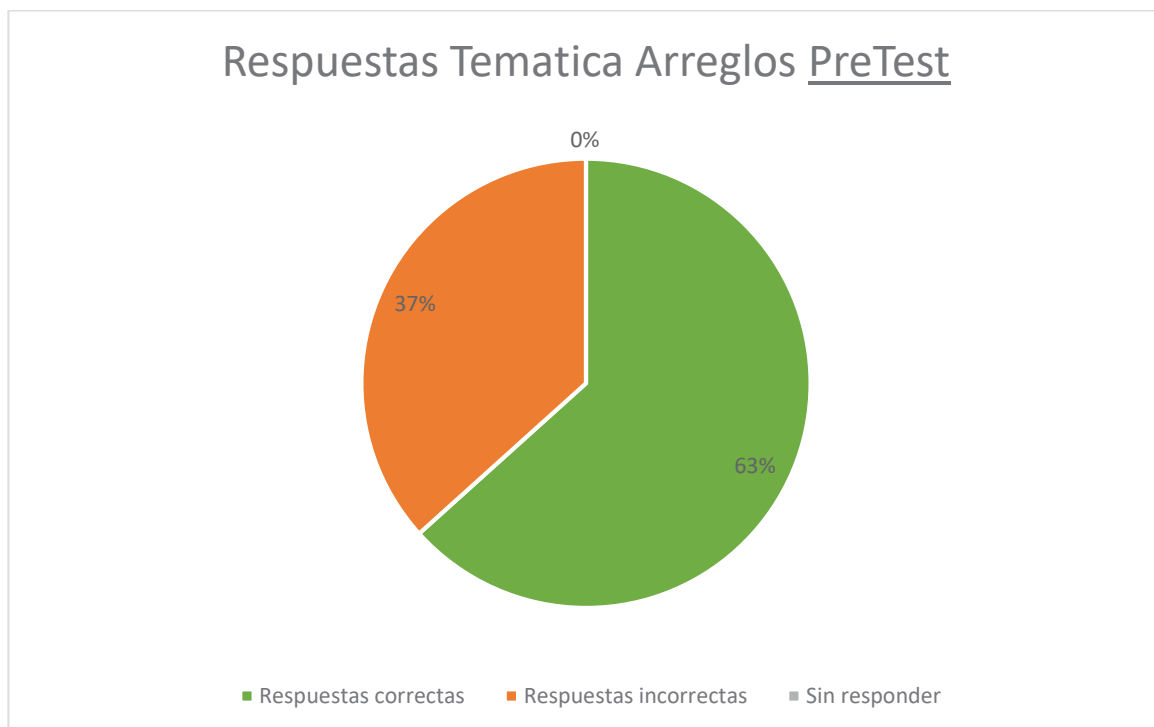


Figura 9.34 Grafico temática arreglos preTest, participantes sin saber Java

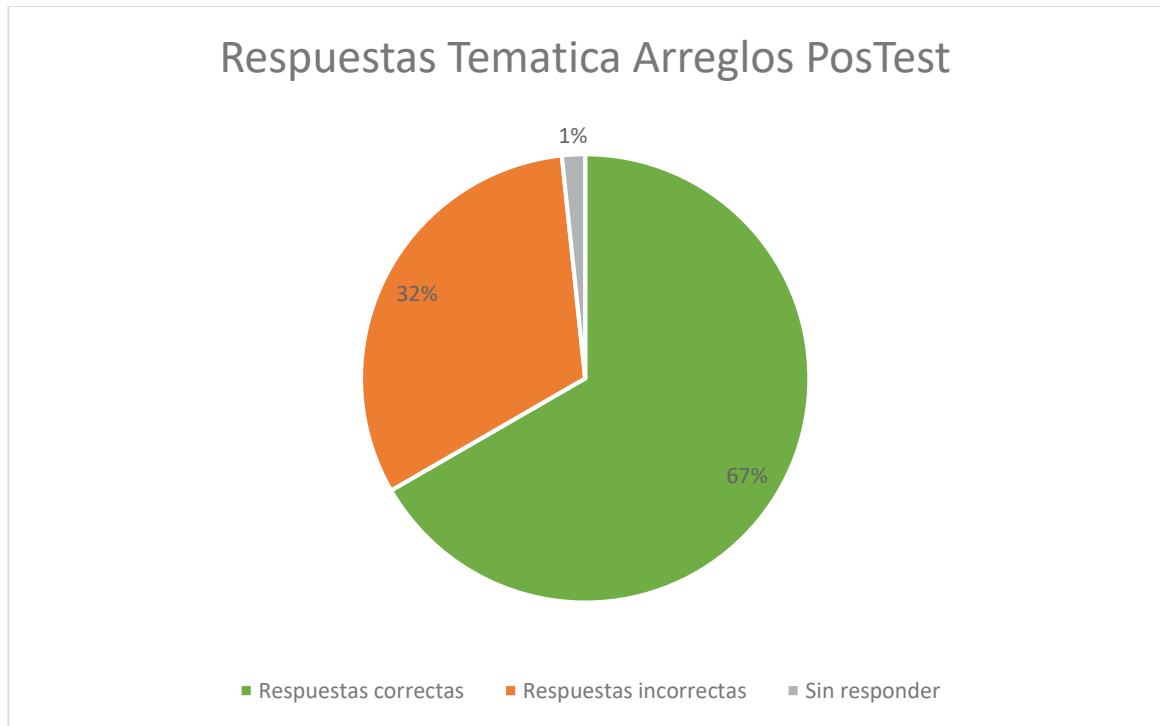


Figura 9.35 Grafico temática arreglos posTest, participantes sin saber Java

En esta temática hubo una mejora de un 4% se presume que es debido a que los alumnos ya tenían conocimientos previos de este contenido debido a que en el lenguaje de programación en C es similar la manera de declarar arreglos que en Java, sin embargo es muy bajo el porcentaje de mejoras, además de que un participante se abstuvo de responder alguna pregunta de esta categoría, por lo que hay que revisar como se están planteando estos contenidos dentro del software para obtener mejor porcentaje de mejoras.

### 9.3.3 Temática: Métodos

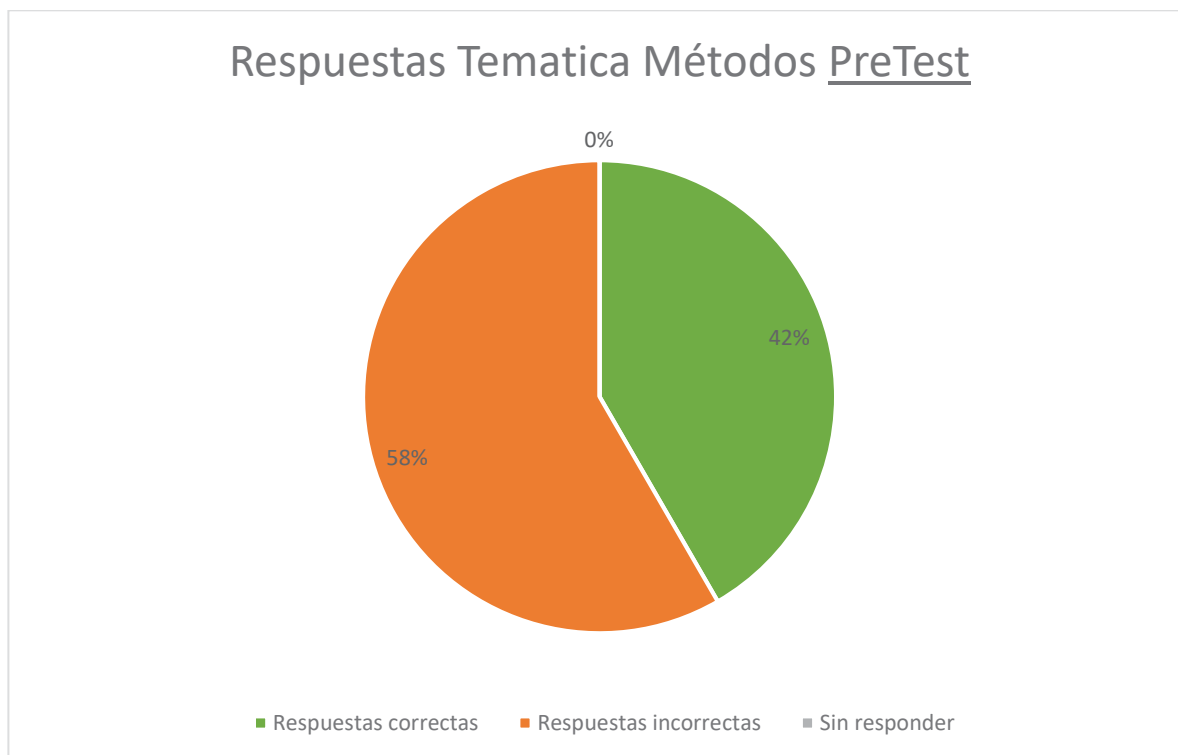


Figura 9.36 Grafico temática métodos preTest, participantes sin saber Java

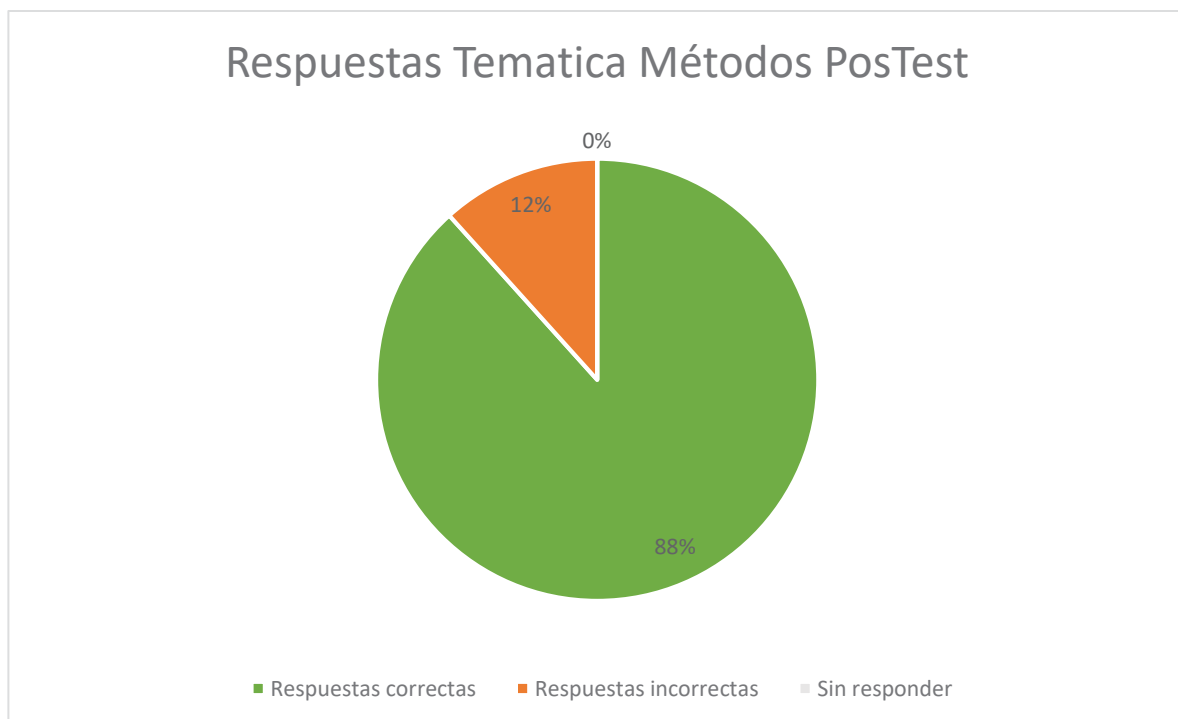


Figura 9.37 Grafico temática métodos posTest, participantes sin saber Java

Se observa que hay mejoras de un 46% con respecto a la temática métodos, lo cual refleja que los estudiantes llegaron al PreTest con poco conocimiento de la materia exhibida y luego de probar el software educativo aumentaron su conocimiento con respecto a este contenido.

#### 9.3.4 Temática: Atributos

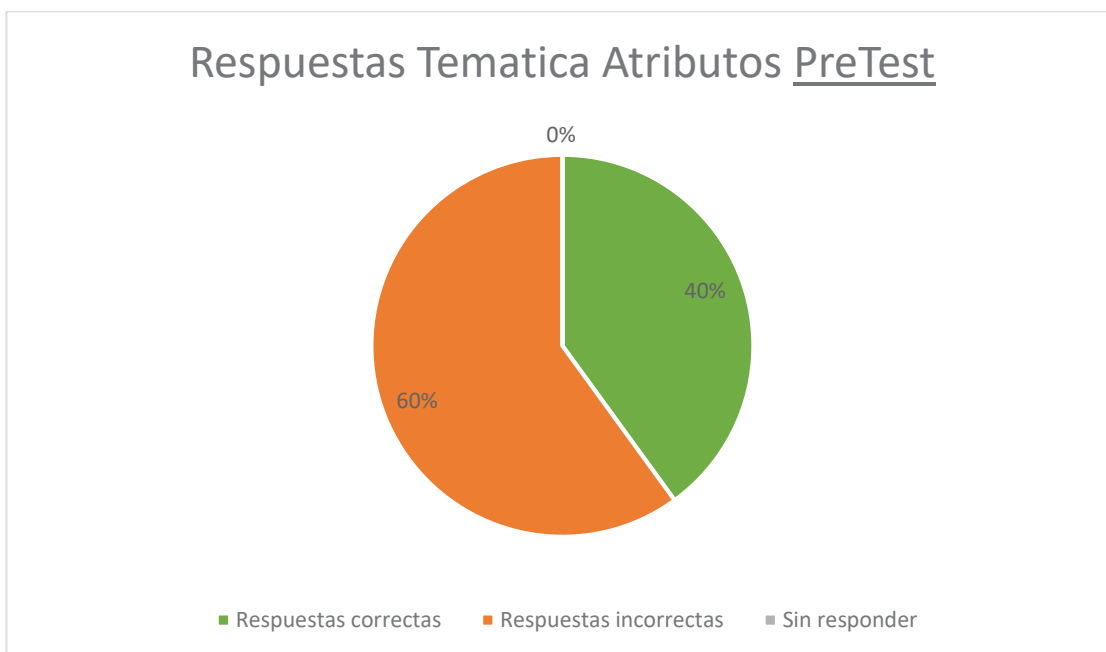


Figura 9.38 Grafico temática atributos preTest, participantes sin saber Java

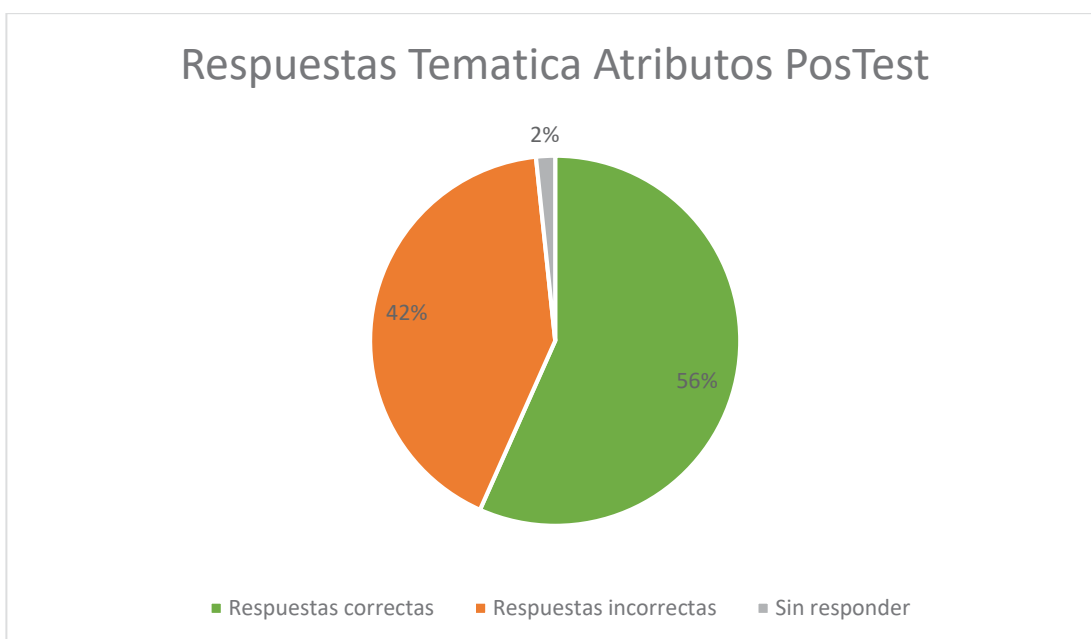


Figura 9.39 Grafico temática métodos posTest, participantes sin saber Java

Con respecto a esta tematica, hay que considerar que esta tiene incluido la pregunta con los modificadores de visibilidad que no fueron enseñados en el software educativo por lo que la gran mayoría de los estudiantes la tuvo erroneo. Por lo que, se observa que hubo un aumento del 16% y un 2% que se abstino a contestar, afectando al resultado la falta de tutoriales en el software educativo que tenga que ver con el topico modificadores de visibilidad. Se concluye que se debe modificar el software educativo agregando este contenido para obtener mayor porcentaje en mejoras.

