

# Nuevas tendencias de programación visual para la definición de la lógica de las aplicaciones (ID2016/053)

---



## Autores

Juan Francisco De Paz Santana  
Juan Manuel Corchado Rodríguez  
Ángel Luis Sánchez Lázaro  
Vivian Félix López Batista  
Gabriel Villarrubia González  
Alberto López Barriuso  
Javier Bajo Pérez  
Yanira Navarro Marrero



## Contenido

1	<i>Datos</i> .....	1
2	<i>Introducción</i> .....	2
3	<i>Objetivos del proyecto</i> .....	4
4	<i>Ejecución del proyecto</i> .....	5
	<b>4.1 Elaboración de material docente</b> .....	<b>5</b>
	<b>4.2 Valoración de los resultados</b> .....	<b>22</b>
	<b>4.3 Resultados y conclusiones</b> .....	<b>25</b>





## 1 Datos

Título del proyecto: Nuevas tendencias de programación visual para la definición de la lógica de las aplicaciones

Referencia: ID2016/053

Curso: 2016-2017

Participantes:

- Juan Francisco De Paz Santana
- Juan Manuel Corchado Rodríguez
- Ángel Luis Sánchez Lázaro
- Vivian Félix López Batista
- Gabriel Villarrubia González
- Alberto López Barriuso
- Javier Bajo Pérez
- Yanira Navarro Marrero

Adicionalmente a los participantes inicialmente incorporados en la propuesta también participó **Yanira Navarro Marrero** ayudando en la edición de material docente tal y como se muestran en el apartado 4.1. Por este motivo se ha incorporado al listado anterior de participantes.



## 2 Introducción

El desarrollo de herramientas visuales para la gestión de contenidos ha sido de alguna manera el precursor de las herramientas de programación gráfica o visual. En el origen de la web sólo las personas con ciertos conocimientos de programación y de administración de sistemas podían publicar contenidos en la web por lo que los generadores de contenidos en la web eran un número muy reducido si se comparaba con los usuarios que consumían esos contenidos.

De la mano de la web 2.0 surgió la popularización de Internet ya que aparecieron numerosas herramientas que facilitaban a los Internautas el ser productores a la vez que consumidores de información. Mediante el uso de blogs, redes sociales o gestores de contenidos, todo el mundo era capaz de poder publicar contenido en Internet sin necesidad de tener conocimientos avanzados en informática ni los medios tecnológicos necesarios. Esto más que una amenaza para el trabajo de los Ingenieros Informáticos fue una gran oportunidad porque supuso la creación de este tipo de herramientas que facilitaban la democratización de Internet. La programación gráfica o visual lleva años intentando aumentar su presencia en las herramientas, pero su uso se limita al desarrollo de las ciertas partes de la interfaz gráfica de usuario sin que sea posible introducir la funcionalidad de la aplicación. El uso de estos sistemas facilita el desarrollo de las aplicaciones ya que reduce el tiempo de diseño de las interfaces, no obstante, son herramientas muy sencillas que ayudan a escribir parte de código o plantillas de que posteriormente son interpretadas y representadas. La programación gráfica o visual no debe quedarse sólo en la parte de diseño de las interfaces y sería necesario dar un paso más para el desarrollo también de la lógica de negocio de las aplicaciones de modo que sea posible crear un desarrollo integral mediante este tipo de herramientas. Así empresas como el banco Santander han desarrollado sus propias herramientas de programación gráfica adaptadas a sus necesidades como es el caso de Alhambra. En este proyecto de innovación se plantea la necesidad de integrar en el proceso formativo este tipo de herramientas de desarrollo con las técnicas tradicionales para que los alumnos tengan una visión general de las alternativas existentes y aprendan a tomar decisiones sobre la conveniencia de cada una de ellas en función de los proyectos a desarrollar.

El uso de la programación gráfica o visual se aplica desde hace años en componentes muy concretos del desarrollo por ejemplo para las interfaces gráficas. Para el desarrollo de interfaces es común el uso herramientas que faciliten el desarrollo. Para los componentes asociados a la lógica de negocio es más complicado encontrar herramientas que faciliten la programación gráfica o visual por lo que la introducción de esta alternativa supondría una ampliación de las competencias de los alumnos ya que les facilitará el desarrollo de aplicaciones de manera más eficiente y multiplataforma. Por tanto, la programación gráfica o visual permitirá a los alumnos adquirir nuevas competencias en el desarrollo de software.

La impartición de estas nuevas técnicas de programación permitirá a los alumnos introducirse en las nuevas tendencias de programación del mercado. Además, el manejo de estas herramientas es el primer paso para que los alumnos



no sean meros usuarios de estos sistemas, sino que sean los desarrolladores de este tipo de herramientas que facilitan el desarrollo de software de manera más eficiente. Este es el paso natural como así ocurrió con la Web 2.0 y los alumnos deben de estar preparados para poder dar el paso, primero como usuarios de estas herramientas y luego ya como desarrolladores.

La visión a diferentes niveles en las asignaturas permitirá a los alumnos tener una comprensión de los conocimientos en diferentes niveles de abstracción que permitan a los alumnos el uso de herramientas de bajo y alto nivel para el desarrollo de software. De este modo los alumnos desarrollarán una visión crítica sobre las diferentes técnicas y tendrán la capacidad de determinar el tipo de herramientas más adecuadas para los proyectos que deban desarrollar en cada momento.



### 3 Objetivos del proyecto

Desarrollar el material docente adecuado para el desarrollo de las competencias de los alumnos en este tipo de programación que está alejado de las técnicas tradicionales de desarrollo de software. El desarrollo de software tradicional requiere de la implementación del código fuente lo que implica un conocimiento a bajo nivel de los diferentes componentes software y requiere de la definición de arquitecturas adecuadas que permitan abordar el desarrollo del proyecto. Las técnicas de programación gráfica o visual permiten integrar la funcionalidad en entornos estandarizados reduciendo el tipo de desarrollo de arquitecturas y de integración de componentes.

El manejo de estos entornos de programación gráfica o visual formará también a los alumnos para ser desarrolladores de este tipo de herramientas y no meros usuarios. Esta es la diferencia entre un Graduado en Ingeniería Informática y alguien con conocimientos en Informática, por este motivo no se pueden descuidar las técnicas de programación tradicionales. Para el desarrollo de este tipo de software avanzado que permite realizar esta programación gráfica es necesario tener amplios conocimientos en diseño y desarrollo de software.

Impartir diferentes aproximaciones de programación gráfica existente en entornos de desarrollo modernos y que se usan actualmente en la creación de la lógica de negocio de las aplicaciones. En la actualidad existen herramientas profesionales que incorporan estas técnicas de programación y que permiten el desarrollo de software multiplataforma lo que facilita el mantenimiento del software.

Impartir docencia aplicando tres niveles diferentes de programación (programación a bajo nivel, uso de lenguajes avanzados y uso de herramientas de programación gráfica o visual) de modo que los alumnos tengan una visión general de las técnicas existentes y las posibilidades que presenta cada una de ellas. De este modo el alumno conseguirá realizar valoraciones de las diferentes herramientas existentes dotando a los alumnos de una visión crítica que le permita valorar las diferentes alternativas.

Comparar los resultados obtenidos por los estudiantes en los diferentes niveles de programación. La comparación tendrá en cuenta aspectos como: esfuerzo asociado al número de horas dedicadas en formación y tiempo dedicado para el desarrollo de software teniendo especial cuidado en la curva de aprendizaje de las herramientas, calidad del resultado final de los desarrollos realizados en los diferentes proyectos, relación entre esfuerzo y calidad de los resultados finales obtenidos. Para llevar a cabo esta evaluación se elaborarán una serie de encuestas que se pasarán a los alumnos para que las rellenen.