### 9.3.5 Temática: Objetos

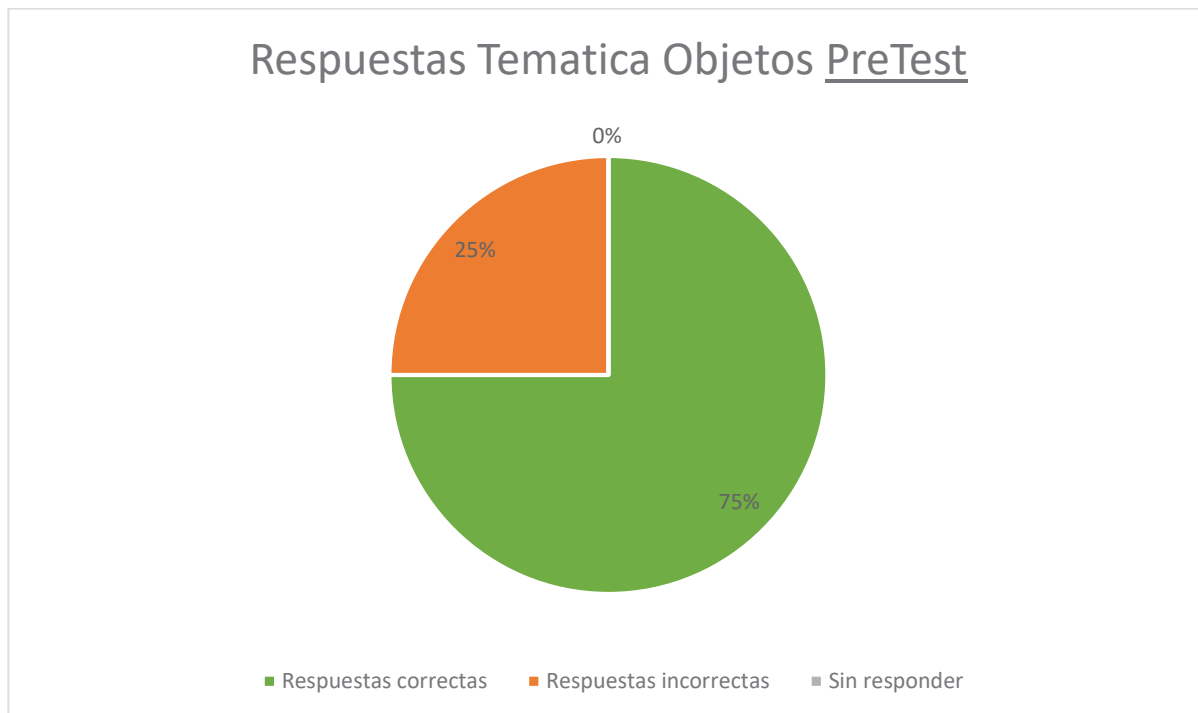


Figura 9.40 Grafico temática objetos preTest, participantes sin saber Java

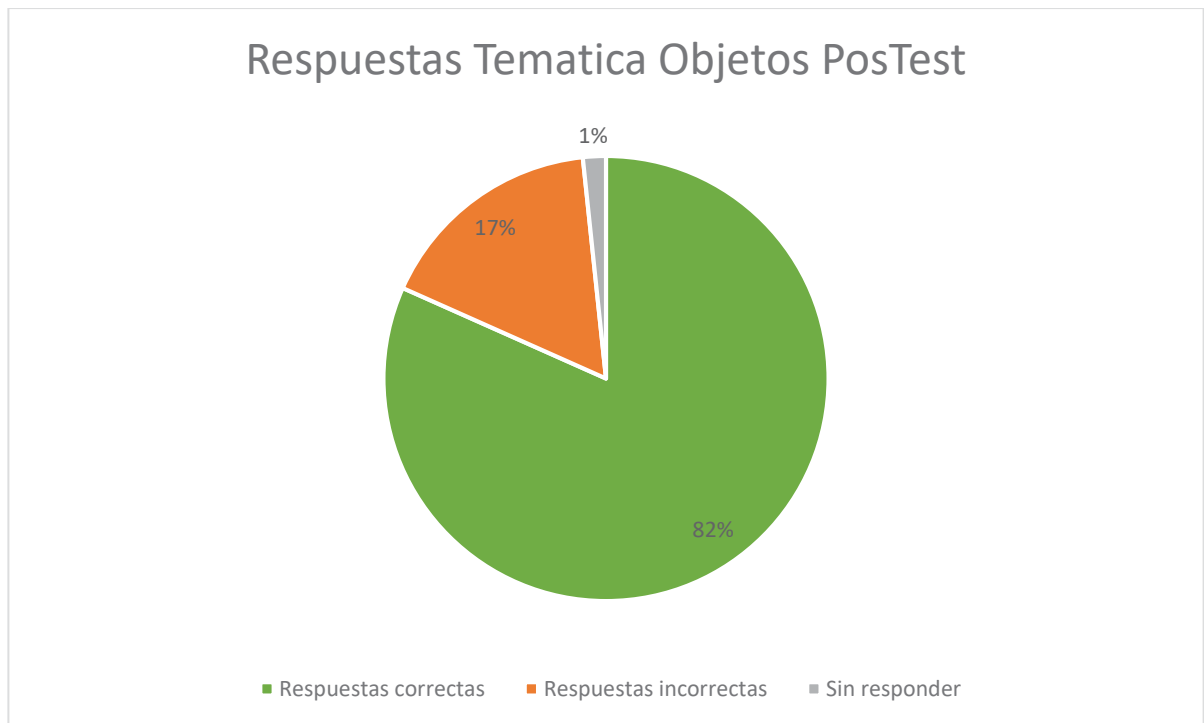


Figura 9.41 Grafico temática objetos posTest, participantes sin saber Java

Se observa un 7% de mejora y una abstinencia con respecto a esta temática, se presume que los resultados fueran afectados por la falta de tiempo en terminar de realizar el juego, por lo que algunos estudiantes no vieron los tutoriales con respecto a esta temática, de igual forma se observa de que los participantes tenían conocimientos previos de esta temática.



## 10 Pruebas de Usuario con Conocimientos Previos a *POO*

Las pruebas se realizaron en el laboratorio de la escuela de Ingeniería en informática de la *PUCV* se pretendía tomar las pruebas a los estudiantes que estaban cursando *POO* este semestre, pero se tuvo un percance con la configuración del JDK en los computadores de la escuela, por lo que no pudieron ser efectivas las pruebas, realizando las pruebas a tres alumnos del ramo *POO* por falta de computadores funcionales. Luego de tener solucionado el problema se busco gente que quisiera probar el software de manera voluntaria y que ya hubiera cursado el ramo de *POO*. Bajo estas circunstancias se consiguió tener 7 sujetos de pruebas por lo que estas pruebas no permiten sacar grandes conclusiones.

A diferencia del anterior este conto con el tiempo que tuvieran los participantes en realizar las pruebas.

En estas pruebas se tomo grupo de control y grupo experimental, pero por problemas de que es muy poca la cantidad de participantes se decidió analizar las pruebas sin necesidad de separos.

### 10.1 Respuestas de los Participantes por cada Pregunta PreTest

Las respuestas correctas se reflejarán de color verde en el gráfico de torta.

Pregunta 1:

Tabla 10.1 Tabla pregunta 1 preTest, participantes que sabían Java.

Es un concepto similar al de 'array'	0
Es un tipo particular de variable	0
Es un modelo o plantilla a partir de la cual creamos objetos	6
Es una categoría de datos ordenada secuencialmente	1

¿Cuál es la descripción que crees que define mejor el concepto 'clase' en la programación orientada a objetos?

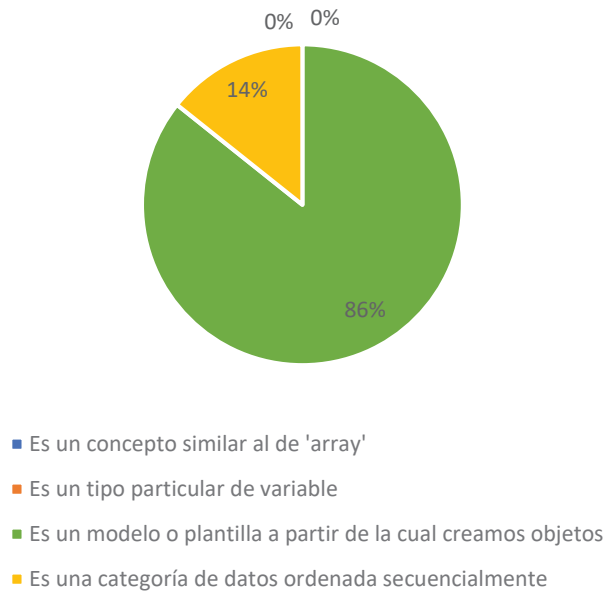


Figura 10.1 Grafico pregunta 1 preTest, participantes que sabían Java.

Pregunta 2:

Tabla 10.2 Tabla pregunta 2 preTest, participantes que sabían Java.

Publico, Privado y mixto	0
public,private,protected y por default	0
import, package, java.util	0
public, private, protected	7

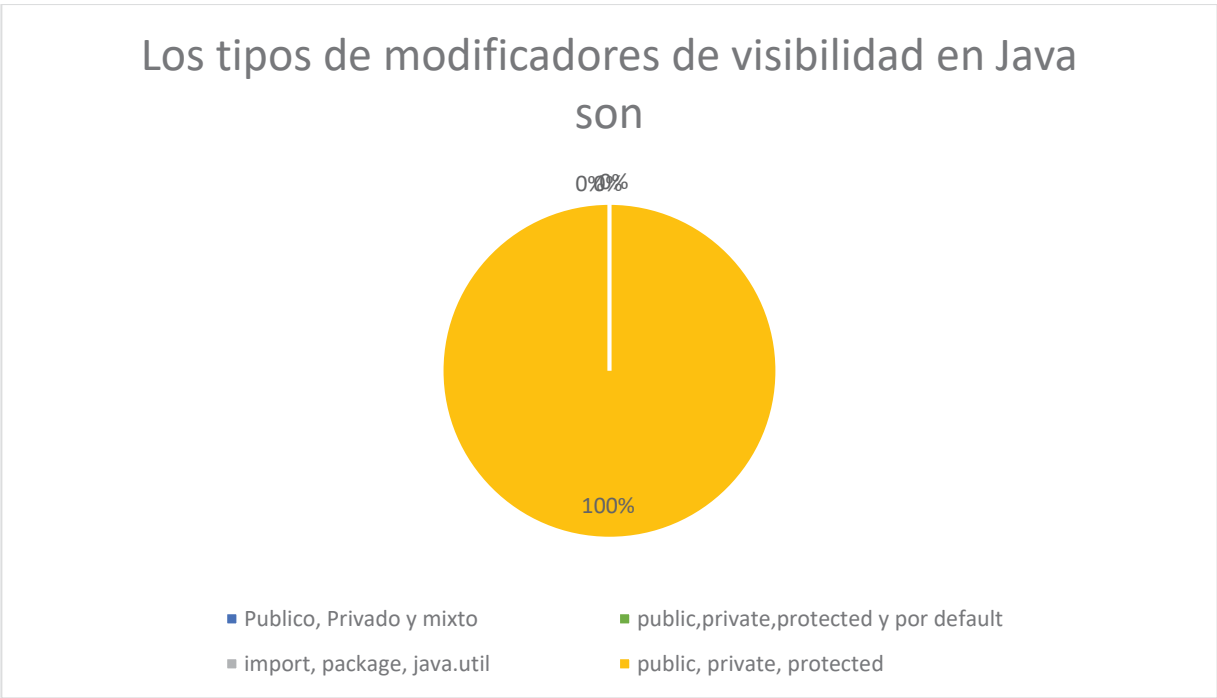


Figura 10.2 Grafico pregunta 2 preTest, participantes que sabían Java.

Pregunta 3:

Tabla 10.3 Tabla pregunta 3 preTest, participantes que sabían Java.

Sus cardinalidades y su tipo	1
Sus atributos y sus métodos	6
La forma en que establece comunicación e intercambiar mensajes	0
Su interfaz y los eventos asociados	0

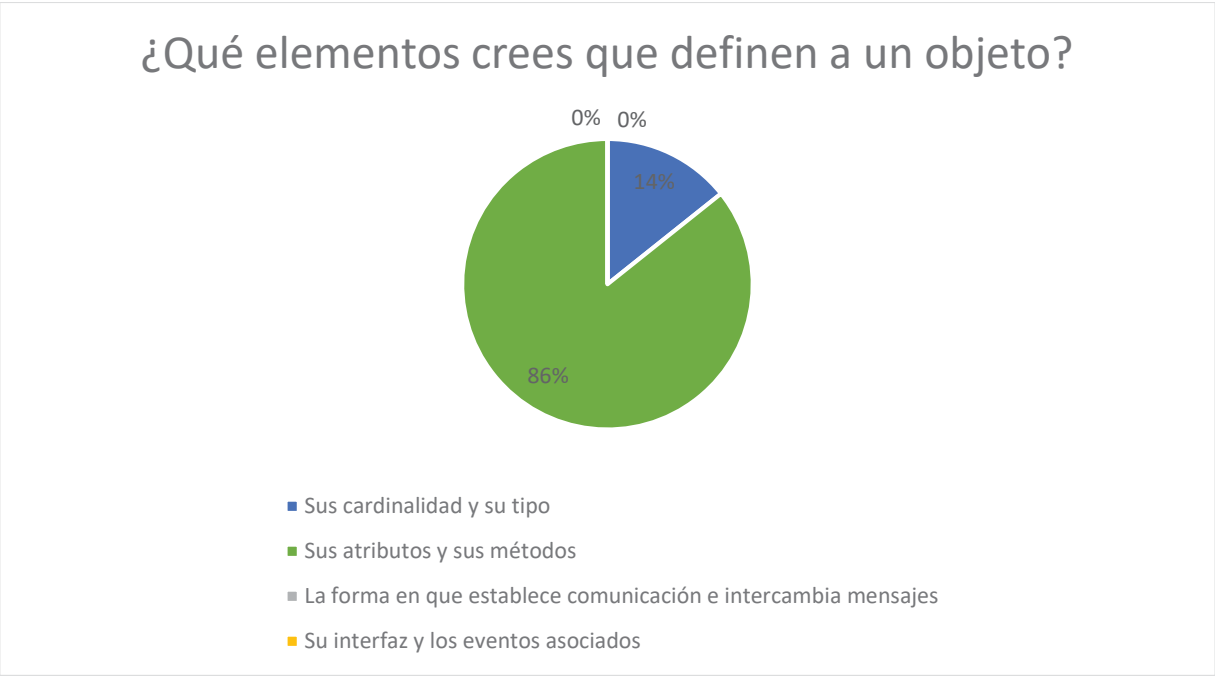


Figura 10.3 Grafico pregunta 3 preTest, participantes que sabían Java.

Pregunta 4:

Tabla 10.4 Tabla pregunta 4 preTest, participantes que sabían Java.

Duplicar una clase	0
Eliminar una clase	0
Crear un objeto a partir de la clase	7
Conectar dos clases entre sí	0



Figura 10.4 Grafico pregunta 4 preTest, participantes que sabían Java.

Pregunta 5:

Tabla 10.5 Tabla pregunta 5 preTest, participantes que sabían Java.

saludar();	0
persona->saludar();	0
persona.saludar();	6
ninguna de las anteriores	1

Desde una clase externa, la llamada del método "saludar()", que posee el objeto "persona" se realiza de la forma:

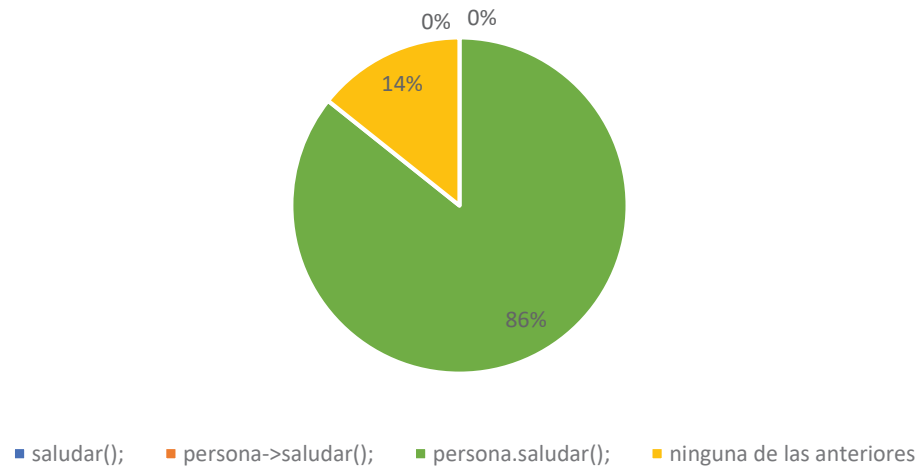


Figura 10.5 Grafico pregunta 5 preTest, participantes que sabían Java.

Pregunta 6:

Tabla 10.6 Tabla pregunta 6 preTest, participantes que sabían Java.

Clase	0
Herencia	0
Método	6
Casting	1



Figura 10.6 Grafico pregunta 6 preTest, participantes que sabían Java.

Pregunta 7:

Tabla 10.7 Tabla pregunta 7 preTest, participantes que sabían Java.

<code>int vector [] = new int [10];</code>	6
<code>int vector [] = new int [10]</code>	0
<code>int vector [][] = new int [5][5];</code>	0
<code>int vector = new int [10];</code>	1

## ¿Cual es la forma correcta de declarar un vector?

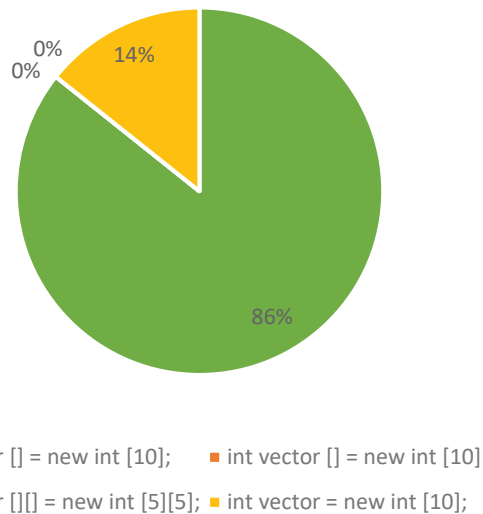


Figura 10.7 Grafico pregunta 7 preTest, participantes que sabían Java.

### Pregunta 8:

Tabla 10.8 Tabla pregunta 8 preTest, participantes que sabían Java.

Reloj, ObtenerHora, Fecha	2
MiClas, extens, int b	0
Lápiz, Pez, Reloj	5
abstract,assert,Hola	0



## Algunos ejemplos de nombres correctos para objetos son:

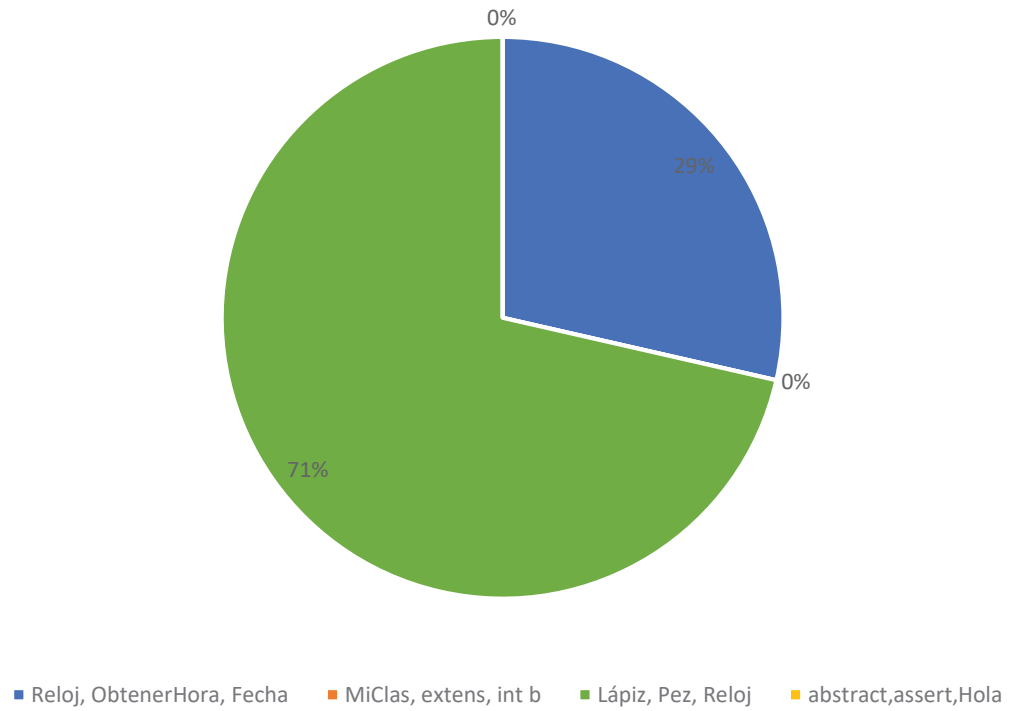


Figura 10.8 Grafico pregunta 8 preTest, participantes que sabían Java.

### Pregunta 9:

Tabla 10.9 Tabla pregunta 9 preTest, participantes que sabían Java.

Entero, punto flotante, logico, caracter	3
Entero,punto flotante, caracter	1
Númeroico, carácter, logico	1
Numerico, punto flotante, logico, string	2

## Java soporta los siguientes tipos de datos primitivos

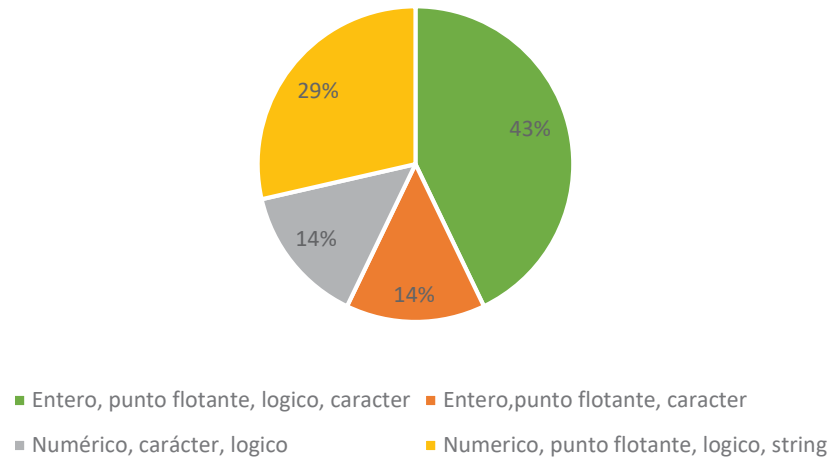


Figura 10.9 Grafico pregunta 9 preTest, participantes que sabían Java.

### Pregunta 10:

Tabla 10.10 Tabla pregunta 10 preTest, participantes que sabían Java.

Si	4
No	3

## Una clase siempre tiene un constructor por default

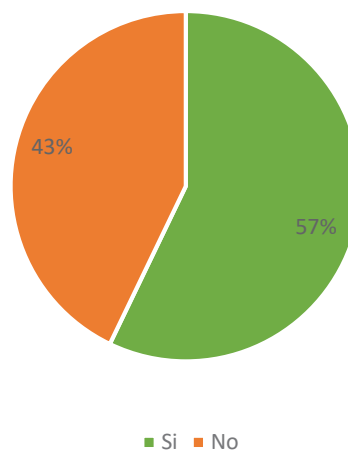


Figura 10.10 Grafico pregunta 10 preTest, participantes que sabían Java.

Pregunta 11:

Tabla 10.11 Tabla pregunta 11 preTest, participantes que sabían Java.

Nombrar con mas facilidad una clase	0
Utilizar métodos , clases y atributos predefinidos	6
Manejar excepciones	0
Importar clases del mismo package	1

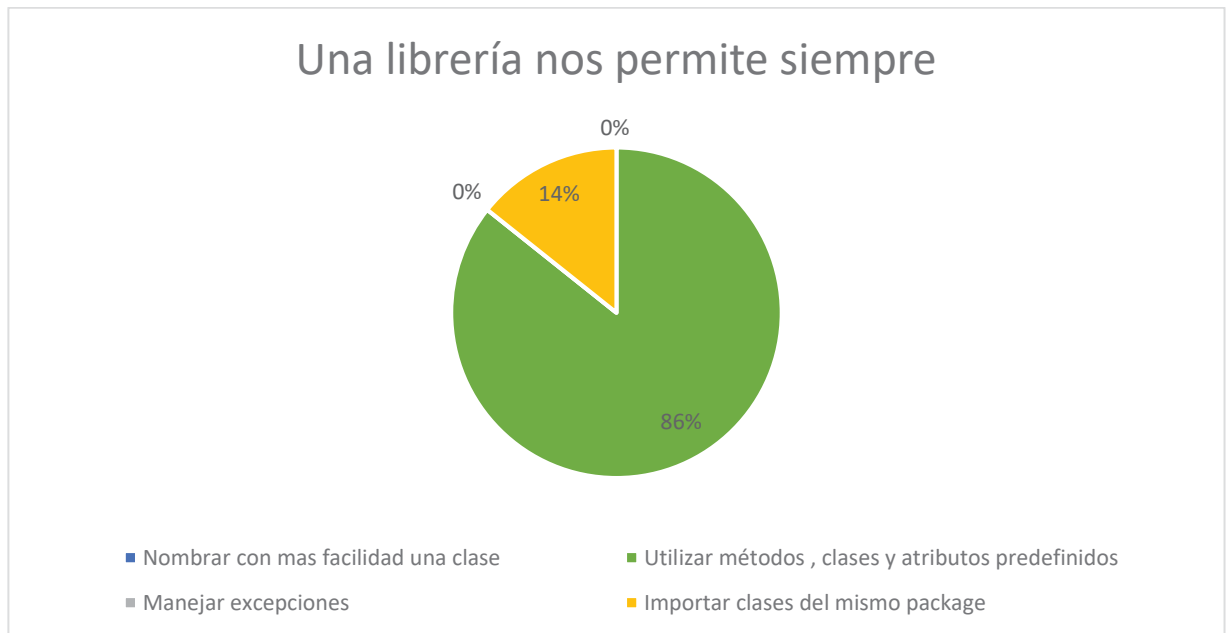


Figura 10.11 Grafico pregunta 11 preTest, participantes que sabían Java.

Pregunta 12:

Tabla 10.12 Tabla pregunta 12 preTest, participantes que sabían Java.

Si y depende del tipo de dato del arreglo	3
No, contiene contenido basura	4



Figura 10.12 Grafico pregunta 12 preTest, participantes que sabían Java.

Pregunta 13:

Tabla 10.13 Tabla pregunta 13 preTest, participantes que sabían Java.

Instanciando la clase	7
Agregando librerías a una clase	0
Haciendo sobrecarga de metodos	0
Creando una interfaz de usuario	0

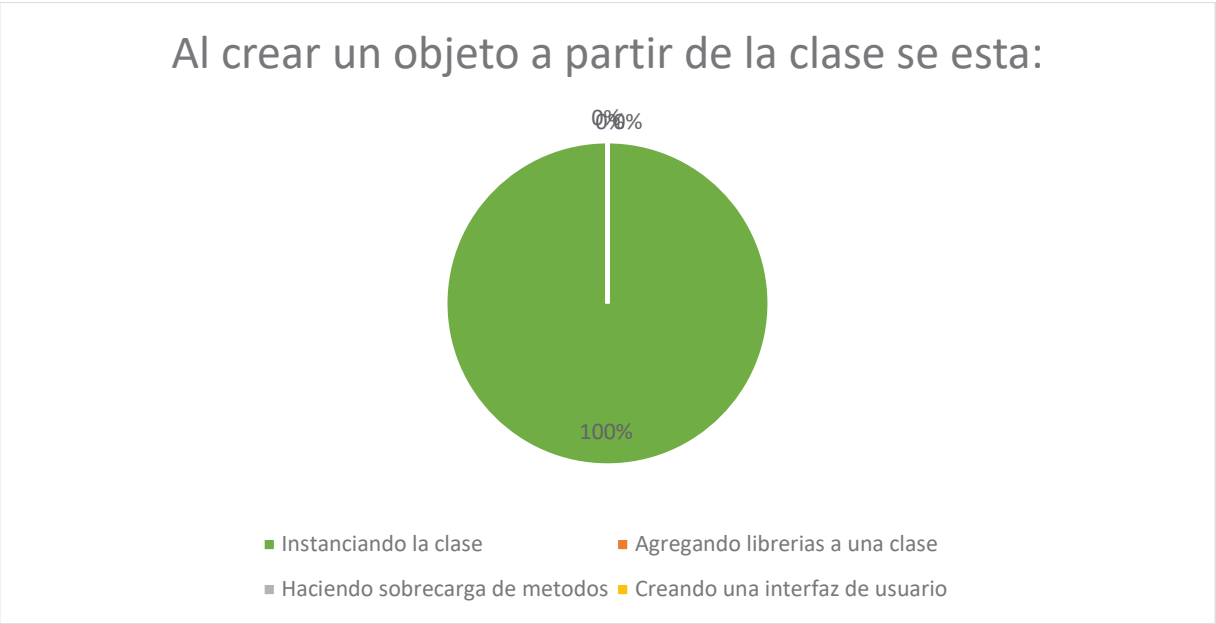


Figura 10.13 Grafico pregunta 13 preTest, participantes que sabían Java.

Pregunta 14:

Tabla 10.14 Tabla pregunta 14 preTest, participantes que sabían Java.

Sobre carga de un metodo	0
La sintaxis para declarar e inicializar un arreglo de String	6
La sintaxis para declarar e inicializar un arreglo de enteros	0
Un error de sintaxis	1

La sintaxis " String[ ] nombre = new String[4]; "  
corresponde a

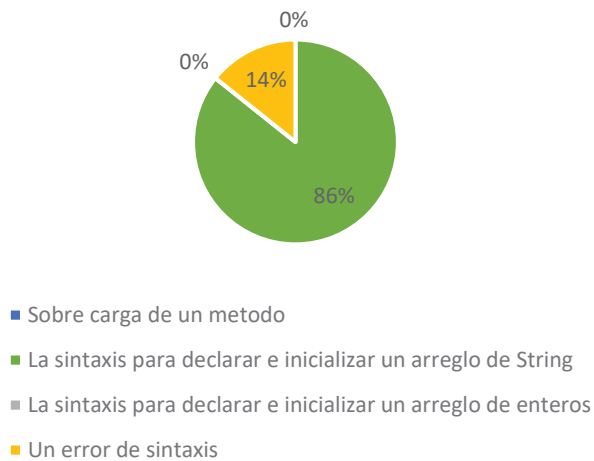


Figura 10.14 Grafico pregunta 14 preTest, participantes que sabían Java.

Pregunta 15:

Tabla 10.15 Tabla pregunta 15 preTest, participantes que sabían Java.

Error de compilación	0
Excepción	0
SobrEscritura de metodos	0
SobreCarga de metodos	7

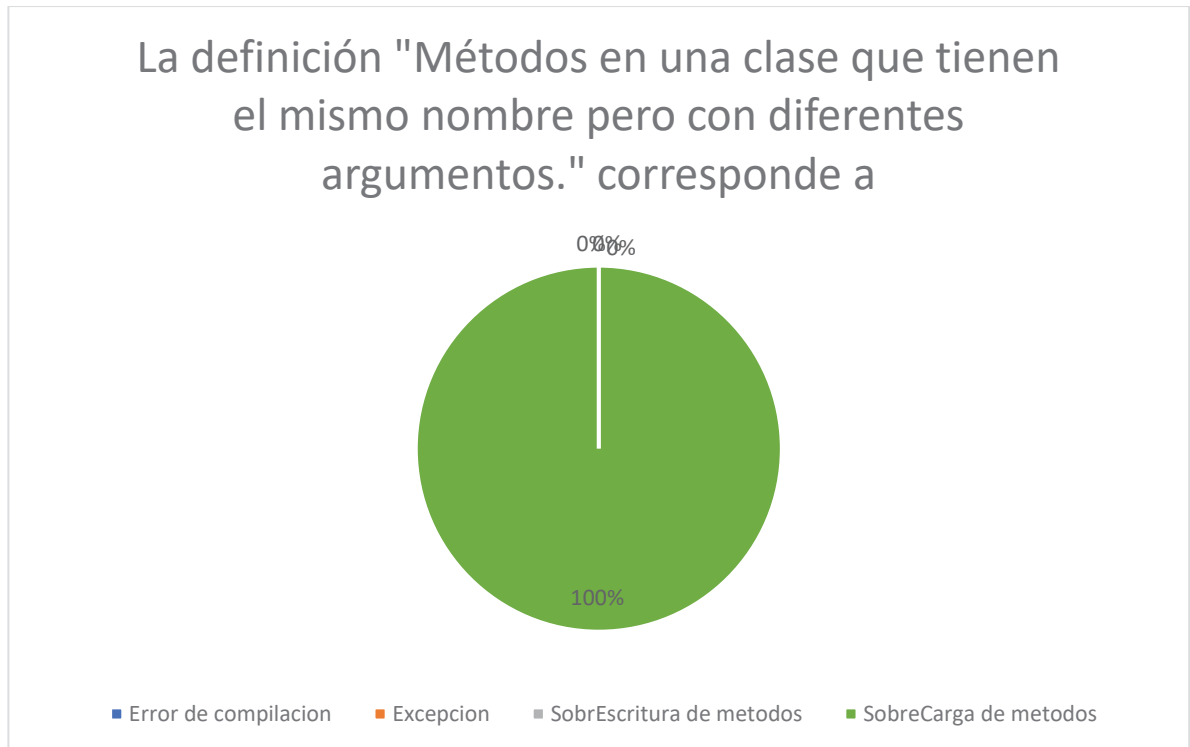


Figura 10.15 Grafico pregunta 15 preTest, participantes que sabían Java.

### 10.1.1 Conclusiones de las Respuestas

Con los pocos sujetos de pruebas que se lograron capturar, se puede apreciar que a pesar de que los alumnos ya habían visto estos conceptos con anterioridad, los estudiantes se equivocaron en varias preguntas, destacando que en una pregunta el 100% de los participantes se equivocó, también podemos destacar que en más de una pregunta el 100% de los participantes contestó de manera correcta. Se espera recibir mejoras en el posTest.

## 10.2 Respuestas de los Participantes por cada Pregunta PosTest

Las respuestas correctas estarán de color verde

Pregunta 1:

Tabla 10.16 Tabla pregunta 1 posTest, participantes que sabían Java.

Excepciones	0
Generar una instancia	0
Atributos primitivos	6
Datos enteros	0
Datos	1

La definición: "sirven para definir variables que contienen valores numéricos, lógicos y caracteres unitarios " corresponde a

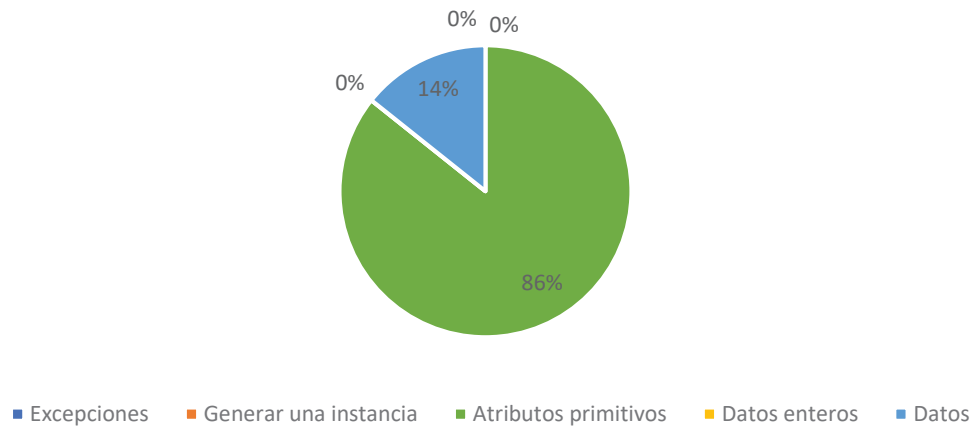


Figura 10.16 Grafico pregunta 1 posTest, participantes que sabían Java.

Pregunta 2:

Tabla 10.17 Tabla pregunta 2 posTest, participantes que sabían Java.

Comentarios	0
Metodo main	1
Clase	0
Todas las anteriores	6

La estructura principal de una aplicación en java contiene

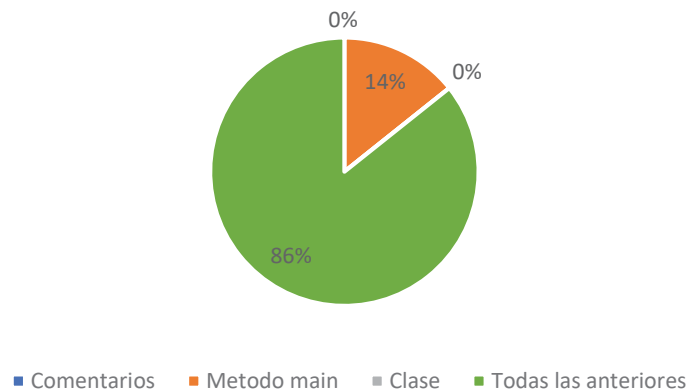


Figura 10.17 Grafico pregunta 2 posTest, participantes que sabían Java.



Pregunta 3:

Tabla 10.18 Tabla pregunta 3 posTest, participantes que sabían Java.

Variables de instancia	0
Constructor	0
Métodos	0
Todas las anteriores	7

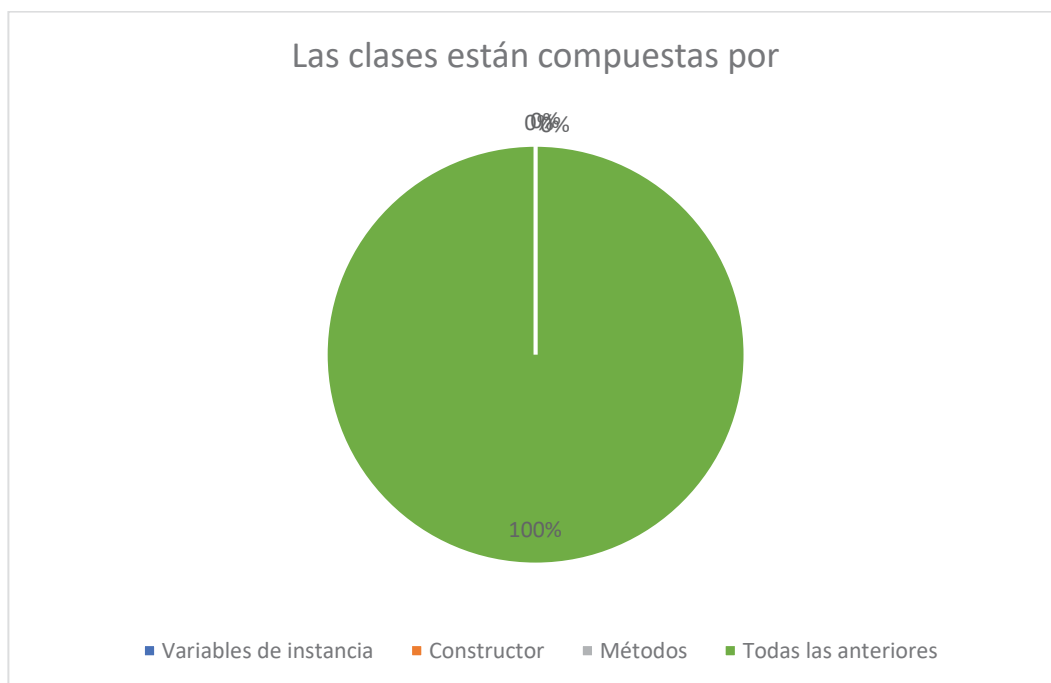


Figura 10.18 Grafico pregunta 3 posTest, participantes que sabían Java.

Pregunta 4:

Tabla 10.19 Tabla pregunta 4 posTest, participantes que sabían Java.

Una plantilla que define las variables y métodos a ser incluidas en un tipo de objeto específico	7
Un concepto similar a un vector	0
Una lista de objetos ordenados de forma no secuencial	0
Una categoria que ordena datos primitivos	0

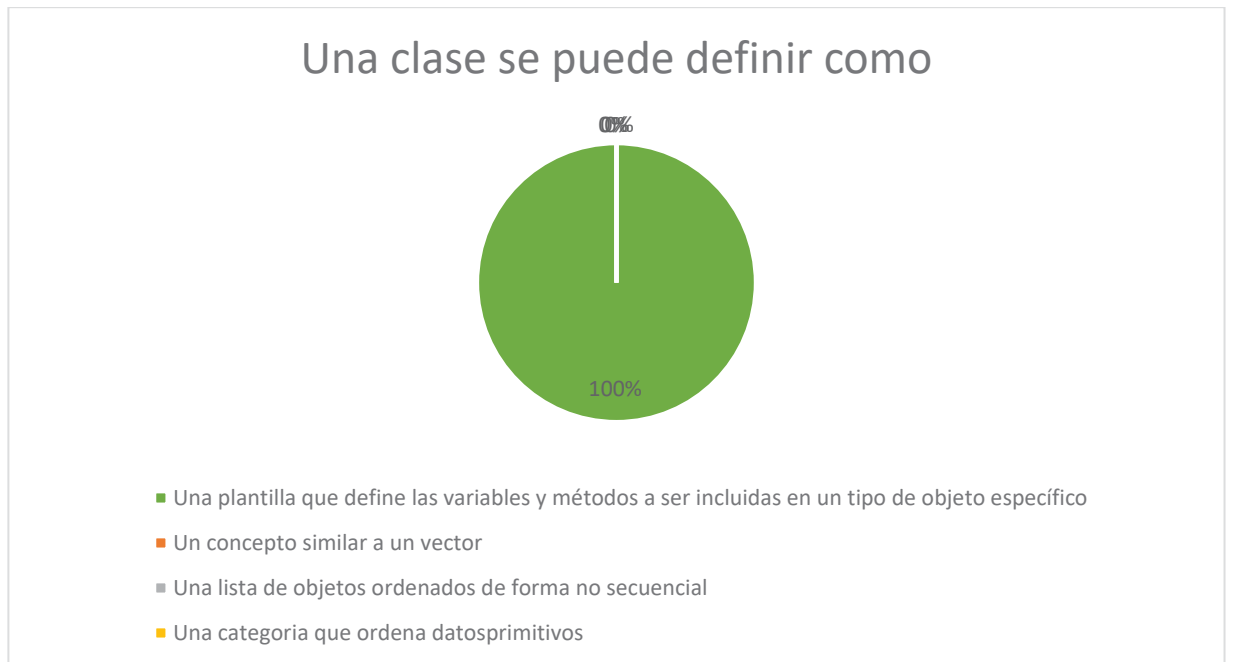


Figura 10.19 Grafico pregunta 4 posTest, participantes que sabían Java.