## 4 Ejecución del proyecto

Para la ejecución del proyecto se ha tenido en cuenta los tres niveles que se consideran relevantes para la ejecución del proyecto. En base a la memoria presentada se consideran tres niveles de programación diferentes:

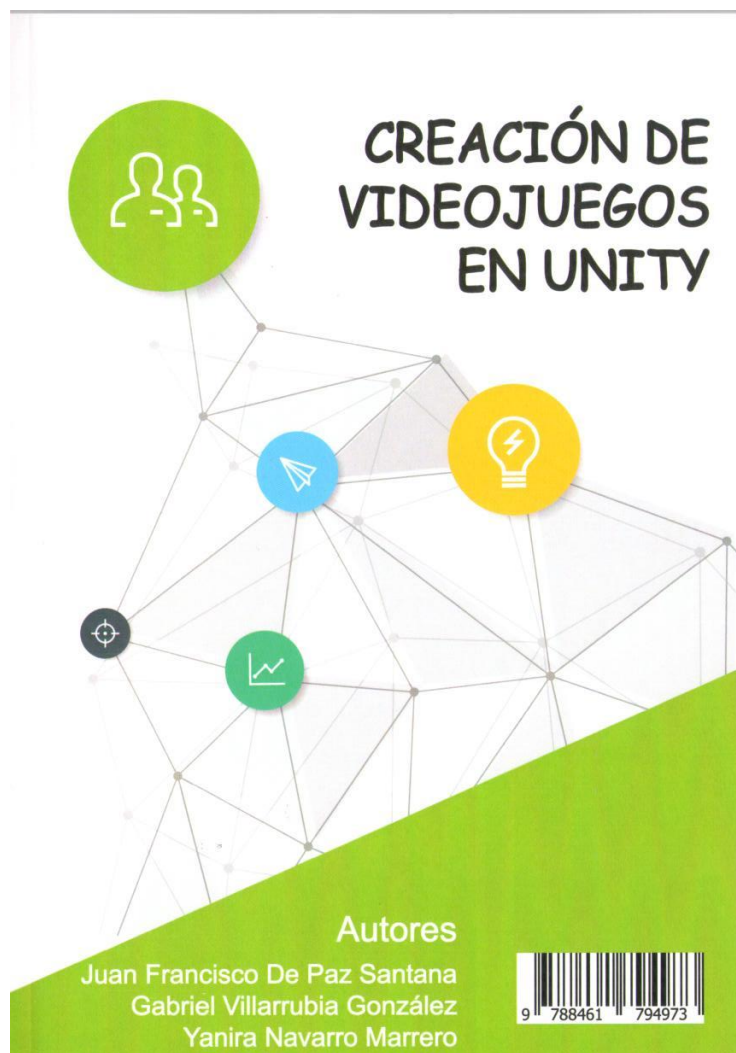
- Programación en lenguajes de bajo nivel: se corresponde con asignaturas básicas en las que se dan las bases para el desarrollo en ensamblador o con puertas lógicas.
- Programación en lenguajes de alto nivel: lenguajes de alto nivel junto con el uso de herramientas que faciliten la depuración y el desarrollo de aplicaciones.
- Programación gráfica/visual: uso de entornos que permiten la definición de la funcionalidad e interfaces usando herramientas visuales que simplifican la definición del código.

### 4.1 Elaboración de material docente

Para la impartición de la docencia mediante la programación gráfica/visual se ha elaborado el material docente siguiente:

- Creación de videojuegos en Unity. ISBN: 978-84-617-9497-3, 178 páginas, 2017.
- Gamemaker: creación de juegos en 2D0. ISBN: 978-84-617-9498-0, 115 páginas, 2017.

A continuación se muestran las capturas de pantalla de los dos libros editados de Unity Fig. 1, Fig. 2, Fig. 3, Fig. 4, Fig. 5, gamemaker Fig. 6, Fig. 7, Fig. 8, Fig. 9, Fig. 10.



**Fig. 1. Creación de videojuegos en Unity portada**



<b>Autores</b>	
Juan Francisco De Paz Santana fofob@usal.es Departamento Informática y Automática Universidad de Salamanca	
Gabriel Villarubia González gvg@usal.es Departamento Informática y Automática Universidad de Salamanca	
Yarita Navarro Marrero marayan@usal.es Universidad de Salamanca	
Este libro se ha realizado para apoyar a los alumnos de la Universidad de Salamanca del Grado de Ingeniería en Informática y Automática, y en particular a los estudiantes que cursan el primer curso de la carrera, indicando la procedencia de los mismos.	
La portada del libro ha sido diseñada por Rachel / Freepik.	
ISBN: 978-84-617-9497-3	
<b>Contenido</b>	
1	<i>Introducción</i> ..... 1
1.1	<b>Game objects</b> ..... 1
1.1.1	Ejercicio 1..... 3
1.2	<b>Prefab</b> ..... 4
1.3	<b>Luces</b> ..... 5
1.4	<b>Scripts</b> ..... 6
1.5	<b>Transformaciones</b> ..... 8
1.6	<b>Market</b> ..... 9
2	<i>Animación</i> ..... 10
2.1	<b>Animator Component</b> ..... 10
2.2	<b>Animator Controller</b> ..... 11
2.3	<b>Blend trees</b> ..... 19
2.3.1	Ejercicio 2 (ejemplo)..... 22
2.4	<b>Animation scripts</b> ..... 27
2.4.1	JavaScript..... 27
2.4.2	Ejercicio 3 (ejemplo)..... 31
2.4.3	C#..... 31
2.4.4	Ejercicio 4 (ejemplo)..... 32
3	<i>Físicas</i> ..... 34
3.1	<b>Colliders</b> ..... 34
3.1.1	Ejercicio 5..... 38
3.2	<b>Triggers vs Colliders (ejemplo 1)</b> ..... 38
3.3	<b>RigidBody (ejemplo 2)</b> ..... 39
3.4	<b>Physics materials</b> ..... 40
3.5	<b>RayCasting</b> ..... 42
3.5.1	Ejercicio 6 (ejemplo 1)..... 43
4	<i>Escenarios</i> ..... 46
4.1	<b>Terrenos</b> ..... 46
4.2	<b>Skybox</b> ..... 52
4.3	<b>Efectos de agua</b> ..... 56
4.4	<b>Viento</b> ..... 57
4.5	<b>Ejercicio 1</b> ..... 57
4.6	<b>Ejercicio 2</b> ..... 58
4.7	<b>Ejercicio 3</b> ..... 59
	i