#### Pregunta 5:

Tabla 10.20 Tabla pregunta 5 posTest, participantes que sabían Java.

Tipos de excepciones	0
Clases de dominio	0
Modificadores de visibilidad en Java	7
Un tipo particular de variable	0

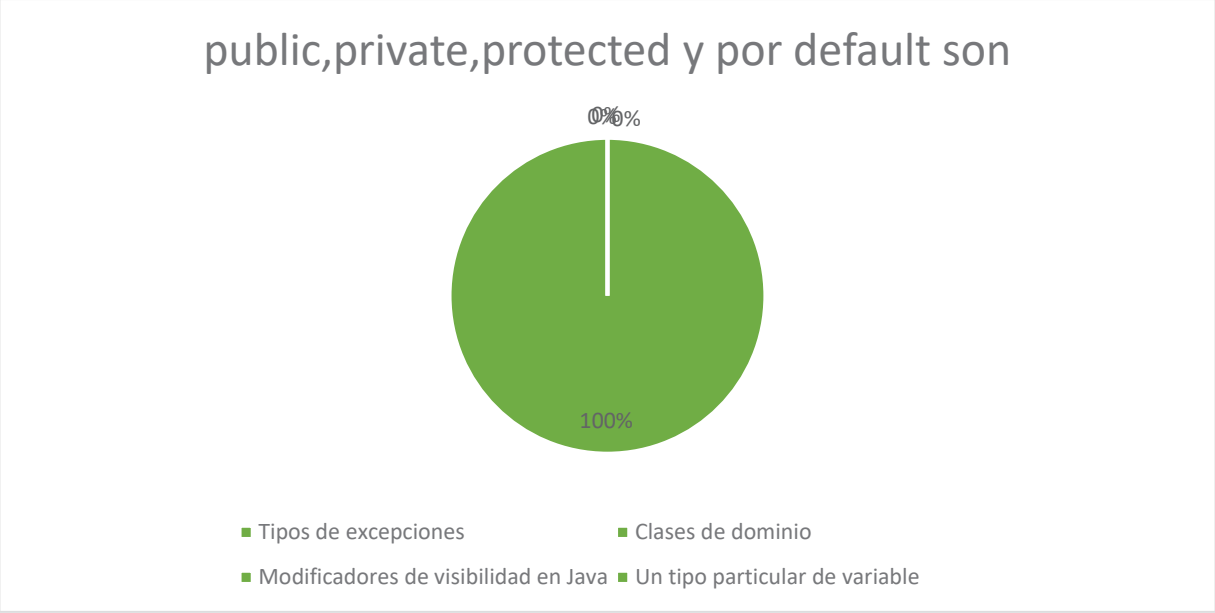


Figura 10.20 Grafico pregunta 5 posTest, participantes que sabían Java.

Pregunta 6:

Tabla 10.21 Tabla pregunta 6 posTest, participantes que sabían Java.

Duplicar la clase	0
Instanciar la clase	7
Eliminar la clase	0
Conectar la clase	0

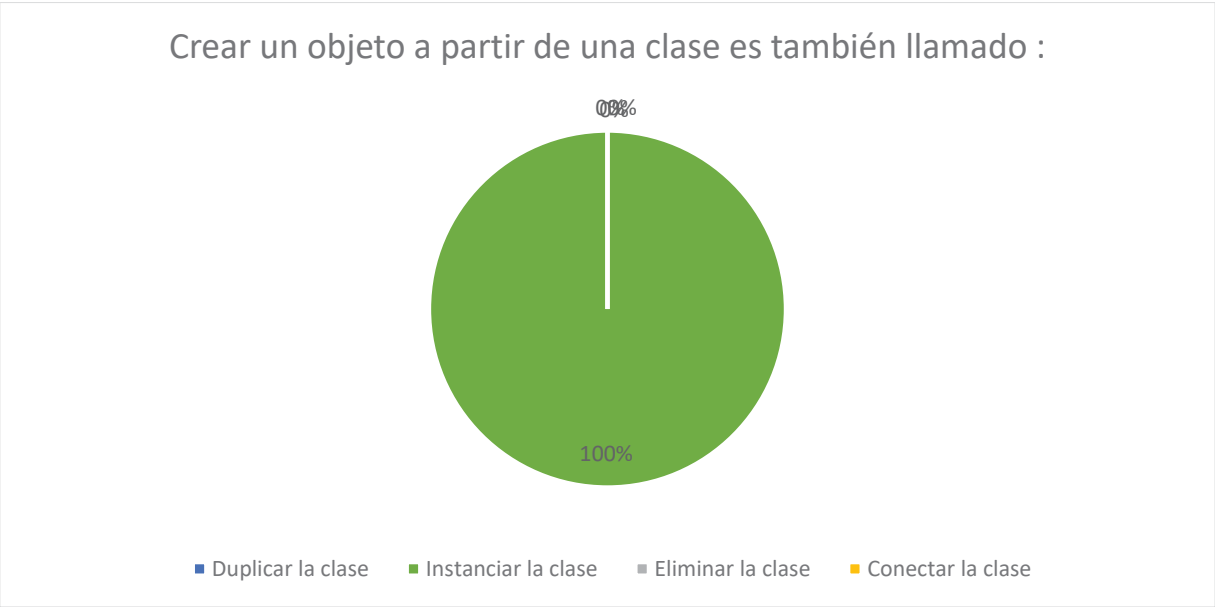


Figura 10.21 Grafico pregunta 6 posTest, participantes que sabían Java.

Pregunta 7:

Tabla 10.22 Tabla pregunta 7 posTest, participantes que sabían Java.

Respuestas correcta	5
Respuestas incorrectas	2

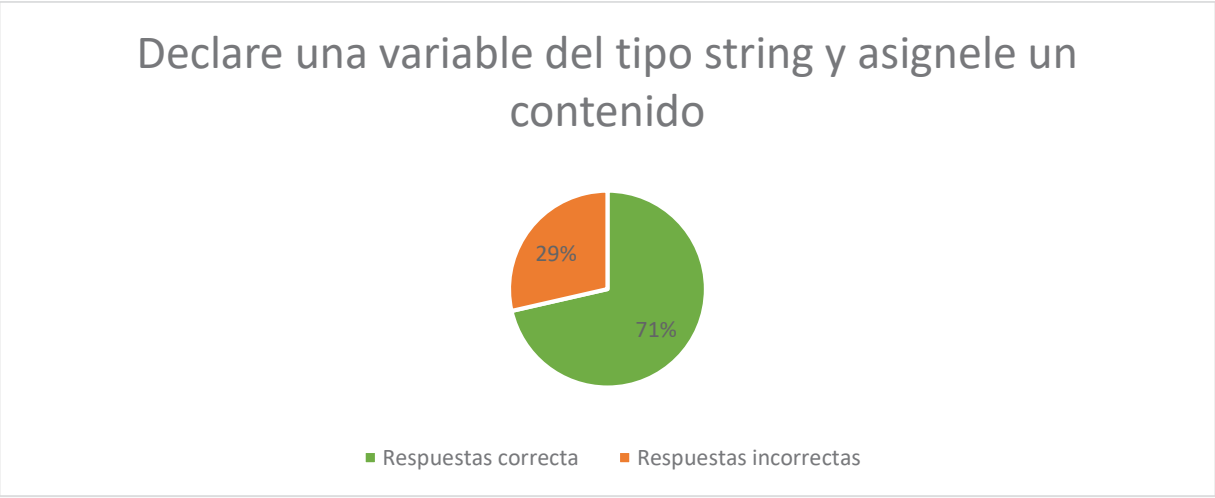


Figura 10.22 Grafico pregunta 7 posTest, participantes que sabían Java.

Pregunta 8:

Tabla 10.23 Tabla pregunta 8 posTest, participantes que sabían Java.

Respuestas correcta	3
---------------------	---

Respuestas incorrectas	4
------------------------	---

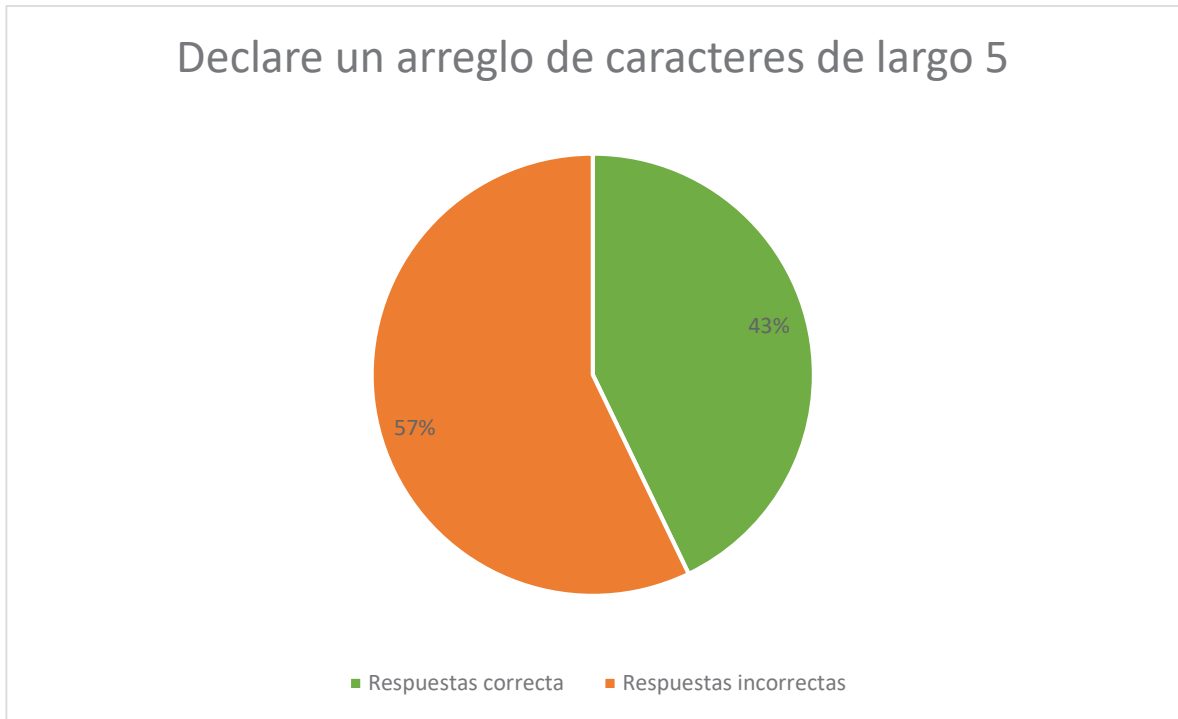


Figura 10.23 Grafico pregunta 8 posTest, participantes que sabían Java.

Pregunta 9:

Tabla 10.24 Tabla pregunta 9 posTest, participantes que sabían Java.

0	0
Null	7
Contenido basura	0
'a'	0



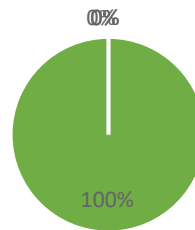
Figura 10.24 Grafico pregunta 9 posTest, participantes que sabían Java.

Pregunta 10:

Tabla 10.25 Tabla pregunta 10 posTest, participantes que sabían Java.

Nombre_del_array [ ][ ] = Tipo_de_variable[dimensión];	0
New Tipo_de_variable[ ] Nombre_del_array = Tipo_de_variable[dimensión];	0
Tipo_de_variable Nombre_del_array [ ] = Tipo_de_variable[dimensión];	0
Tipo_de_variable[ ] Nombre_del_array = new Tipo_de_variable[dimensión];	7

La sintaxis para declarar e inicializar un array  
será:



- Nombre\_del\_array [ ][ ] = Tipo\_de\_variable[dimensión];
- New Tipo\_de\_variable[ ] Nombre\_del\_array = Tipo\_de\_variable[dimensión];
- Tipo\_de\_variable Nombre\_del\_array [ ] = Tipo\_de\_variable[dimensión];
- Tipo\_de\_variable[ ] Nombre\_del\_array = new Tipo\_de\_variable[dimensión];

Figura 10.25 Grafico pregunta 10 posTest, participantes que sabían Java.

Pregunta 11:

Tabla 10.26 Tabla pregunta 11 posTest, participantes que sabían Java.

Excepción	0
Variable de control	0
Metodo	6
Funcion Main()	1

La definición "conjunto de instrucciones definidas dentro de una clase, que realizan una determinada tarea y a las que podemos invocar mediante un nombre " corresponde a :

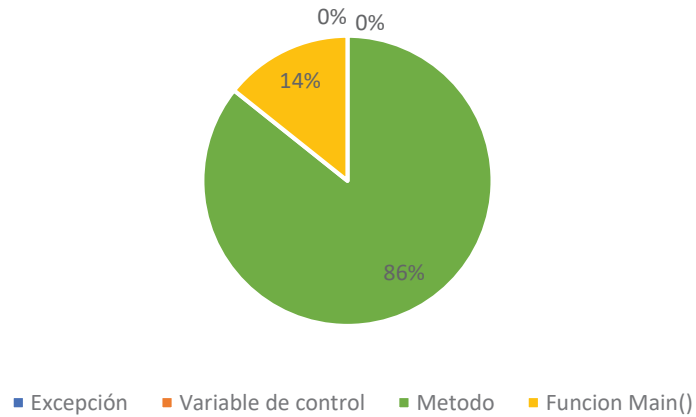


Figura 10.26 Grafico pregunta 11 posTest, participantes que sabían Java.

Pregunta 12:

Tabla 10.27 Tabla pregunta 12 posTest, participantes que sabían Java.

Su retorno es variable segun los parametros de entrada	0
Su retorno es variable y no depende de los parametros de entrada	0
No posee retorno	7
Es una funcion con retorno indefinido	0



Si un método posee el tipo de retorno "void"  
significa que:

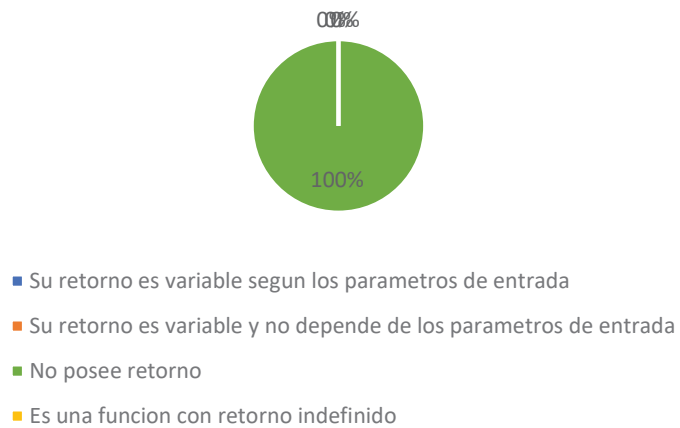


Figura 10.27 Grafico pregunta 12 posTest, participantes que sabían Java.

Pregunta 13:

Tabla 10.28 Tabla pregunta 13 posTest, participantes que sabían Java.

Se puede instanciar un objeto a de la clase MiClase	7
Siempre provoca un error en el compilador	0
Siempre se provoca un error por falta de argumentos	0

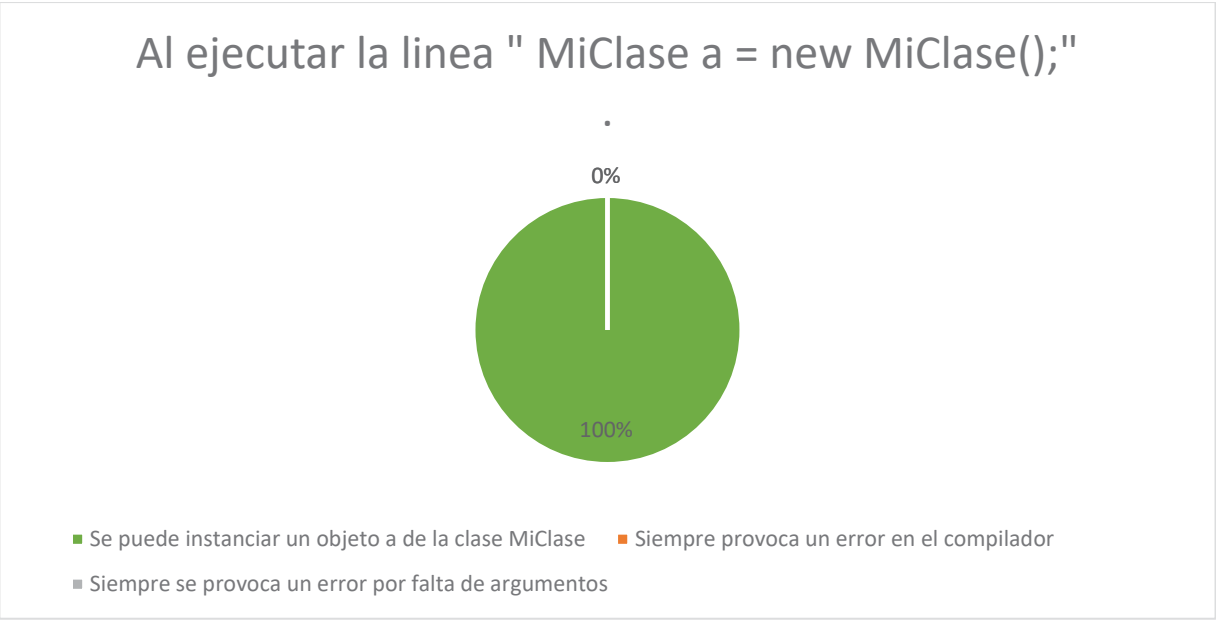


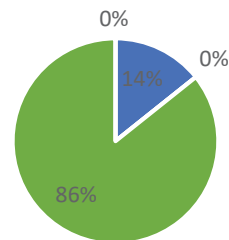
Figura 10.28 Grafico pregunta 13 posTest, participantes que sabían Java.

Pregunta 14:

Tabla 10.29 Tabla pregunta 14 posTest, participantes que sabían Java.

private char cambiaLetra (char letra);	1
public int saludar(int hola);	0
public int cambiaLetra(char letra);	6
public void cambiaLetra(char letra);	0

Cuál sintaxis corresponde a la cabecera de un método que tiene por nombre cambiaLetra, modificador de visibilidad público, retorna un valor tipo entero y recibe por parámetro un carácter.



■ private char cambiaLetra (char letra); ■ public int saludar(int hola);  
■ public int cambiaLetra(char letra); ■ public void cambiaLetra(char letra);

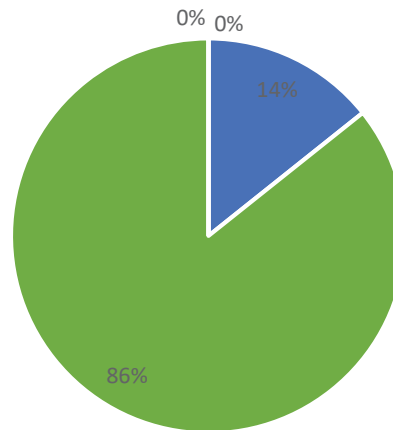
Figura 10.29 Grafico pregunta 14 posTest, participantes que sabían Java.

Pregunta 15:

Tabla 10.30 Tabla pregunta 15 posTest, participantes que sabían Java.

Escribir cualquier tipo de código en Java	1
Reservar una porción determinada de memoria en la que se almacenarán los datos, y las direcciones de memoria de sus métodos	6
Ejecutar una excepción	0
Limpiar una variable	0

"Instanciar" corresponde a:



### 10.2.1 Conclusiones de las Respuestas

Se observa que los resultados han mejorado al comparar ambos Test, sin embargo aun hay respuesta incorrectas, la parte que contiene mas errores es en la declaración del char, pues todos los usuarios supieron declarar un arreglo correctamente lo que les faltó a la mayoría fue dar el largo 5, por lo tanto al no cumplir con las instrucciones se les considera incorrecta, en conclusión los alumnos saben declarar arreglos, pero no siguen las instrucciones.

## 10.3 Comparación PreTest y PosTest por Temáticas

### 10.3.1 Temática: Clases

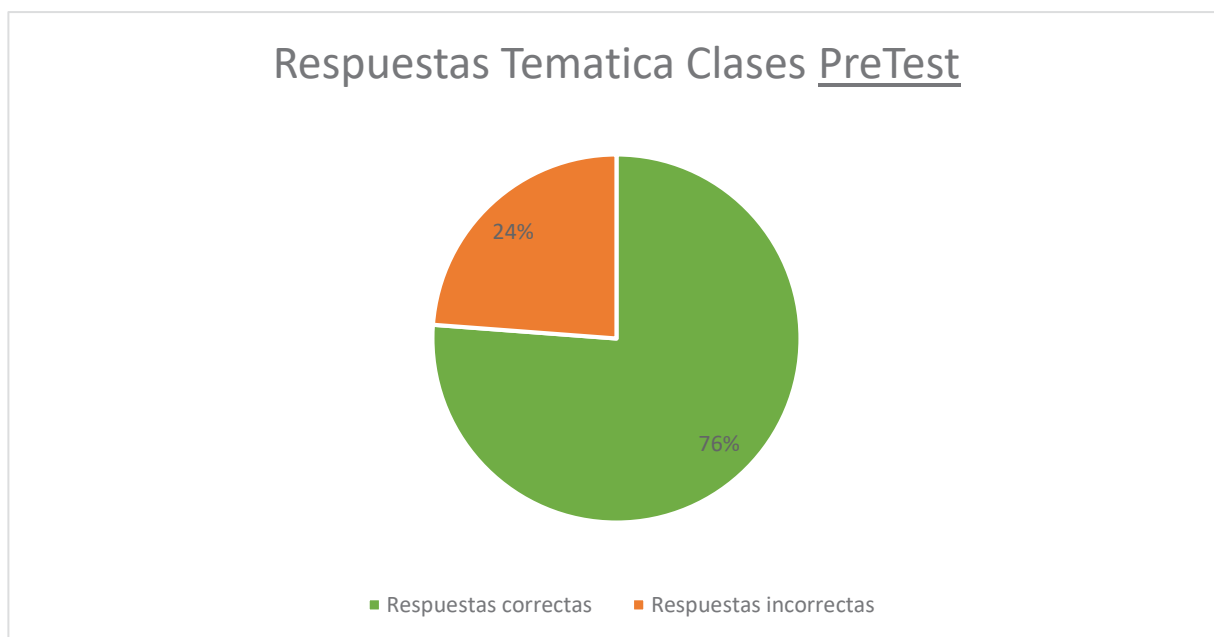


Figura 10.30 Grafico temática clases PreTest, Alumnos que saben Java

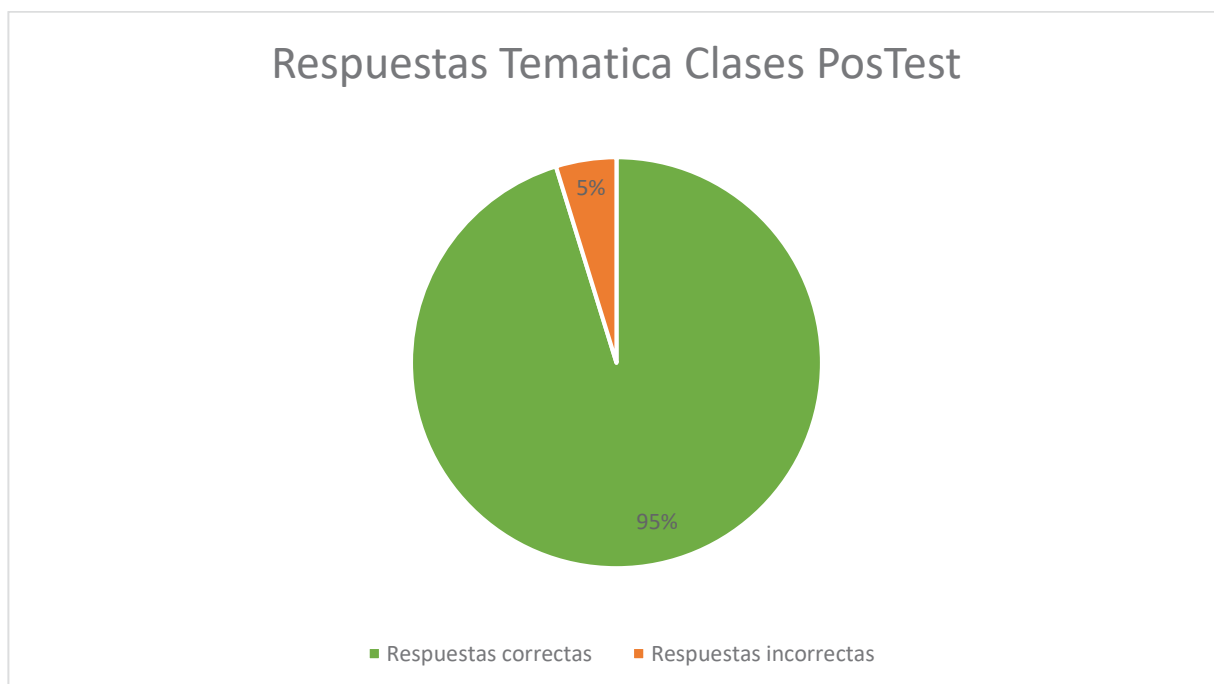


Figura 10.31 Grafico temática clases PosTest, Alumnos que saben Java

Se demuestra mejoras después de probar el software educativo casi el 100% pudo mejorar los conceptos tratados.

### 10.3.2 Temática: Arreglos

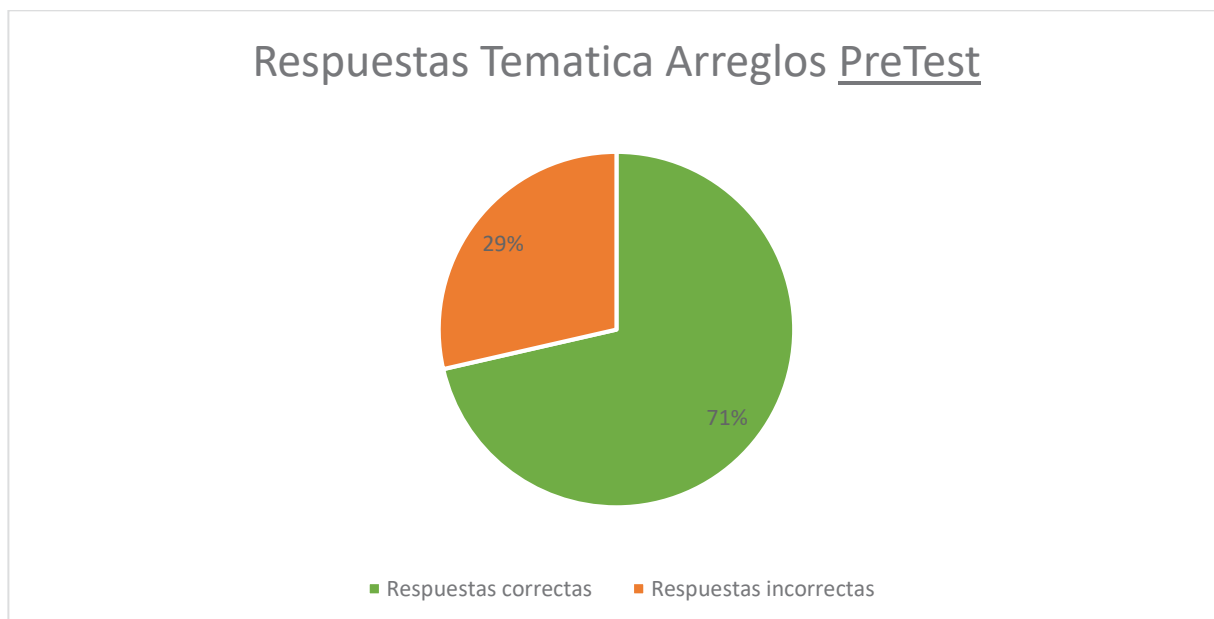


Figura 10.32 Grafico temática arreglos PreTest, Alumnos que saben Java

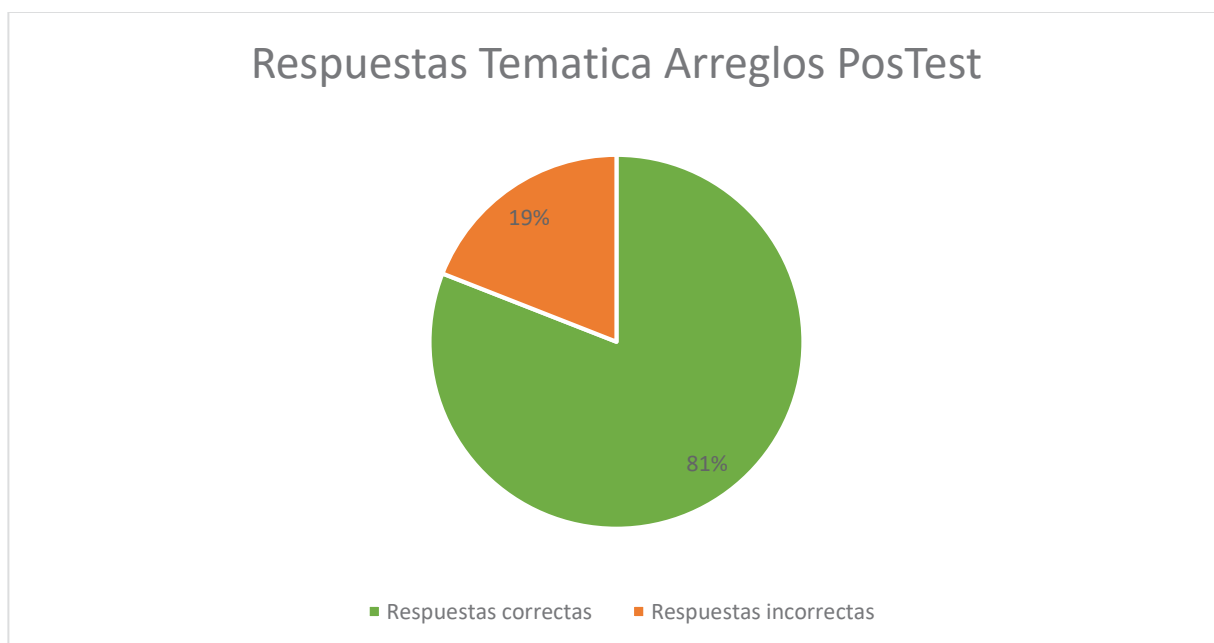


Figura 10.33 Grafico temática arreglos PostTest, Alumnos que saben Java

En esta temática se aprecia mejoran de un 10% por lo que al igual que el anterior test se concluye que hay que mejorar los tutoriales que involucren arreglos.

### 10.3.3 Temática: Atributos

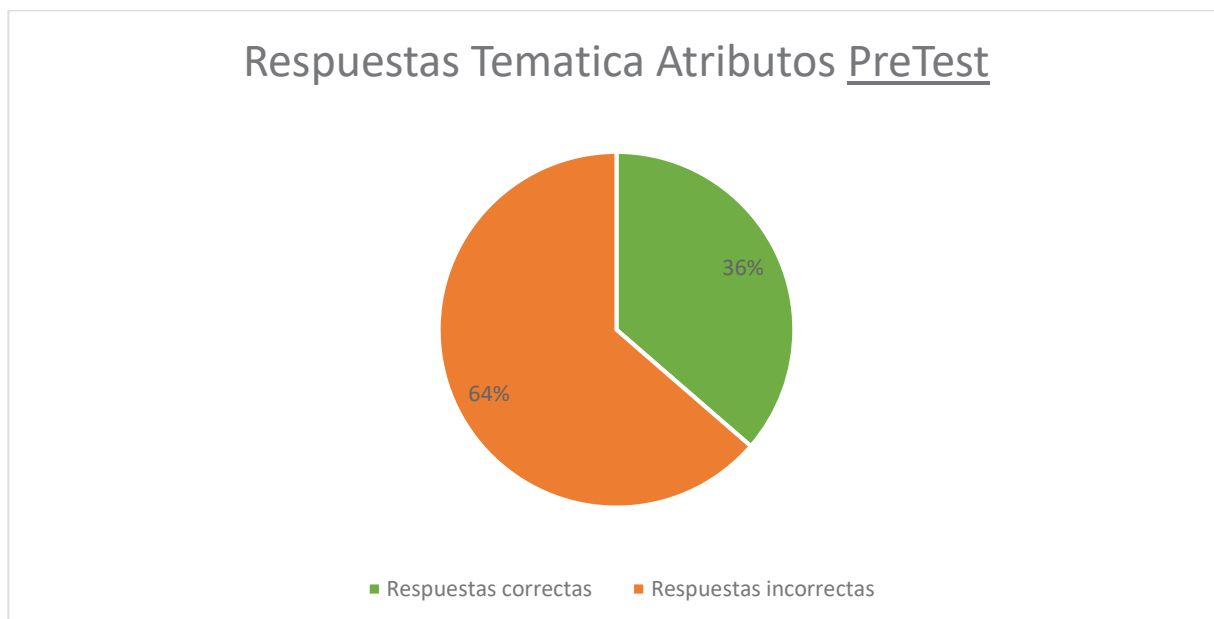


Figura 10.34 Grafico temática atributos PreTest, Alumnos que saben Java

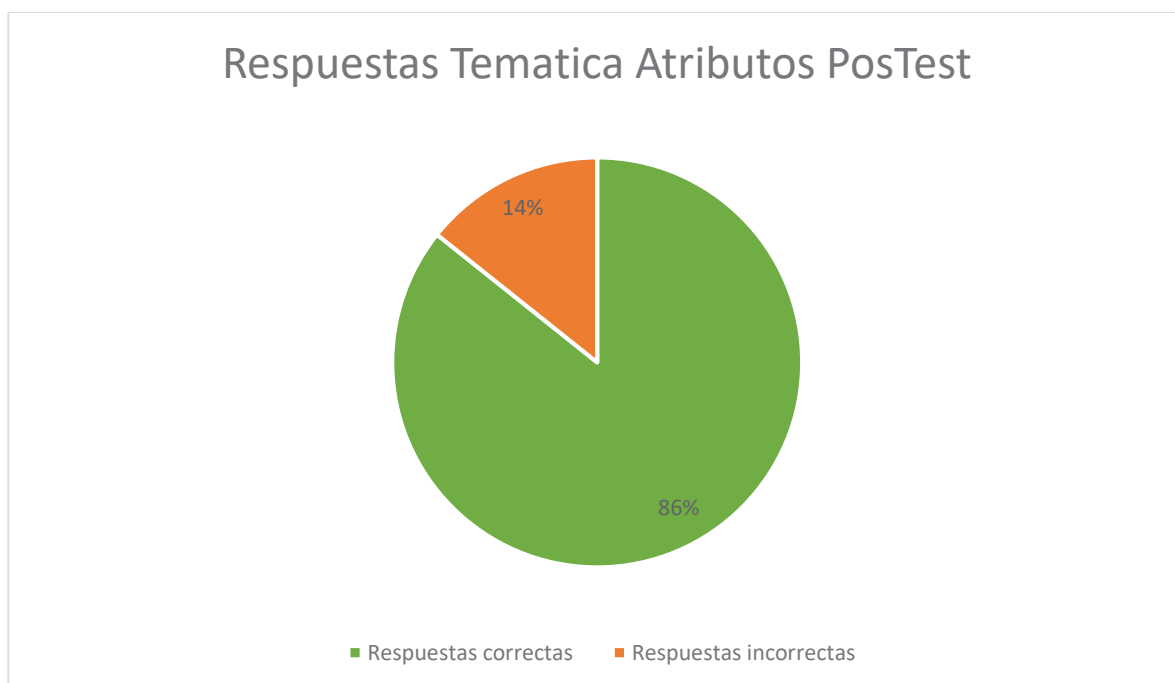


Figura 10.35 Grafico temática atributos PosTest, Alumnos que saben Java

Se puede ver una clara mejora de parte de los estudiantes al probar el software educativo obteniendo un 50% de mejoras en los conceptos evaluados. Por lo tanto, lo que habría que mejorar en los tutoriales es el concepto de modificador de visibilidad que se pudieron apreciar en los Test con el grupo anterior.