Fig. 2. Índice Unity 1

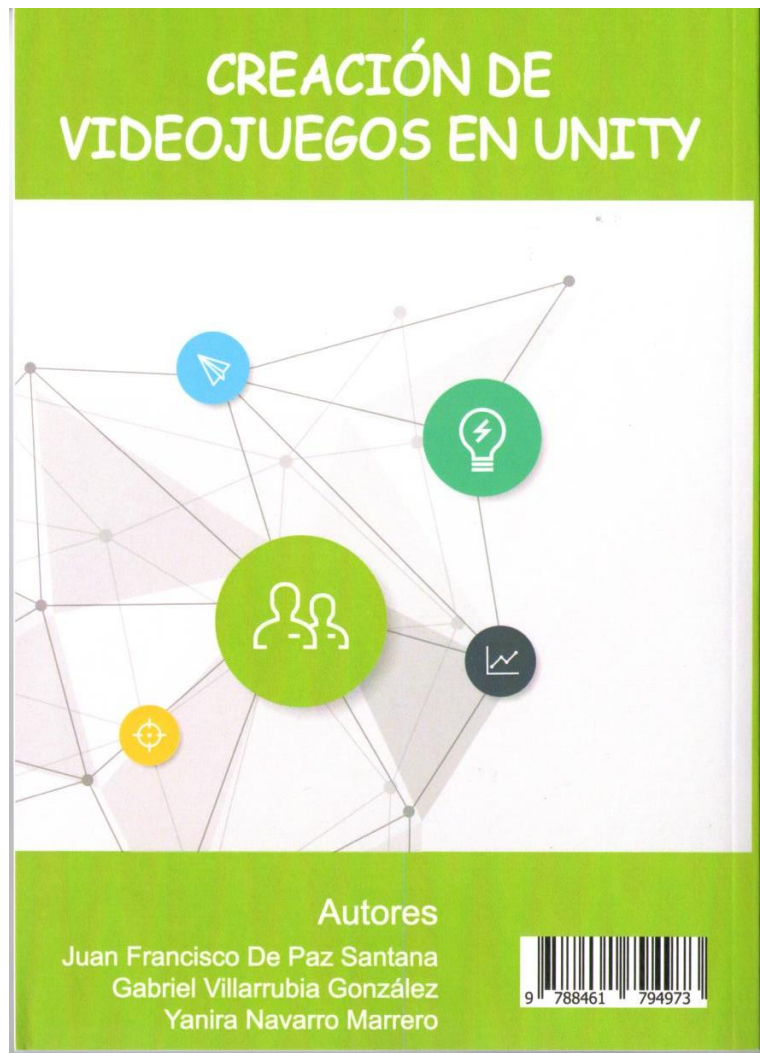


5	UI.....	61
5.1	Canvas.....	61
5.2	Botones.....	63
5.3	Text.....	68
5.4	InputField.....	69
5.5	EventSystem.....	70
5.6	Eventos.....	71
6	Scripting.....	73
6.1	Recordatorio C#.....	73
6.2	Modificadores de acceso.....	80
6.3	Ciclo de vida.....	81
6.4	Clases.....	82
6.5	Propiedades y métodos de GameObjects.....	83
6.6	Component.....	84
6.7	Transform.....	84
6.8	Vector3 y Vector2.....	84
6.9	Input.....	85
6.10	Time.....	86
6.11	Object.....	86
6.12	MonoBehaviour.....	87
6.13	Physics.....	87
6.14	Almacenar datos.....	87
6.15	SceneManager.....	87
6.16	RenderSettings.....	87
6.17	Depuración.....	88
7	Ejercicio 1.....	89
8	Ejercicio 2.....	93
8.1	Entorno.....	93
8.2	Movimiento jugador.....	95
8.3	Enemigos.....	100
8.4	Interfaz.....	108
8.5	Vida jugador.....	110
8.5.1	Ejercicio.....	112
8.6	Vida Enemigo.....	116
8.7	Crear enemigos.....	123
8.8	Cargar un escenario.....	126
9	Ejercicio 1 2D.....	129
9.1	Fondo.....	130
9.2	Música.....	133
9.3	Jugador y disparos.....	134
9.3.1	Animación y movimiento.....	134
9.3.2	Disparos.....	139
9.3.3	Colisiones.....	145
9.4	Enemigos.....	148
9.4.1	Animación y Movimiento.....	148
9.4.2	Colisiones.....	153
9.4.3	Ejercicio Disparos.....	157
9.5	GUI.....	157
9.6	Ejercicio.....	162
10	Ejercicio 2 2D.....	168

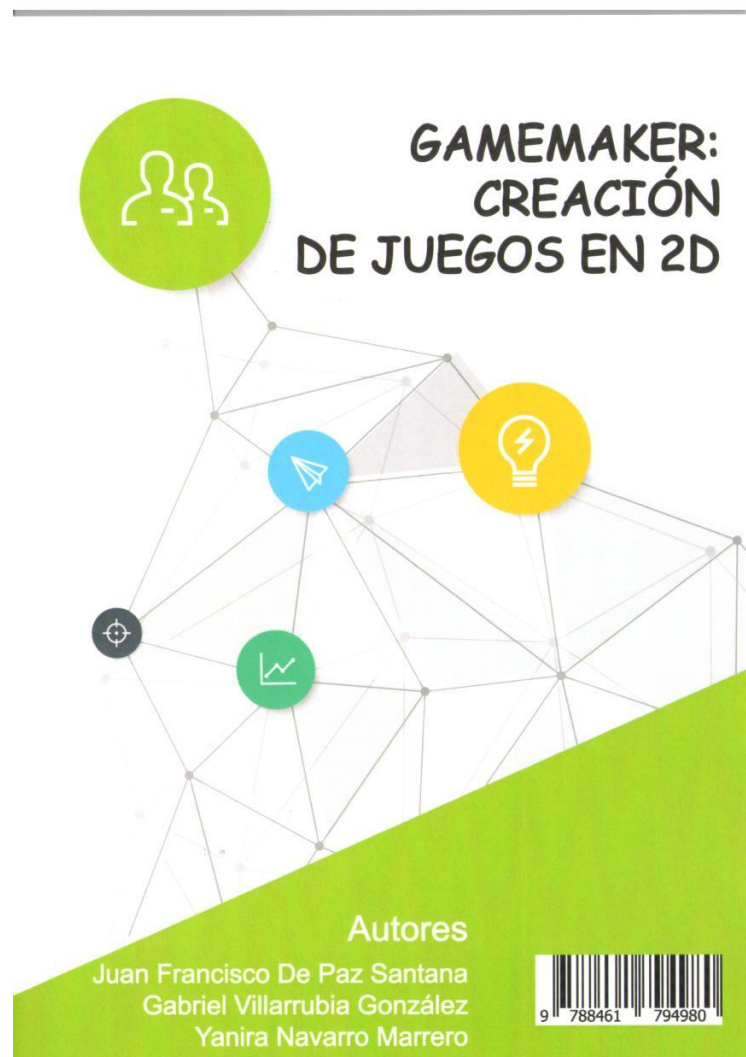
Fig. 3. Índice Unity 2



Fig. 4. Final Unity



**Fig. 5. Contraportada Unity**



**Fig. 6. Gamemaker: creación de juegos en 2D**





**Autores**

Juan Francisco De Paz Santana  
 jcofols@usal.es  
 Departamento Informática y Automática  
 Universidad de Salamanca

Gabriel Villarrubia González  
 gv@usal.es  
 Departamento Informática y Automática  
 Universidad de Salamanca

Yamira Navarro Murreto  
 murreto@usal.es  
 Universidad de Salamanca

Este libro se ha realizado para apoyo a los alumnos de la Universidad de Salamanca del Grado de Ingeniería en Informática o de cualquier estudiante con el objeto de que ayude durante su aprendizaje en el desarrollo y creación de juegos en 2D. Los recursos usados en este documento no son de elaboración propia y se ha indicado la procedencia de los mismos.  
 La portada del libro ha sido diseñada por Raquel / Freepik.