### 10.3.4 Temática: Métodos

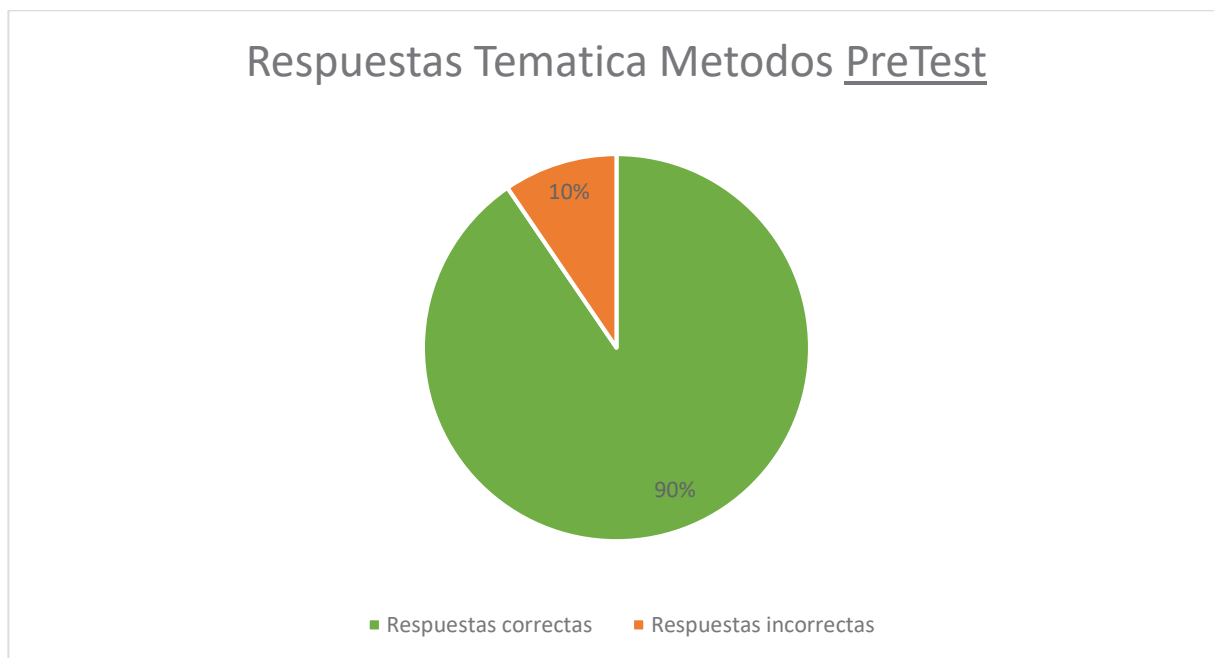


Figura 10.36 Grafico temática métodos PreTest, Alumnos que saben Java

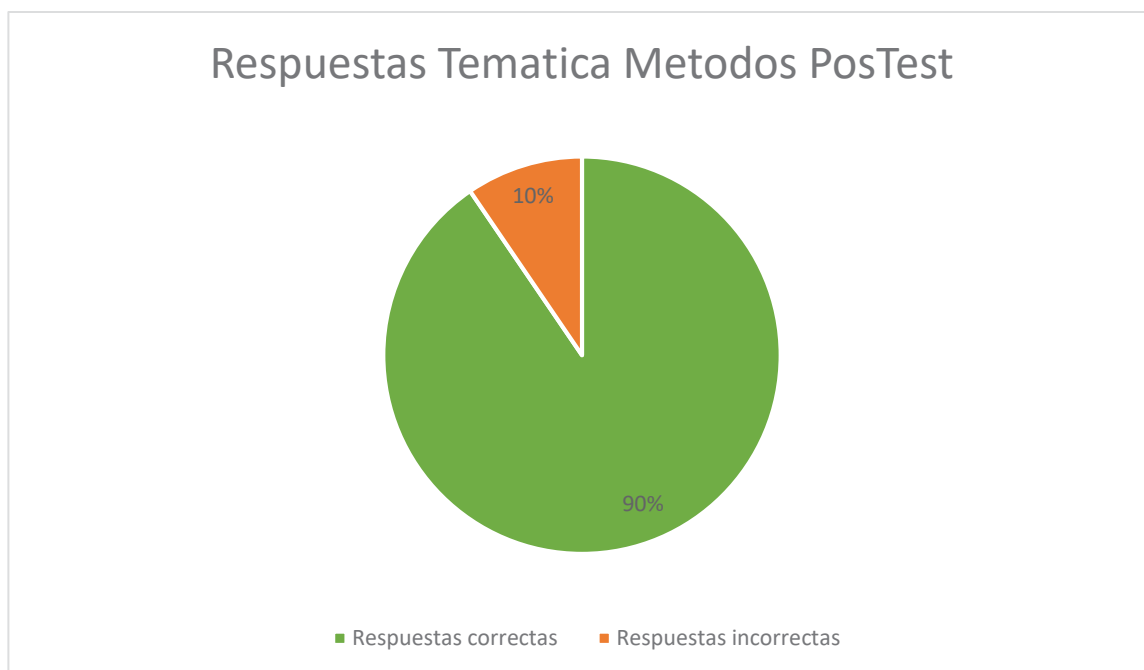


Figura 10.37 Grafico temática métodos PosTest, Alumnos que saben Java

Se observa que los usuarios se mantienen neutro ante este contenido, se concluye que se deben mejorar los tutoriales propuesto para lograr una mejora del 100% para los usuarios que necesiten reforzar.



### 10.3.5 Temática: Objetos



Figura 10.38 Grafico temática objetos PreTest, Alumnos que saben Java

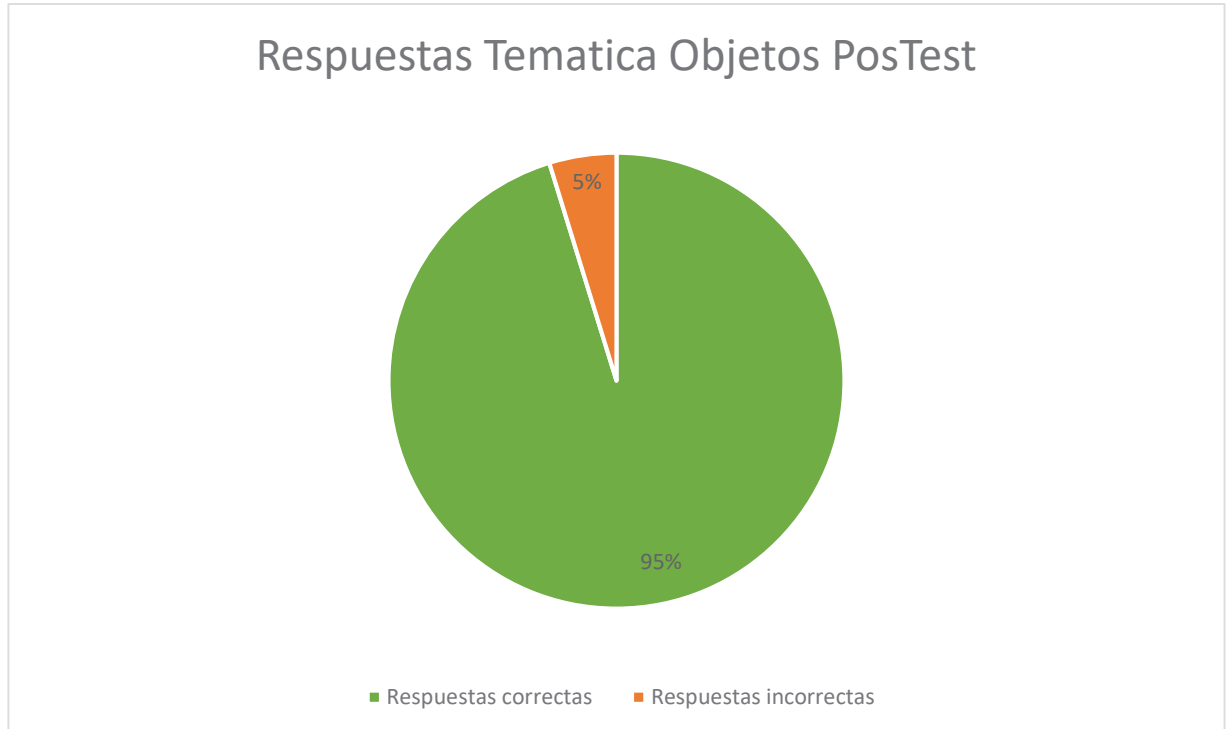


Figura 10.39 Grafico temática objetos PosTest, Alumnos que saben Java

Al igual que el anterior se mantienen intacto las comparaciones de ambos Test.

## 11 Conclusiones de Ambas Pruebas

Al comparar ambas pruebas podemos observar que en ambos casos mejoraron en cuanto a su preparación previa a probar el software educativo, sin embargo, los participantes que ya sabían de ante mano la materia propuesta fueron muy pocos los que realizaron estas pruebas, por lo que, no se puede sacar una conclusión muy efectiva. Por otra parte los usuarios que probaron el software sin tener conocimientos previos de Java mostraron una notable mejora.

Por lo que se concluye que el software educativo esta enseñando y reforzando a los alumnos, esto se ve reflejado en los resultados obtenidos en ambas pruebas, sin embargo se considera hacer mas pruebas para que se vean mejor reflejada esta conclusión.

## 12 Conclusión

Actualmente hay diversos softwares educativos para el aprendizaje de orientación a objetos, en la investigación de este documento se dio a conocer los semejantes al proyecto propuesto, sin embargo estos softwares no cuentan con un sistema de tutoriales capaces de enseñar la *POO*, sino que necesitan manuales externos a la aplicación, que comúnmente se encuentran en la página web del creador. Además, estos no incorporan la gamificación, sino que son utilizados para la creación de juegos. Por lo tanto, el software educativo desarrollado contara con las capacidades de entregar un sistema de tutoriales incorporado al software para que el usuario al no saber o comprender un concepto, pueda guiarse por los tutoriales que contiene el este, a diferencia de los otros softwares, este contara con gamificación por lo que el usuario tendrá que pasar por diferentes niveles, por cada nivel tendrá una misión y podrá acumular puntos a medida que conteste adecuadamente.

Con la implementación de gamificación se espera que el usuario pueda motivarse para seguir adelante en el juego y poder finalizarlo satisfactoriamente. De igual manera la integración de tutoriales ayuda a comprender de mejor manera los contenidos abordados en el juego y permite a los usuarios a apoyarse en esta herramienta de aprendizaje.

Se pretende seguir desarrollando el software educativo, una vez terminado integrar las pruebas a los usuarios, verificar la aceptación de esta herramienta y si sirve como medio de aprendizaje. Si la aceptación es positiva se evaluara agregar más contenido de *OO*.

## 13 Referencias

- [1] J. E. Sánchez-García, M. Urías-Ruiz y B. E. Gutiérrez-Herrera, *ANÁLISIS DE LOS PROBLEMAS DE APRENDIZAJE DE LA PROGRAMACIÓN*, *Ra Ximhai*, vol. 11, nº 4, pp. 289-304, 2015.
- [2] L. G.-M. Juan Vargas-Enríquez, *Análisis de uso de la Gamificación en la Enseñanza de la Informática*, Instituto de Tecnologías y Sistemas de Información, Ciudad Real, España, 2015.
- [3] J. O. Pabon Guerrero, *El software educativo y su importancia en la educación* Disponible vía web en: <https://soloeducacionblog.wordpress.com/el-software-educativo-y-su-importancia-en-la-educacion/>.
- [4] G. M. Q. Francisco A. Martínez Gil, *Introducción a la programación estructurada en C*, Valencia: Universidad de Valencia, 2003.
- [5] Mineduc, *Recursos Educativos*, 2015. [En línea]. [Último acceso: 2017].
- [6] L. K. Group, *scratch*, MIT Media Lab, 18 Mayo 2007. Disponible vía web en: <https://scratch.mit.edu>. [Último acceso: 26 Agosto 2017].
- [7] D. B. Mejía, *Implementación de Scratch para potenciar el aprendizaje significativo a través lógica de programación en los estudiantes de Nivel Básica*, Tecnológico de Monterrey, Aguachica, Colombia, 2014.
- [8] D. K. B. R. y. J. M. Eric Rosenbaum, «beetleblocks,» 2014. Disponible vía web en: <http://beetleblocks.com>. [Último acceso: 2017].
- [9] E. d. Scratch, «Scratchx,» Laboratorio de Medios, 9 Mayo 2015. Disponible vía web en: <http://scratchx.org/>. [Último acceso: 26 Agosto 2017].
- [10] M. Co, «mBlock,» 2013. Disponible vía web en: <http://www.mblock.cc>. [Último acceso: 2017].
- [11] B. H. Jens, «Snap! — Build Your Own Blocks,» 2011. Disponible vía web en: <http://snap.berkeley.edu/>. [Último acceso: 2017].
- [12] U. d. Berkeley, «Aprendo a programar,» 2015. Disponible vía web en: <https://aprendoaprogramar.com/>. [Último acceso: 2017].
- [13] L. Carroll, «Alice,» 1995. Disponible vía web en: <http://www.alice.org>. [Último acceso: 2017].
- [14] D. A. S. Olga Lucía Monroy Vecino, «ENSEÑANDO PROGRAMACIÓN ORIENTADA A OBJETOS».

- [15] E. T. S. d. I. I. d. I. URJC, «rebeca 3d,» 2011. Disponible vía web en: <https://rebeca3d.wordpress.com/>. [Último acceso: 2017].
- [16] U. d. Kent, «Greengoot,» 8 Septiembre 2011. Disponible vía web en: <https://www.greenfoot.org>. [Último acceso: 26 Agosto 2017].
- [17] S. T. García, «Desarrollo de un escenario en el entorno Greenfoot : hundir la flota,» UNIVERSIDAD CARLOS III DE MADRID, Madrid, 2009.
- [18] I. Kereki, «Enseñando y Aprendiendo Programación Orientada a Objetos en los primeros cursos de Programación: la experiencia en la Universidad ORT Uruguay,» Universidad ORT Uruguay, Uruguay.
- [19] C. A. C. C. G. E. T. B. RICARDO DE JESÚS BOTERO TABARES, Lógica y programación orientada a objetos: Un enfoque basado en problemas, Medellín: Tecnológico de Antioquia, 2009.
- [20] H. Fingerhann, «Aprendizaje por recepción,» 31 Agosto 2010. Disponible vía web en: <http://educacion.laguia2000.com/aprendizaje/aprendizaje-por-recepcion>. [Último acceso: 26 Agosto 2017].
- [21] U. i. d. valencia, «viu,» 9 Marzo 2015. Disponible vía web en: <http://www.viu.es/el-aprendizaje-por-descubrimiento-de-bruner/>. [Último acceso: 26 Agosto 2017].
- [22] M. T. Ruth Benitez, «Teoría del aprendizaje por descubrimiento de bruner,» 18 Junio 2016. Disponible vía web en: <https://es.slideshare.net/Ruth061986/aprendizaje-por-descubrimiento-de-bruner>. [Último acceso: 26 Agosto 2017].
- [23] D. d. I. e. I. Educativa, Escritor, *Aprendizaje Colaborativo Técnicas Didácticas*. [Performance].
- [24] *Aprendizaje Colaborativo*. [Performance]. Dirección de Investigación e Innovación Educativa .
- [25] C. García, «VENTAJAS Y DESVENTAJAS DEL APRENDIZAJE COLABORATIVO,» 10 Agosto 2012. Disponible vía web en: <https://saiaequipo3.wordpress.com/2012/08/10/ventajas-y-desventajas-del-aprendizaje-colaborativo/>. [Último acceso: 26 Agosto 2017].
- [26] S. d. I. E. d. I. U. P. d. Madrid, «Aprendizaje Basado en Problemas,» Madrid, 2008.
- [27] aulaplaneta, «Ventajas del aprendizaje basado en la resolución de problemas,» 25 Agosto 2015. [En línea]. [Último acceso: 28 Agosto 2017].

- [28] C. Sola, Aprendizaje basado en problemas: de la teoría a la práctica, Mexico: Editorial Trillas, 2005.
- [29] M. Sánchez, «Qué son y para qué sirven,» Disponible vía web en: <http://creatic.innova.unia.es/edicionpublicacion/videotutoriales>. [Último acceso: 27 Agosto 2017].
- [30] A. M. Arostegui, «Aprender para enseñar,» 21 Junio 2016. Disponible vía web en: <https://blogs.deusto.es/aprender-ensenar/crea-tus-videos-docentes/>. [Último acceso: 27 Agosto 2017].
- [31] S. Garcia, «SoreyGarcia,» 26 Julio 2008. Disponible vía web en: <http://blog.soreygarcia.me/2008/07/enseando-orientacin-objetos.html>. [Último acceso: 25 Agosto 2017].
- [32] J. D. Zapata, «"El Cuento" y su papel en la enseñanza de la orientación por objetos.,» Disponible vía web en: <http://www.c5.cl/ieinvestiga/actas/ribie98/146.html>. [Último acceso: 25 Agosto 2017].
- [33] J. P. P. y. A. Gardey, «Definición de software educativo,» 2016. [En línea]. [Último acceso: 2017].
- [34] V. Gaitán, «Gamificación: el aprendizaje divertido,» 15 Octubre 2013. Disponible vía web en: <http://www.educativa.com/blog-articulos/gamificacion-el-aprendizaje-divertido/>. [Último acceso: 31 Agosto 2017].
- [35] S. Moll, «Gamificación: 7 claves para entender qué es y cómo funciona,» 05 Junio 2014. Disponible vía web en: <http://justificaturespuesta.com/gamificacion-7-claves-para-entender-que-es-y-como-funciona/>. [Último acceso: 31 Agosto 2017].
- [36] C. Luttecke, «¿Sabes que es UNITY? Descúbrelo aquí,» 04 Abril 2014. Disponible vía web en: <https://deideaaapp.org/sabes-que-es-unity-descubrelo-aqui/>. [Último acceso: 27 Agosto 2017].
- [37] Á. AROCA, «Unity 3D, desarrollo de videojuegos para iOS y Android, gratis hasta el 8 de Abril,» 15 Marzo 2012. Disponible vía web en: <https://www.genbetadev.com/herramientas/unity-3d-desarrollo-de-videojuegos-para-ios-y-android-gratis-hasta-el-8-de-abril>. [Último acceso: 27 Agosto 2017].
- [38] U. Technologies, «Unity,» 31 Marzo 2005. Disponible vía web en: <https://unity3d.com>. [Último acceso: 27 Agosto 2017].
- [39] B. Foundation, «Blender,» 1995. Disponible vía web en: <https://www.blender.org/>. [Último acceso: 2017].

- [40] C. S. Colombia, «¿Qué es Blender 3D?,» 2015. Disponible vía web en: <http://www.cgstudioscolombia.com/blender2/index.php/que-es-blender>. [Último acceso: 2017].
- [41] A. Systems, «Adobe,» 1 Enero 1988. Disponible vía web en: <http://www.adobe.com/products/photoshop.html>. [Último acceso: 27 Agosto 2017].
- [42] J. G. y. S. Microsystems, «Oracle,» 1995. Disponible vía web en: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>. [Último acceso: 2017].
- [43] Networkredux, «LLMMS,» 07 Marzo 2015. Disponible vía web en: <https://lmms.io/>. [Último acceso: 27 Agosto 2017].
- [44] TIOBE, «TIOBE,» Agosto 2017. Disponible vía web en: <https://www.tiobe.com/tiobe-index/>. [Último acceso: Agosto 2017].
- [45] V. G. Salgado, Artist, *El secreto del científico*. [Art]. Técnico de bachillerato en artes, 2017.