ISBN: 978-84-617-9498-0

**Contenido**

1	Introducción	1
2	Introducción a GameMaker (Ejercicio 1)	2
2.1	Añadir sprites	5
2.2	Sonidos	6
2.3	Añadiendo objetos	7
2.4	Eventos y acciones	9
2.4.1	Evento/acción creación	10
2.4.2	Evento/acción colisión	12
2.4.3	Evento de ratón	13
2.5	Fondo	14
2.6	Crear la escena	15
2.7	Añadir música y puntuación	17
2.8	Ejecutar	20
2.9	Ejercicio	20
3	Juegos de Scroll (Ejercicio 2)	20
3.1	Sprites	22
3.2	Sonidos	23
3.3	Crear Fondo	23
3.4	Objetos	23
3.5	Escenario	24
3.6	Constantes	26
3.7	Variables	27
3.8	Islas	29
3.9	Jugador	32
3.9.1	Disparos	34
3.10	Enemigos	37
3.10.1	Disparos	41
3.11	Vida	43
3.11.1	Colisiones	44
3.12	Puntuación	46
3.13	Ejercicio	48
4	Juego de plataformas (Ejercicio 3)	50
4.1	Cargar sprites	50
4.2	Background	51

Fig. 7. Índice gamemaker 1





## 1 Introducción

El desarrollo de videojuegos 2D estuvo muy extendido hasta finales de la década de los 90, posteriormente con la aparición de las videoconsolas como la PlayStation, Sega Saturn o la Nintendo 64 hubo un boom en el desarrollo de los juegos en 3D. Actualmente el desarrollo de videojuegos 2D ha sufrido un renacimiento por la aparición de dispositivos móviles y por la irrupción de HTML5.

Las técnicas de desarrollo de videojuegos en 2D siguen siendo las mismas desde sus orígenes. En este tipo de videojuegos las animaciones y movimientos se realizan mediante una serie de sprites, en la Fig. 1 se puede ver un ejemplo de en el videojuego Alex Kidd.

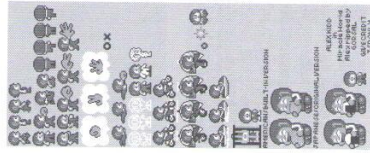
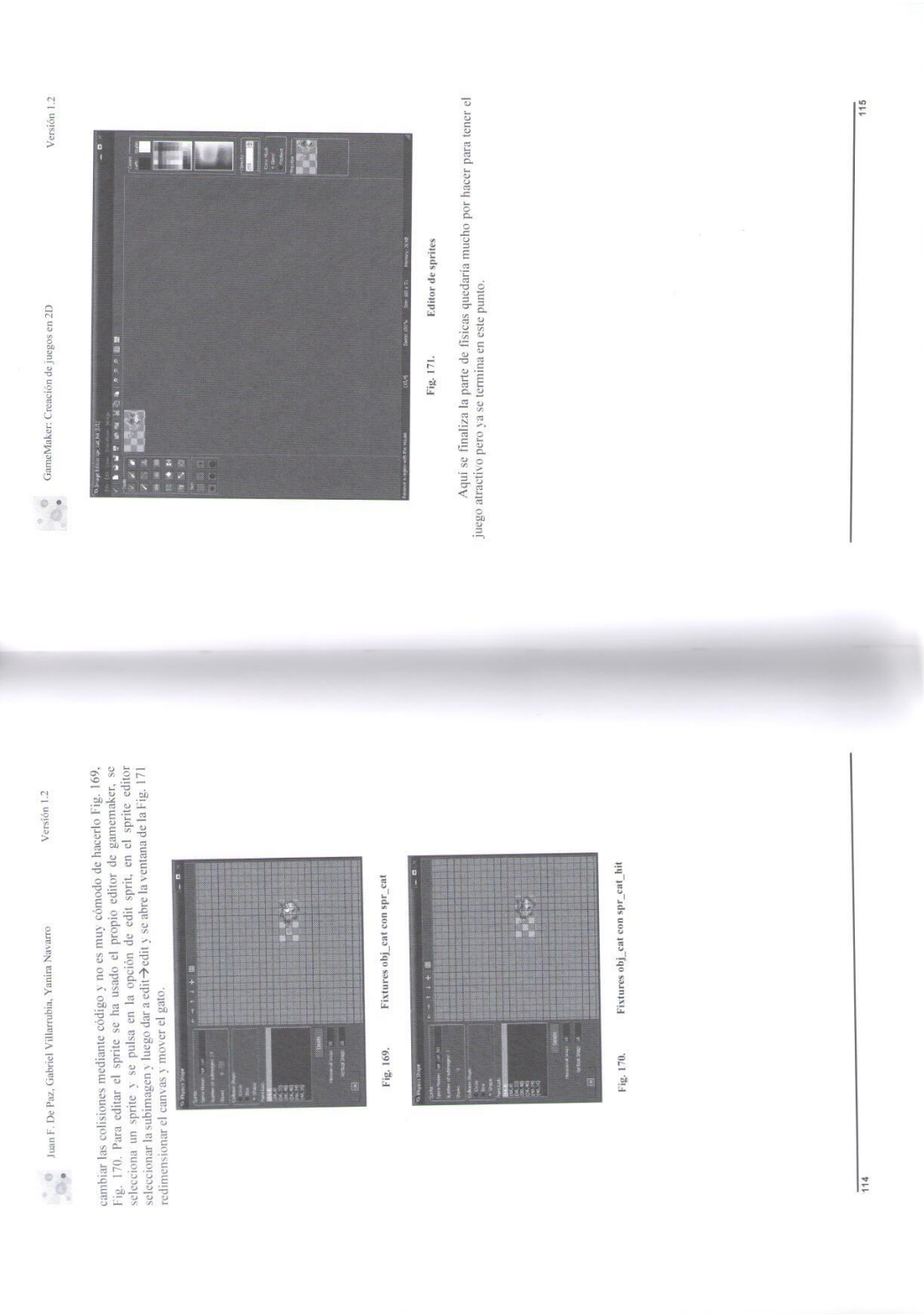


Fig. 1. Sprites Alex Kidd

En la Fig. 2 se puede ver al personaje del videojuego en dos posiciones diferentes, una de ella que está golpeando y en la otra está en estado de reposo. Estas dos posiciones no dejan de ser dos sprites diferentes de los vistos en la Fig. 1.

4.3	Objetos.....	52
4.4	Escenario.....	53
4.5	Rutas.....	57
4.6	Eventos/Acciones.....	58
4.7	Cargar más sprites.....	63
4.8	Instancias.....	65
<i>FÍSICAS</i> ..... 67		
5.1	Mundo, Ejemplo básico (ejercicio4).....	68
5.2	Fixtures.....	74
5.3	FUERZAS.....	78
5.4	Joints.....	83
5.5	Ejercicio 5 Joints.....	83
5.6	Ejercicio 6.....	90

Fig. 8. Índice gamemaker 2



cambiar las colisiones mediante código y no es muy cómodo de hacerlo Fig. 169. Fig. 170. Para editar el sprite se ha usado el propio editor de gamemaker, se selecciona un sprite y se pulsa en la opción de edit sprit, en el sprite editor seleccionar la subimagen y luego dar a edit → edit y se abre la ventana de la Fig. 171 redimensionar el canvas y mover el gato.