## **14 Anexo**



14.1A: Carta Gantt

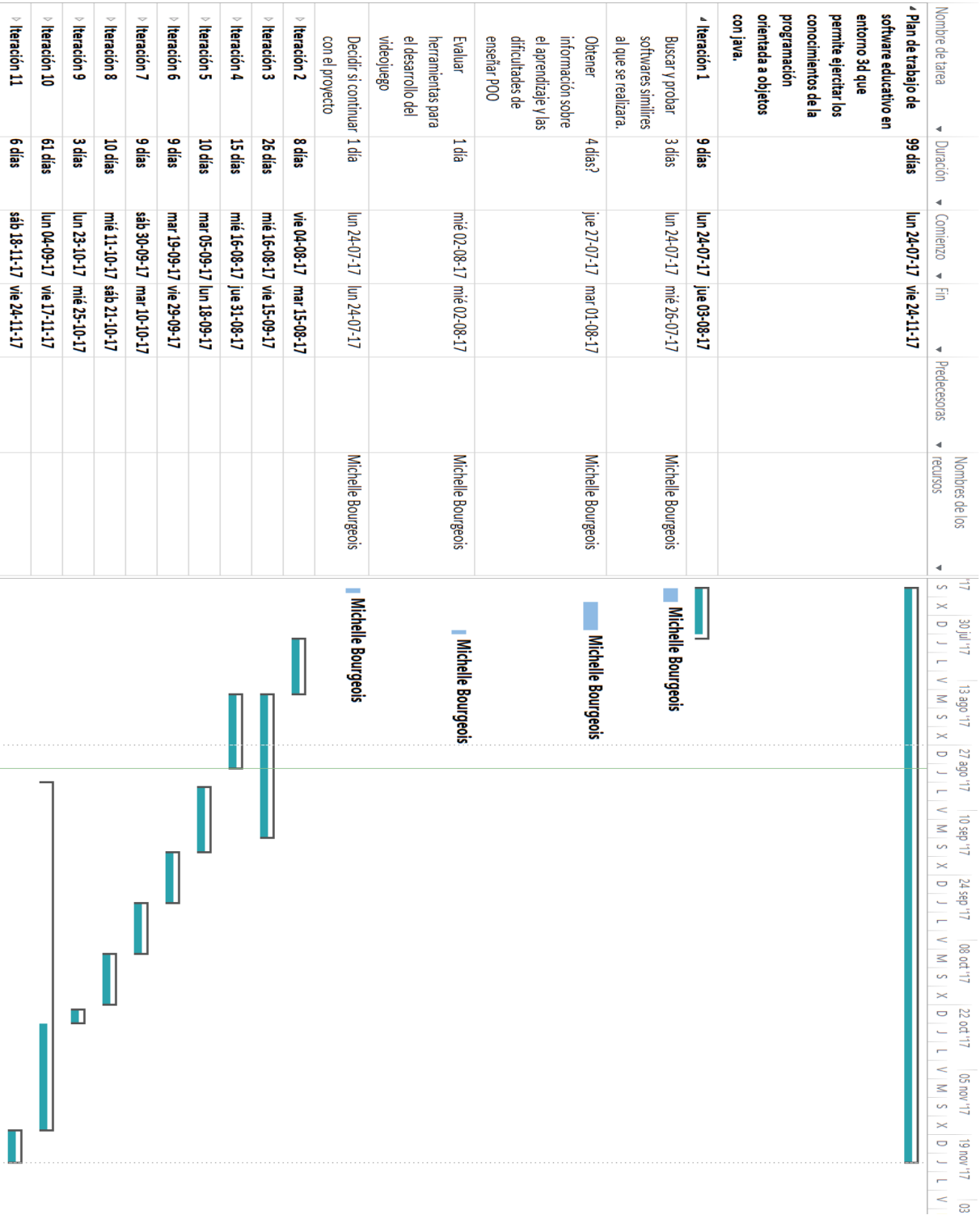


Figura A.1 Carta Gantt Iteración 1

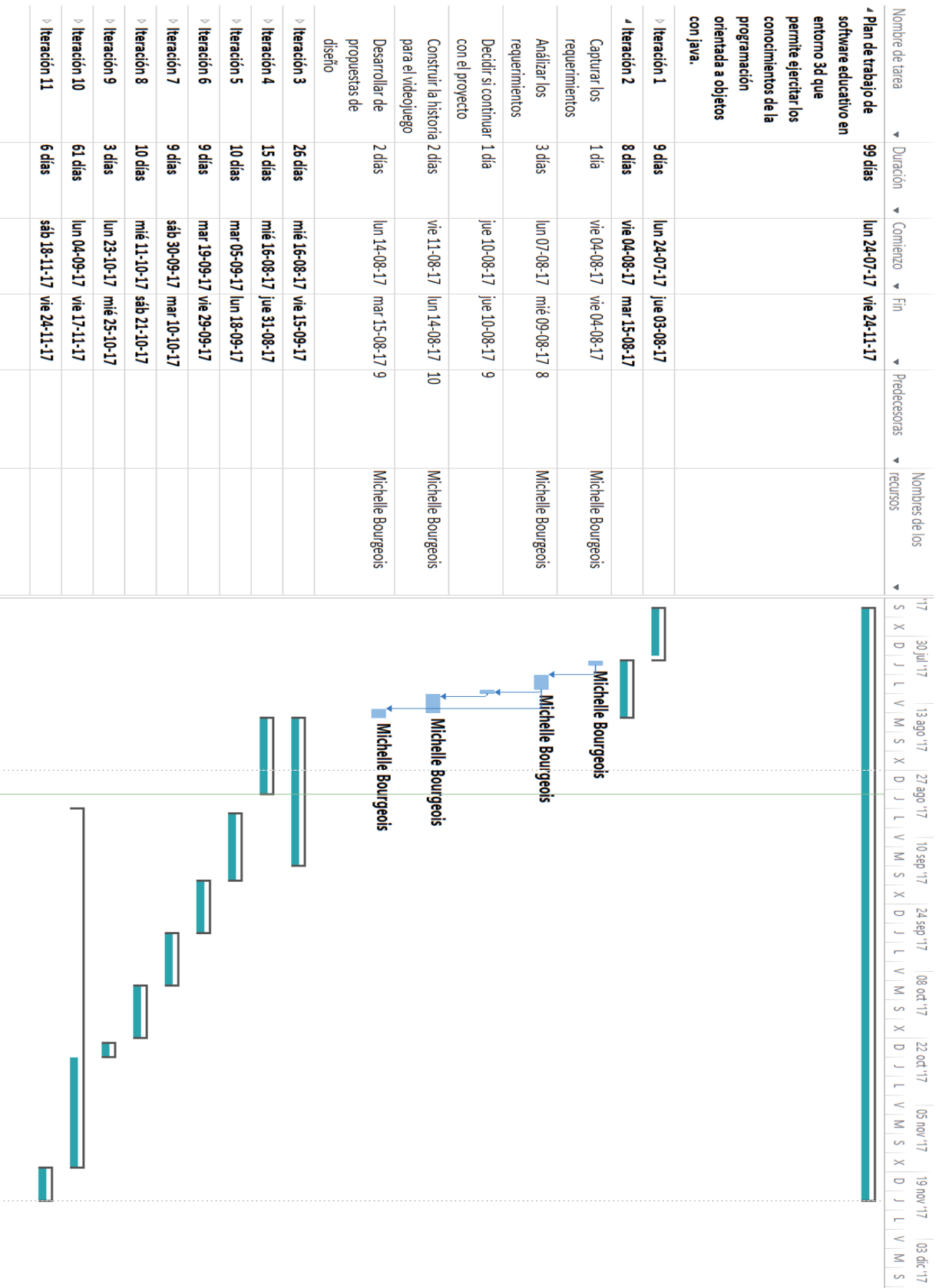


Figura A.2 Carta Gantt Iteración 2

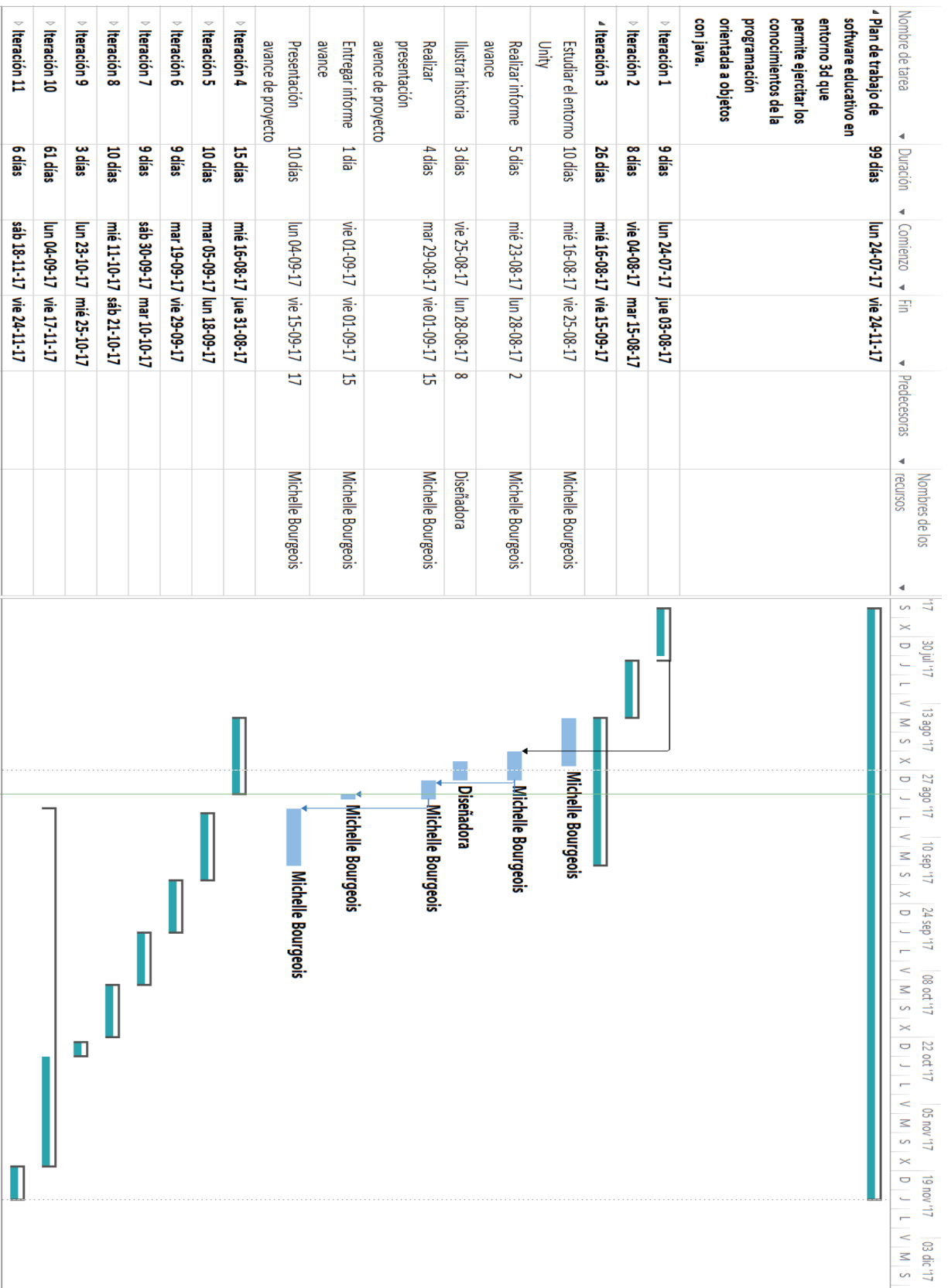


Figura A.3 Carta Gantt Iteración 3

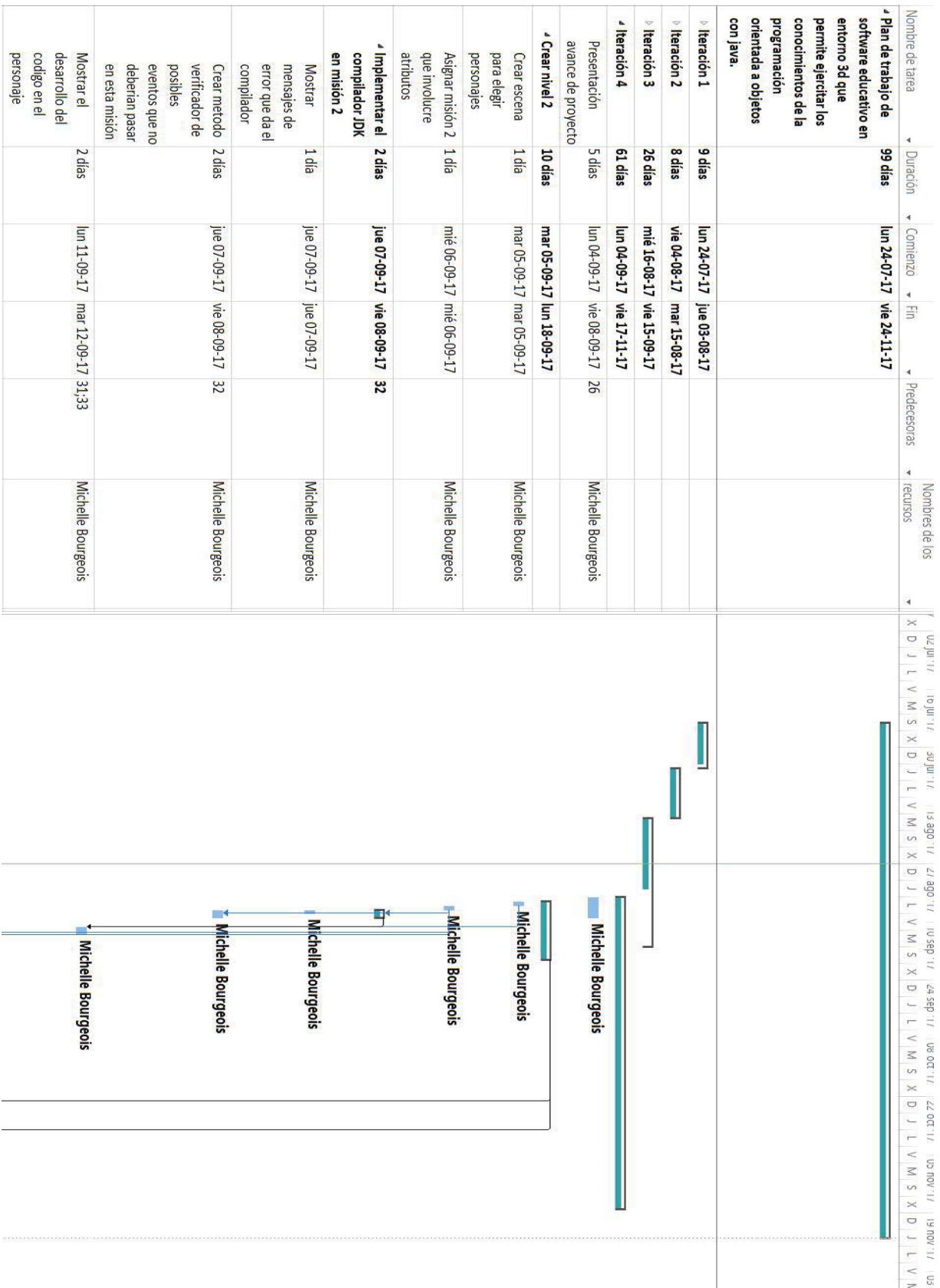


Figura A.4 Carta Gantt Iteración 4

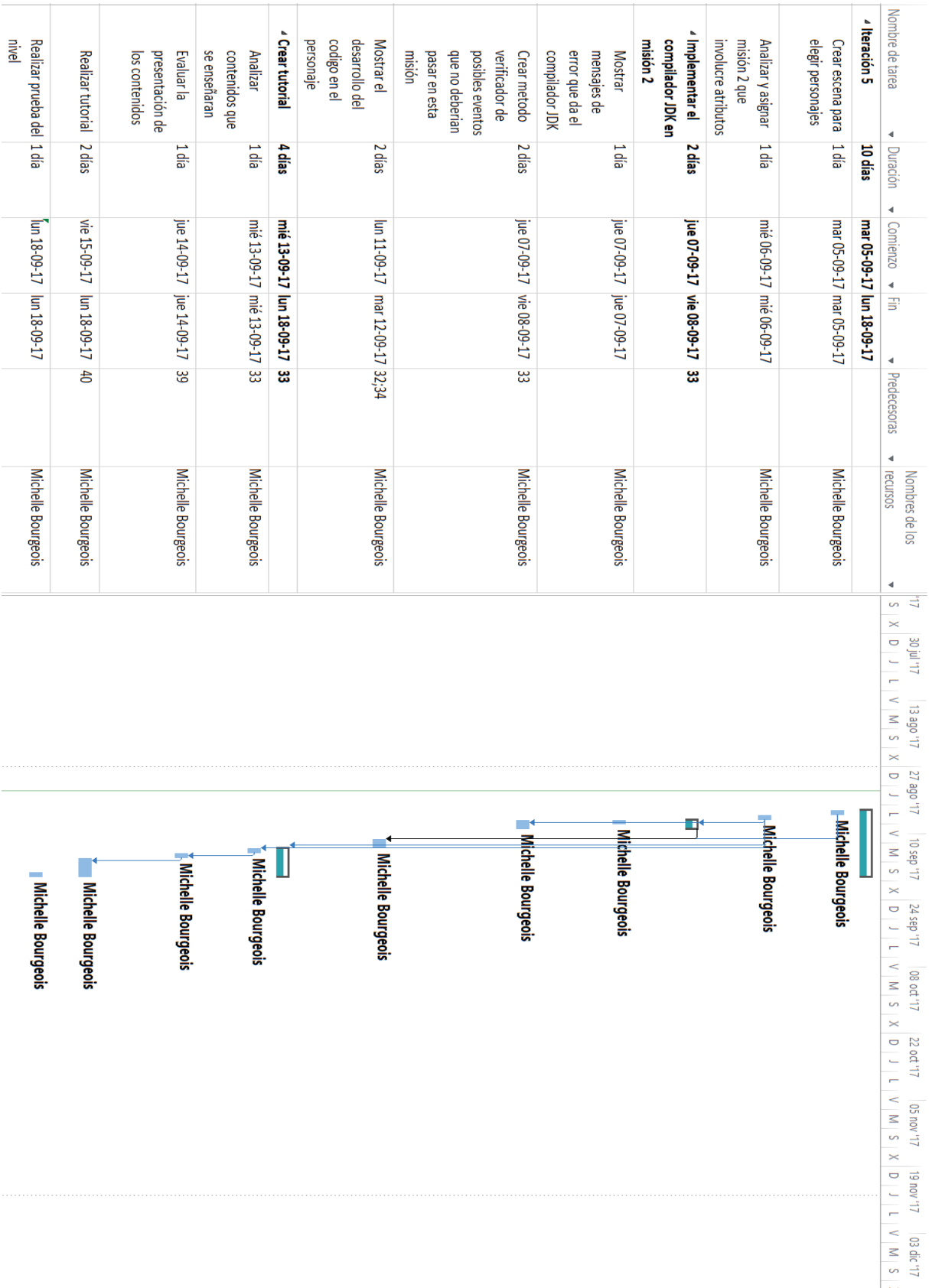


Figura A.5 Carta Gantt Iteración 5

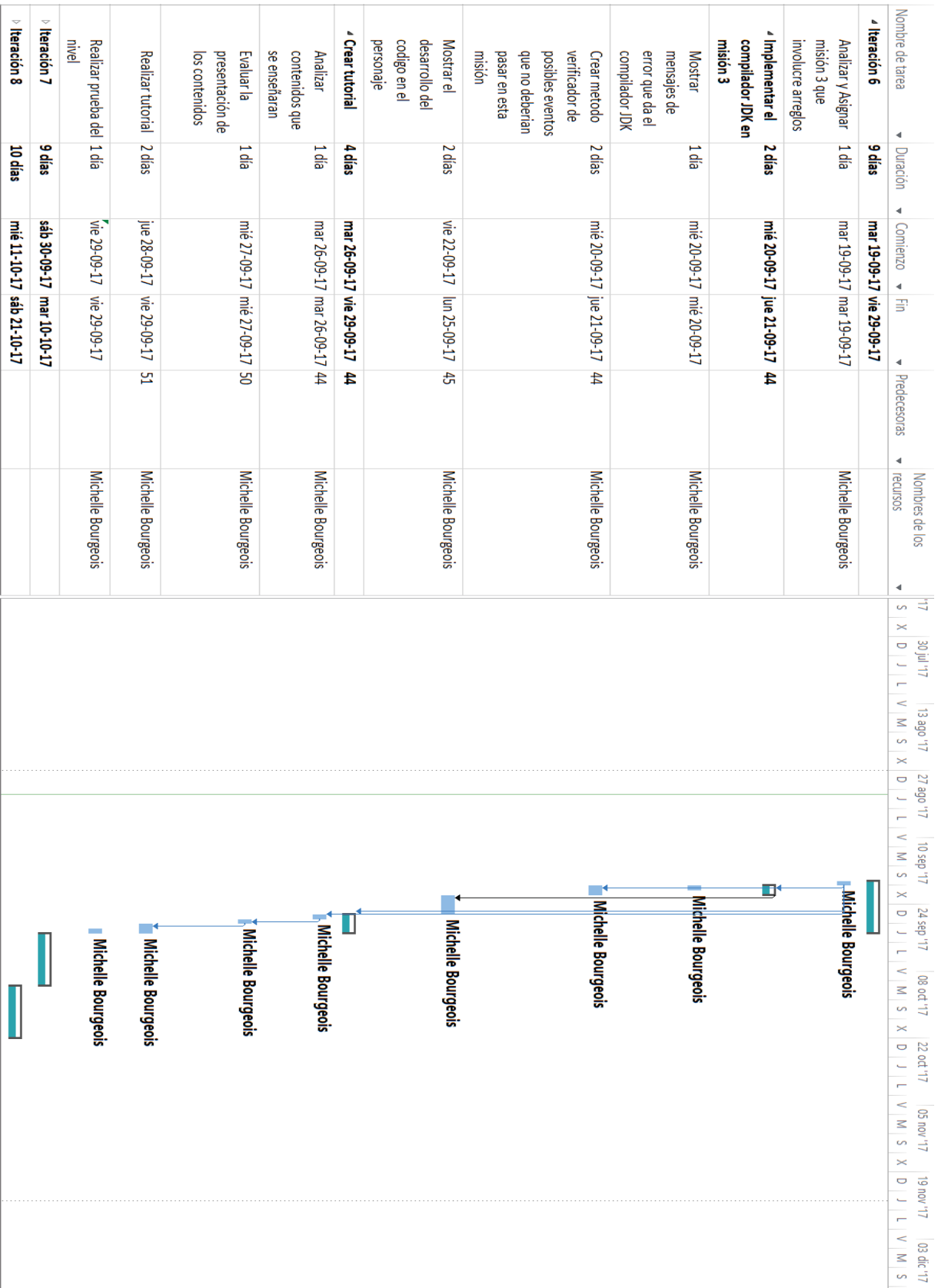


Figura A.6 Carta Gantt Iteración 6

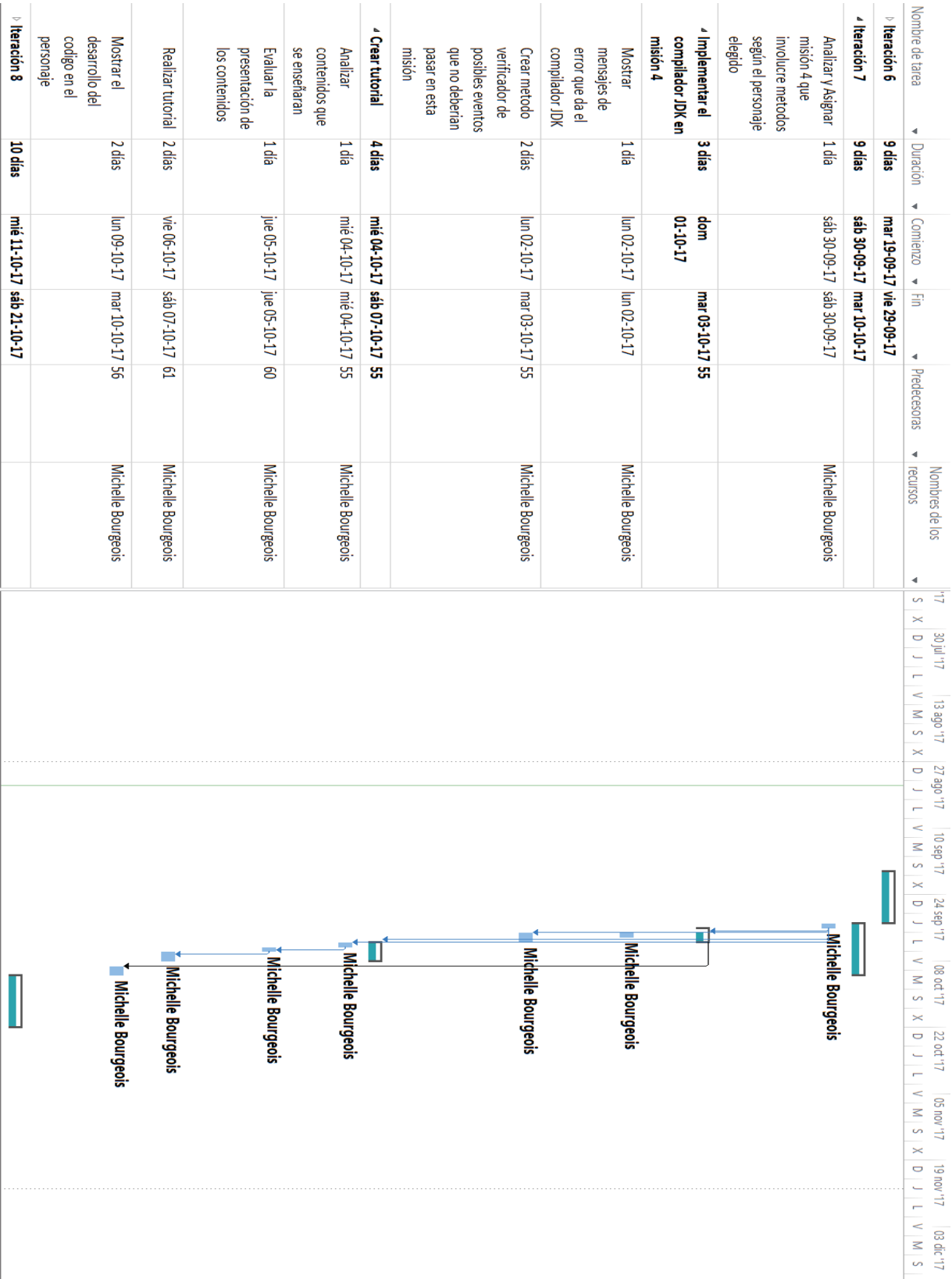


Figura A14.7 Carta Gantt Iteración 7

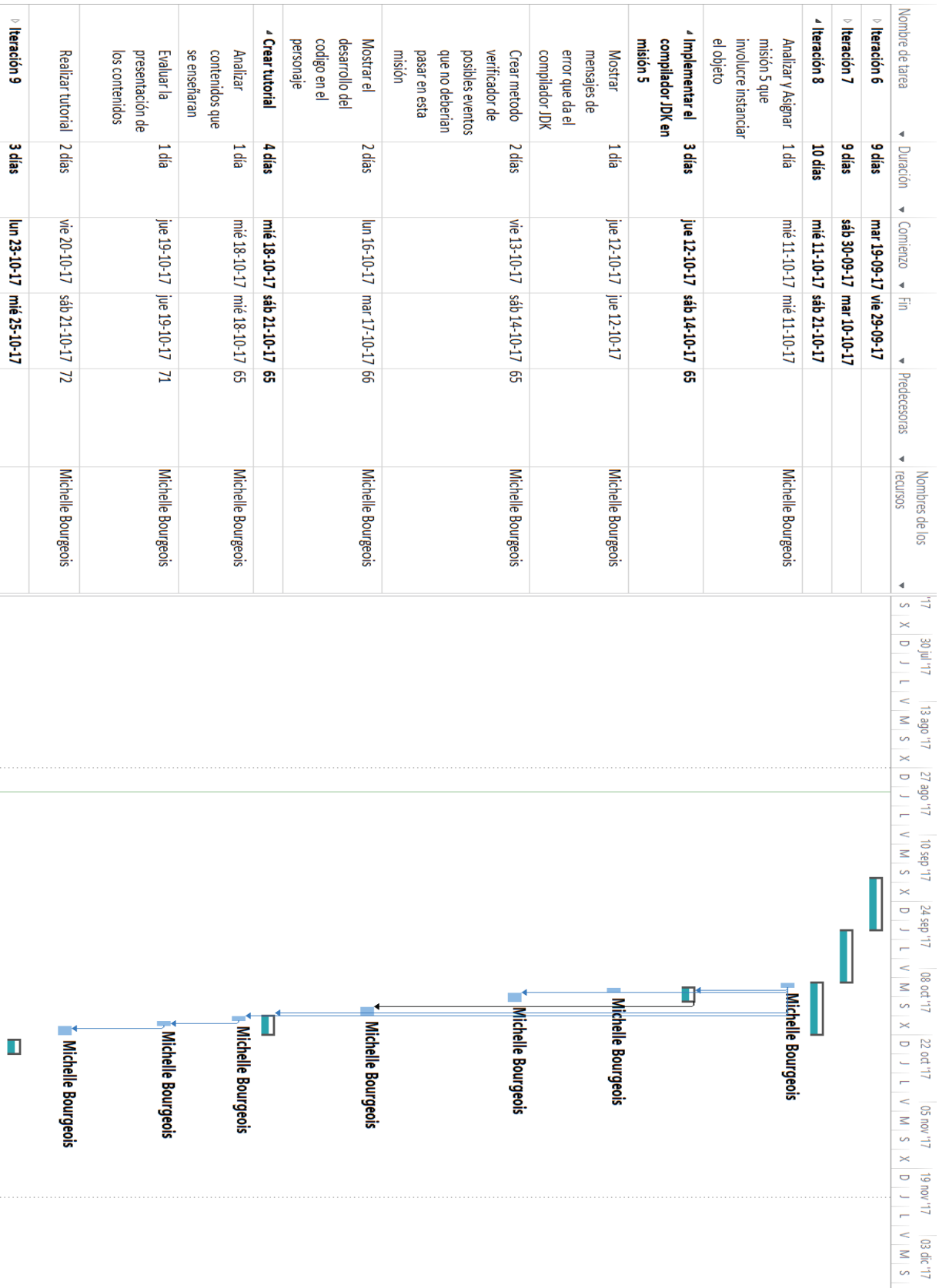


Figura A.8 Carta Gantt Iteración 8



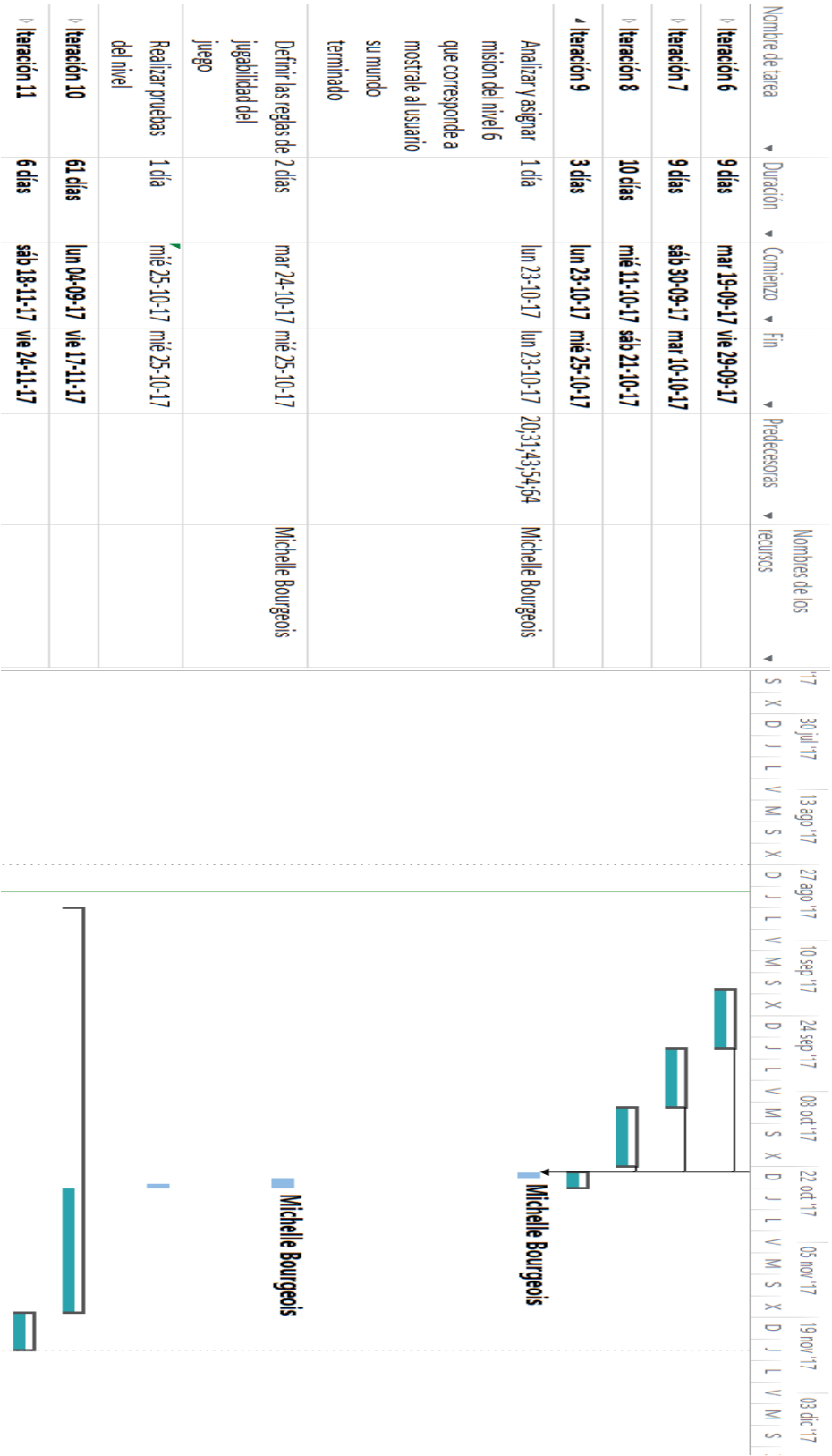


Figura A.9 Carta Gantt Iteración 9

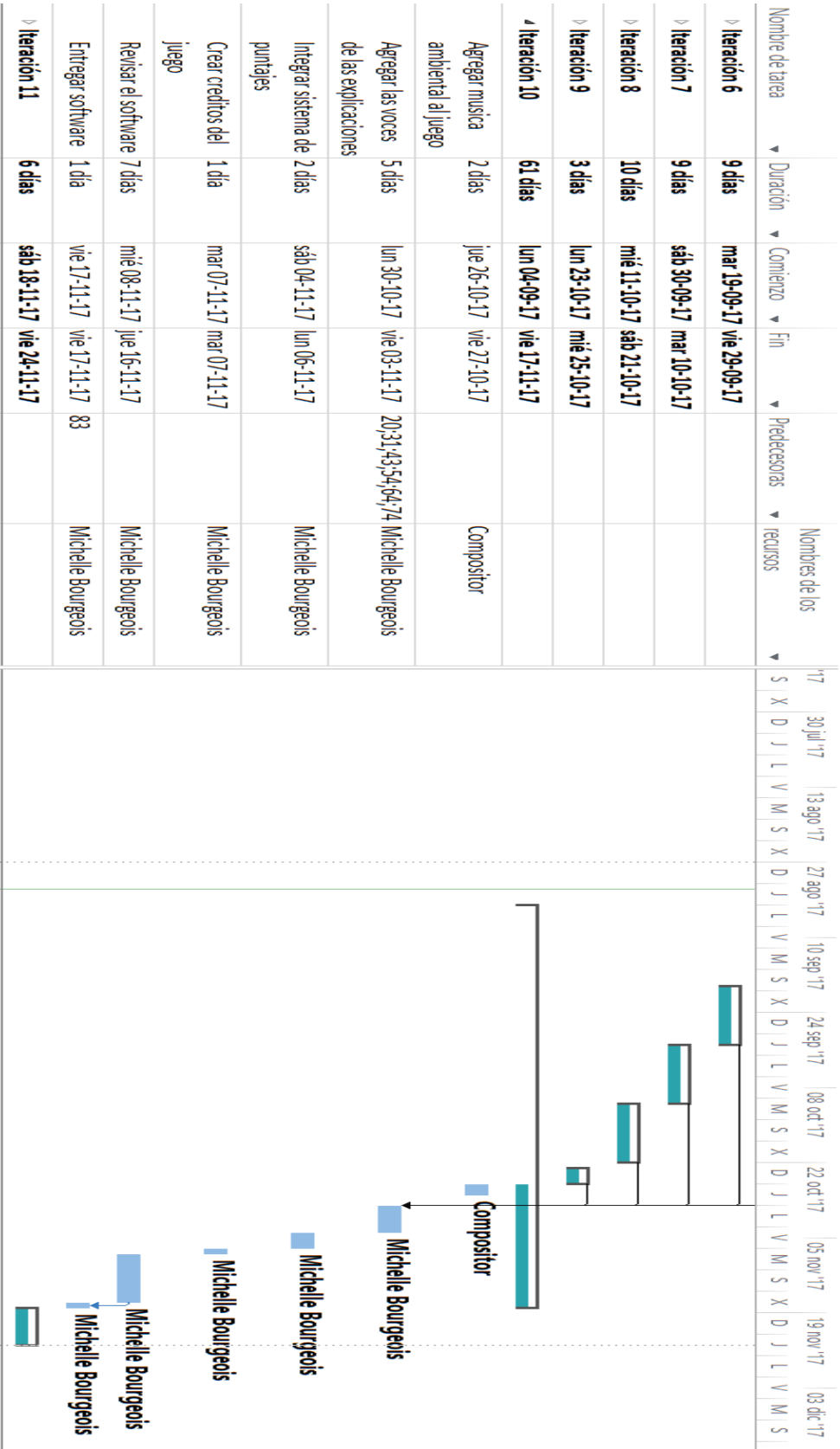


Figura A.10 Carta Gantt Iteración 10

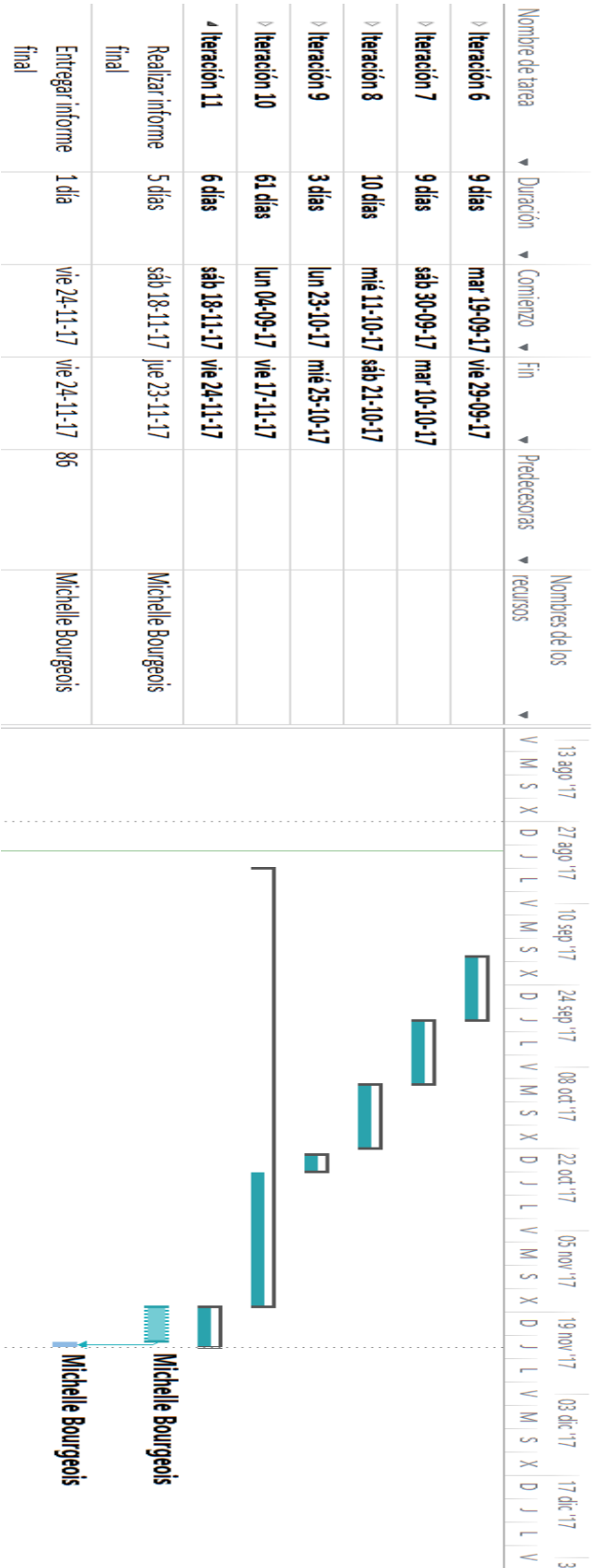


Figura A.11 Carta Gantt Iteración 11