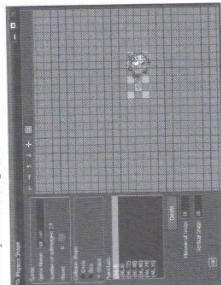


Fig. 169. Fixtures obj\_cat con spr\_cat

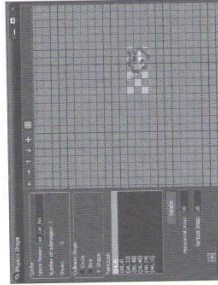


Fig. 170. Fixtures obj\_cat con spr\_cat\_bit

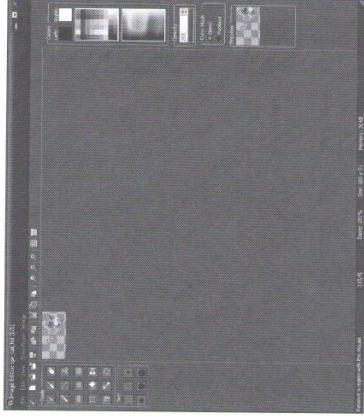
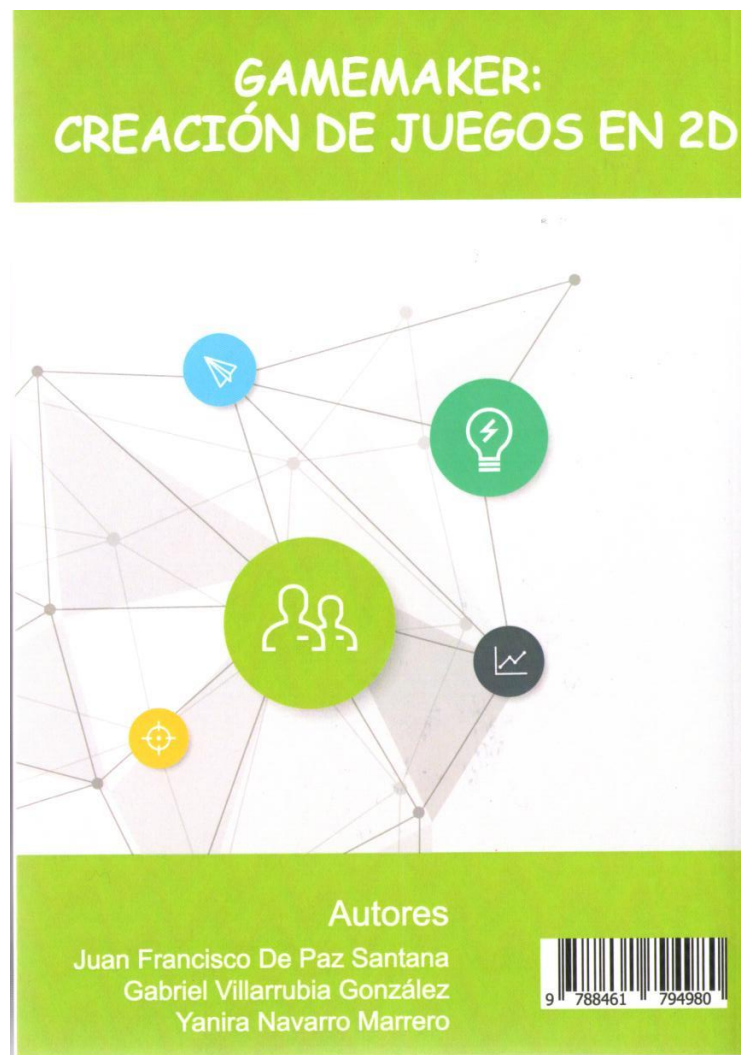


Fig. 171. Editor de sprites

Aquí se finaliza la parte de físicas quedaría mucho por hacer para tener el juego atractivo pero ya se termina en este punto.



**Fig. 10. Contraportada gamemaker**

Algunos ejemplos de programación gráfica o visual en Unity se tiene en las figuras Fig. 11, Fig. 12 .

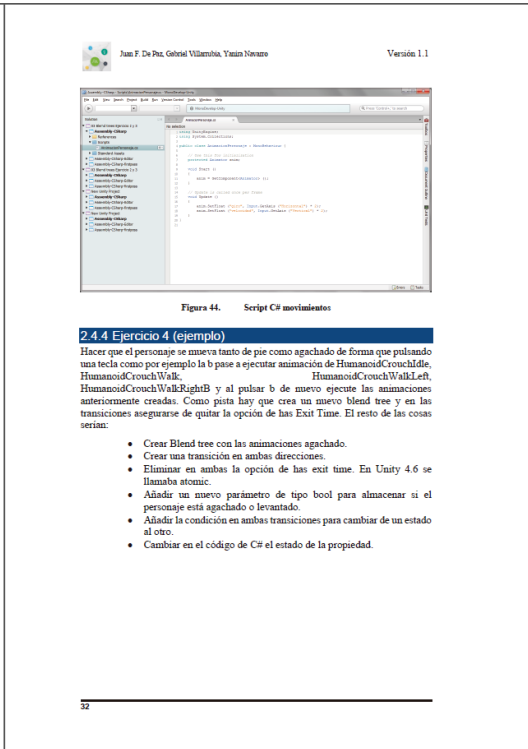
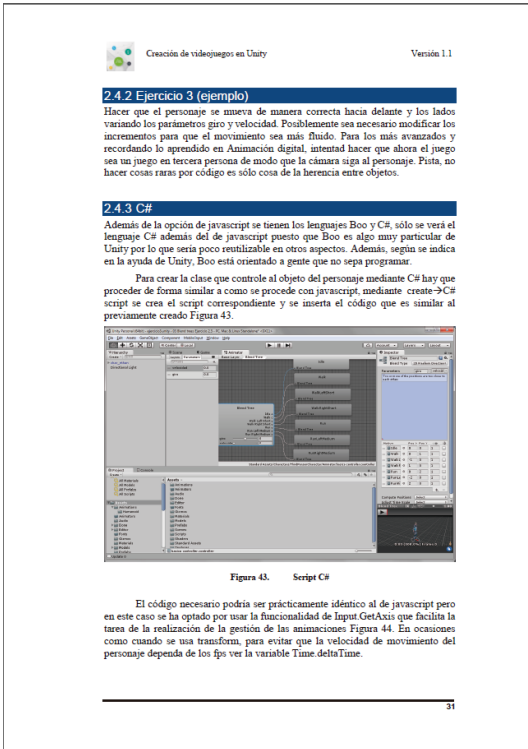


Fig. 11. Programación gráfica de movimientos personajes y código asociado en Unity

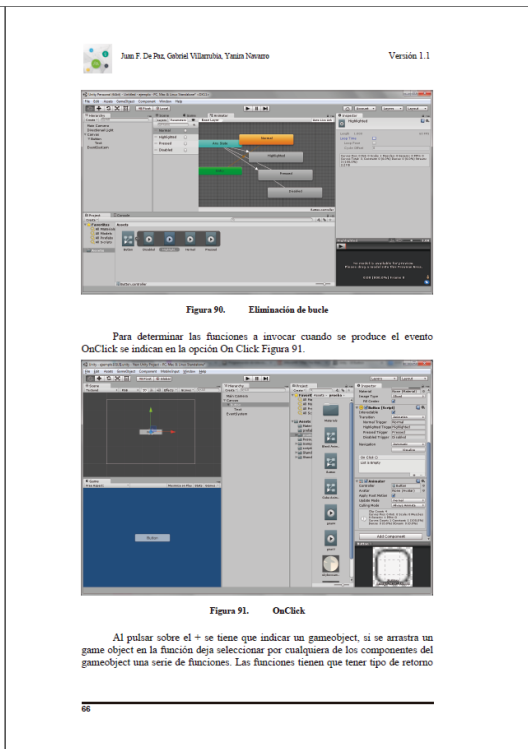
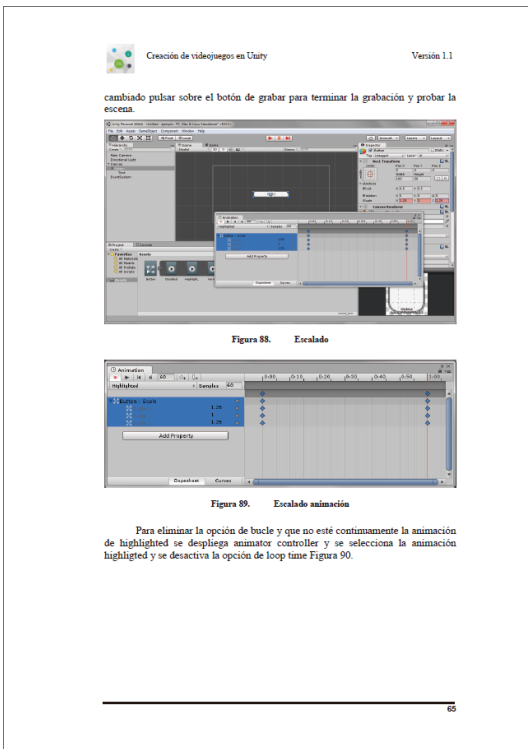
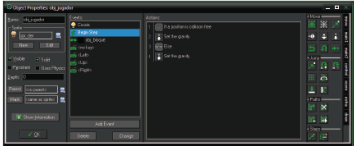

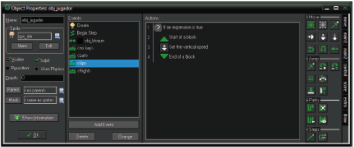



Fig. 12. Control de comportamiento de GUI en Unity



En el caso de gamemaker se tienen también ejemplos de programación gráfica Fig. 13.

 <p>Fig. 93. Evento/acción begin step gravedad</p>  <p>Fig. 94. Formularios para rellenar la información de begin step</p> <p style="text-align: right;">81</p>	 <p>Fig. 95. Evento/acción salto</p> <p>Si se realiza la prueba de saltar se puede ver que el personaje no detecta colisiones con el suelo y con las repisas por lo que hay que añadir eventos de colisión. Cuando la velocidad vertical sea negativa y se detecta una colisión se está chocando contra el techo, se pone la velocidad vertical a 0 y ya se activa automáticamente la gravedad según se indicó en begin step. Cuando la velocidad es positiva y se detecta una colisión se ha chocado contra el suelo se pone la velocidad vertical a cero y mueve el jugador hasta colisionar justo con el objeto sólido move-&gt;Jump-&gt;Move to Contact . Se hace como máximo se mueva 10pixels hasta llegar al contacto si no hay contacto el objeto no se mueve. Si se pone -1 se mueve el valor por defecto que son 1000pixels.</p> <p>El pseudocódigo es el siguiente:</p> <pre> if expression speedy0 is true   set the vertical speed to 0 else   set the vertical speed to 0   move in direction 270 at most 10 till a contact with solid objects     </pre> <p style="text-align: right;">82</p>
---	---

**Fig. 13. Lógica de negocio en Gamemaker**

El material elaborado está disponible para los alumnos en cursos de Moodle Fig. 14, Fig. 15 por lo que no necesitan la adquisición de los libros anteriormente mostrados:

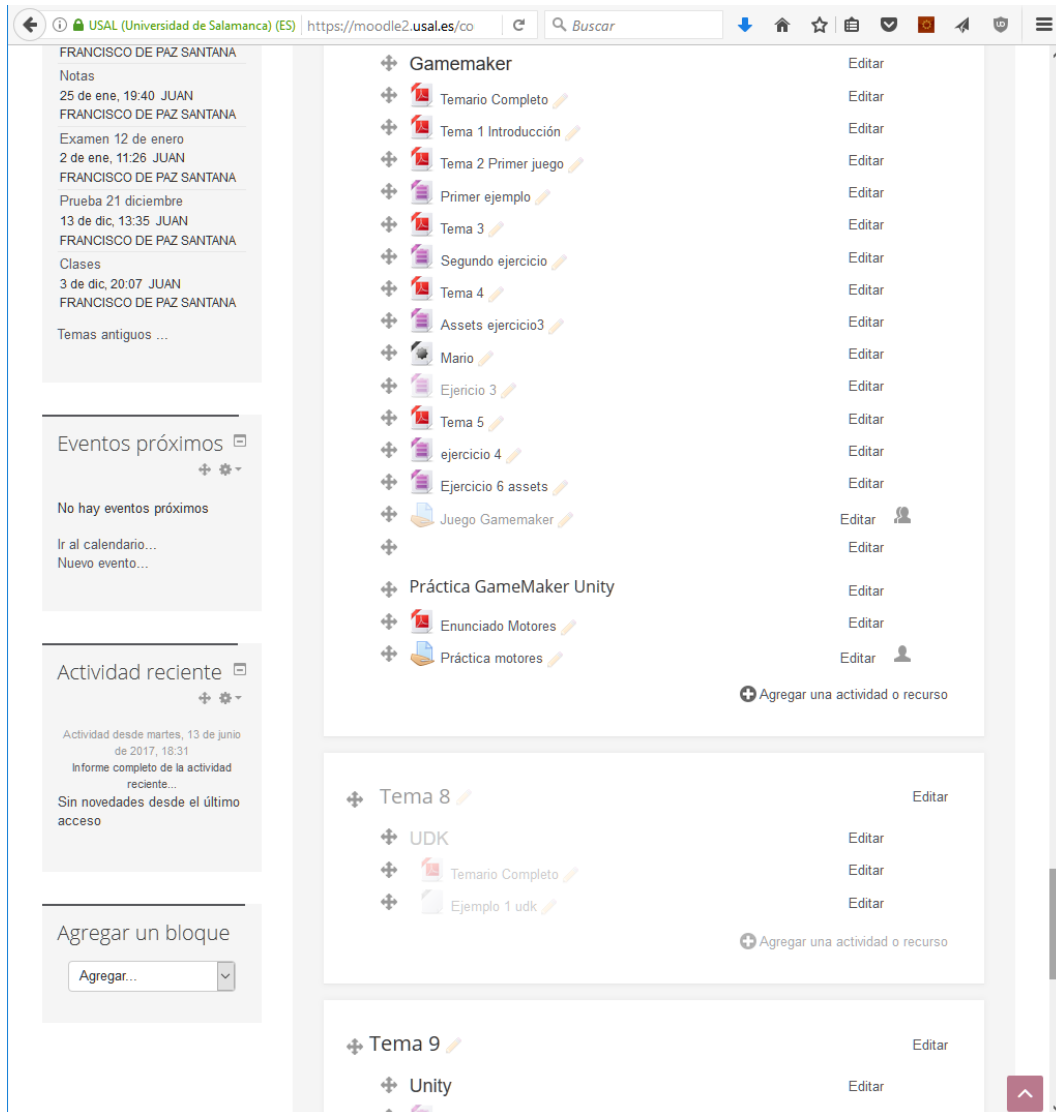


Fig. 14. Moodle 1

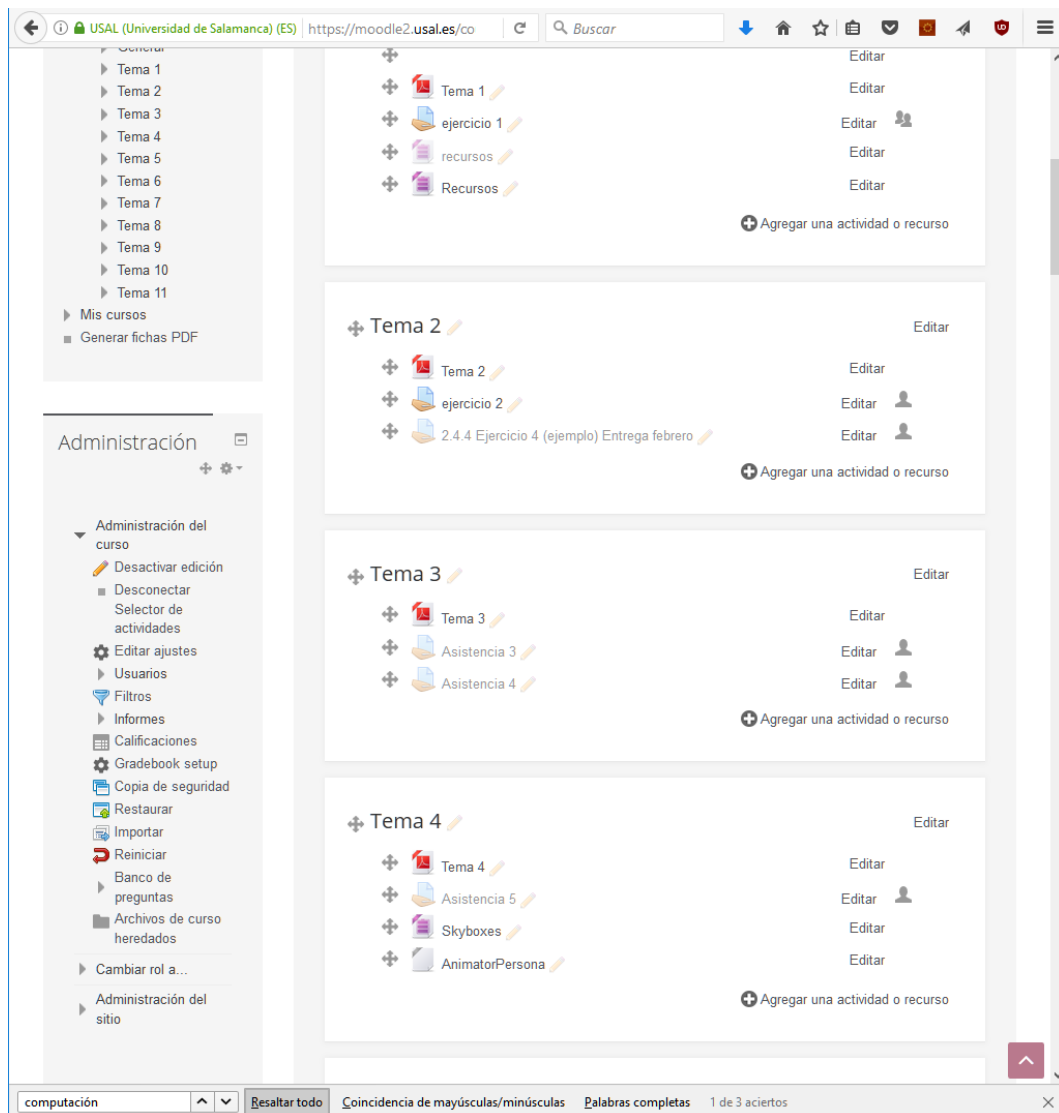


Fig. 15. Moodle

Además de este material desarrollado se parte también de material previo ya usado para las asignaturas indicadas en la memoria del proyecto. En este material se hace uso de lenguajes de programación de bajo nivel como ensamblador o puertas lógicas Fig. 16, lenguajes de alto nivel junto con herramientas de desarrollo que facilitan la depuración Fig. 17 y desarrollo de aplicaciones con herramientas de elaboración de interfaces para aplicaciones móviles Fig. 18.





```
Open ▾ 🔍 Save
1 module multiplexor4x1 (output wire y, input wire s0, input wire s1, input wire
  OE, input wire D0, input wire D1, input wire D2, input wire D3);
2
3   wire ns0, ns1, and0, and1, and2, and3, sal;
4   not (ns0, s0);
5   not (ns1, s1);
6   and (and0, ns0, ns1, D0);
7   and (and1, s0, ns1, D1);
8   and (and2, ns0, s1, D2);
9   and (and3, s0, s1, D3);
10  or (sal, and0, and1, and2, and3);
11  bufif1 (y, sal, OE);
12 endmodule
13
14 module multiplexor8x1 (output tri y, input wire D0, input wire D1, input wire
  D2, input wire D3, input wire D4, input wire D5, input wire D6, input wire D7,
  input wire S0, input wire S1, input wire S2);
15
16   wire nS2;
17   not (nS2, S2);
18   multiplexor4x1 m1 (y, S0, S1, nS2, D0, D1, D2, D3);
19   multiplexor4x1 m2 (y, S0, S1, S2, D4, D5, D6, D7);
20 endmodule
21
22 module test_mux8x1;
23
24   reg[7:0] D; // D0=D[0], D1=D[1], D2=D[2], D3=D[3], D4=D[4], D5=D[5], D6=D
  [6], D7=D[7]
25   reg[2:0] S; // S0=S[0], S1=S[1], S2=S[2]
26   tri y;
27   multiplexor8x1 M(y, D0, D1, D2, D3, D4, D5, D6, D7, S0, S1, S2);
28
29   assign D0=D[0]; assign D1=D[1]; assign D2=D[2]; assign D3=D[3];
30   assign D4=D[4]; assign D5=D[5]; assign D6=D[6]; assign D7=D[7];
31   assign S0=S[0]; assign S1=S[1]; assign S2=S[2];
32
Verilog ▾ Tab Width: 4 ▾ Ln 5, Col 5 ▾ INS
```

Fig. 16. Verilog



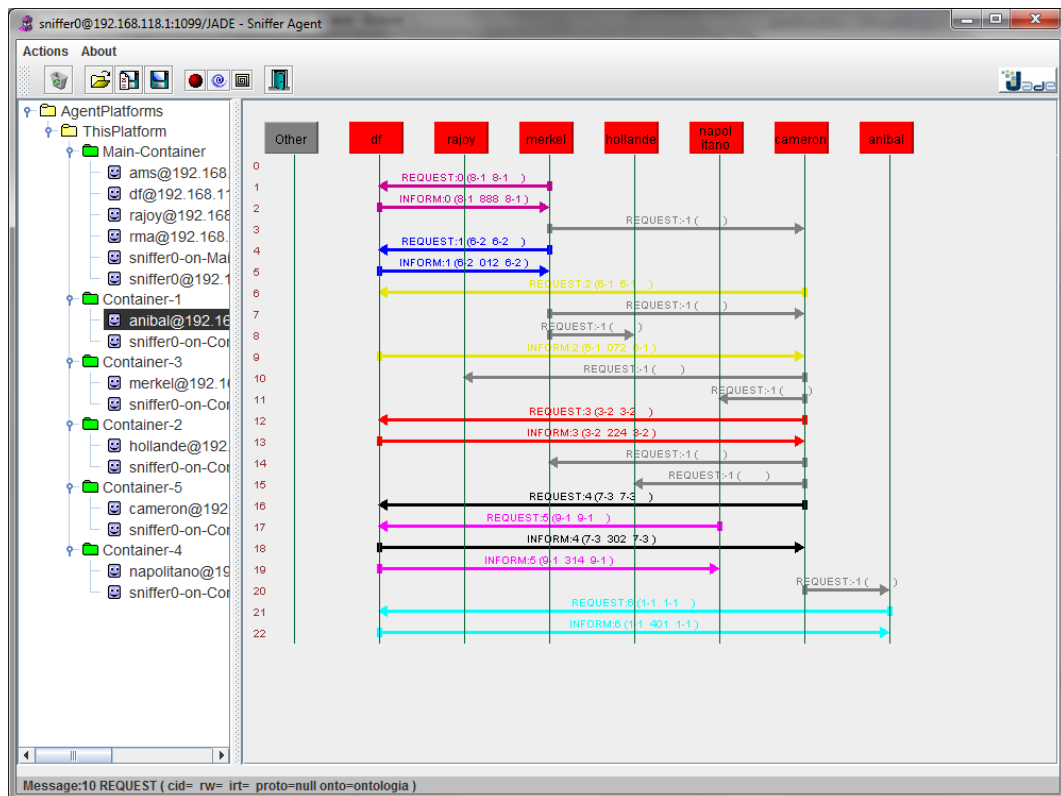


Fig. 17. Herramienta de depuración de mensajes distribuidos

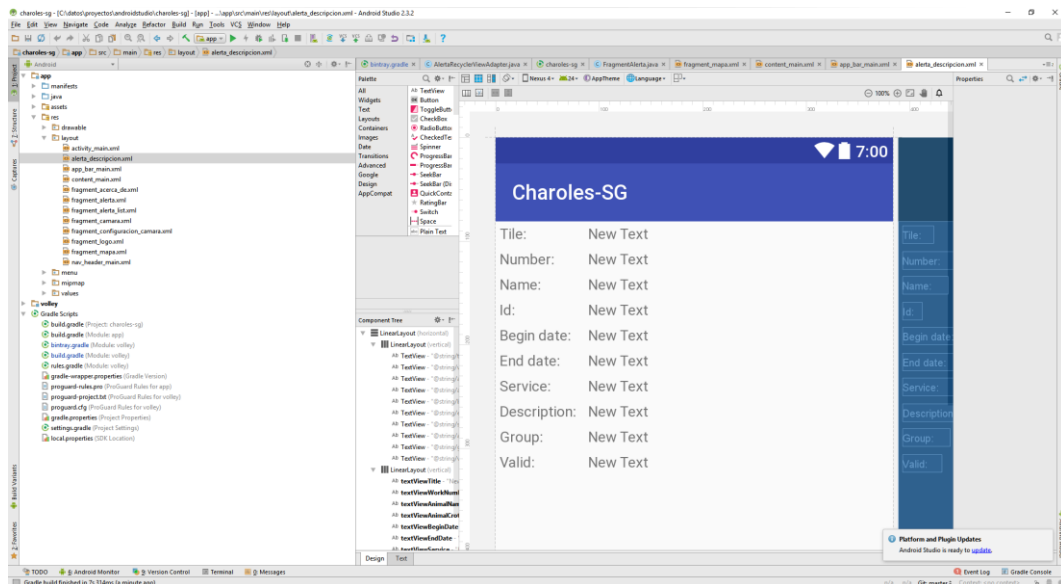


Fig. 18. Android Studio edición de interfaces gráficas



## 4.2 Valoración de los resultados

Para la valoración de los resultados se ha creado un formulario web en google que permite recoger la opinión de los alumnos de los últimos cursos del grado en Ingeniería Informática y del Máster en Ingeniería Informática. No tiene sentido hacer la encuesta a alumnos de primero o segundo ya que todavía no han aprendido a usar herramientas que permitan la programación gráfica/visual.

Para contestar es necesario validarse con una cuenta. La dirección del formulario es <https://goo.gl/forms/3IdUcWCYZFjO7Dsm1>, en las figuras Fig. 19, Fig. 20, Fig. 21.

Antes de iniciar el cuestionario se explica a los estudiantes lo que se considera programación gráfica/visual y se incluye la primera pregunta a modo de comprobación que realmente están considerando en serio a la encuesta. La pregunta número uno debería de ser siempre Sí puesto que el cuestionario está orientado a alumnos de últimos cursos del grado y del máster en Ingeniería Informática.



Nuevas tendencias de programación visual para la definición de la lógi... <https://docs.google.com/forms/d/1ay4pt47Z-13WxtuYZeqPgTSvQrFq...>

## Nuevas tendencias de programación visual para la definición de la lógica de las aplicaciones

Proyecto Innovación docente

\*Obligatorio

**Programación gráfica/visual. Se consideran aquellas técnicas que permiten definir parte de la funcionalidad o las interfaces mediante el uso de herramientas que reducen la necesidad de escritura de código**

---

1. **¿Estás familiarizado con la programación basada en código? \***

Marca solo un óvalo.

- Sí  
 No

2. **En qué tipo de aplicaciones has usado software que te permite la creación de partes sin la escritura de código mediante la programación gráfica o visual**

Selecciona todos los que correspondan.

- Aplicaciones móviles en Android o IOS  
 Videojuegos con herramientas como Unity, Gamemaker o similares  
 Aplicaciones de escritorio con Visual Stuidio o similares  
 Desarrollo web

3. **En general el uso de la programación visual/gráfica te ha facilitado la creación de aplicaciones? \***

Marca solo un óvalo.

- Sí  
 No

4. **En el caso de aplicaciones móviles ¿El uso de los IDEs te ha facilitado la creación de las interfaces? \***

Marca solo un óvalo.

- Sí  
 No

5. **En el caso de aplicaciones de escritorio ¿Los IDEs te han facilitado la creación de las interfaces? \***

Marca solo un óvalo.

- Sí  
 No

**Fig. 19. Encuesta 1**



Nuevas tendencias de programación visual para la definición de la lógi... <https://docs.google.com/forms/d/1ay4pt47Z-l3WxtuYZeqPgTSvQrFq...>

**6. ¿Prefieres el uso de IDEs para la creación de las interfaces o prefieres hacerlas de cero?**

Marca solo un óvalo.

- Prefiero el uso de IDEs
- Prefiero hacerlas desde cero
- Una combinación de ambas

**7. Señala las siguientes afirmaciones con las que estés de acuerdo con gamemaker**

Selecciona todos los que correspondan.

- El uso de herramientas como gamemaker facilita la creación de juegos reduciendo el código a escribir
- El uso de gamemaker facilita la definición de la lógica de manera gráfica o visual
- El uso de la herramienta reduce el tiempo de creación de juegos en 2D comparado con la programación tradicional basada en la escritura de código

**8. Señala las siguientes afirmaciones con las que estés de acuerdo con Unity**

Selecciona todos los que correspondan.

- El uso de herramientas como Unity facilita la creación de juegos reduciendo el código a escribir
- El uso de Unity facilita la definición de la lógica de movimientos de los personajes
- Permite reducir el tiempo de desarrollo comparado con bibliotecas básicas como OpenGL

**9. ¿El uso de gamemaker y Unity ha aumentado tu interés por el desarrollo de aplicaciones usando herramientas similares? \***

Marca solo un óvalo.

- Sí
- No

**10. Valora de 1 a 10 la posibilidad de creación aplicaciones complejas en ensamblador \***

Marca solo un óvalo.

	1	2	3	4	5	6	7	8	9	10	
Complicado	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sencillo

**11. Valora de 1 a 10 la posibilidad de creación aplicaciones complejas en lenguajes de alto nivel como java, C# etc sin el uso de IDEs que faciliten la programación visual \***

Marca solo un óvalo.

	1	2	3	4	5	6	7	8	9	10	
Complicado	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sencillo

**Fig. 20. Encuesta 2**



Nuevas tendencias de programación visual para la definición de la lógi... <https://docs.google.com/forms/d/1ay4pt47Z-l3WxtuYZeqPgTSvQrFq...>

12. Valora de 1 a 10 la posibilidad de creación aplicaciones complejas en lenguajes de alto nivel como java, C# etc mediante el uso de IDEs que faciliten la programación visual \*

Marca solo un óvalo.

1	2	3	4	5	6	7	8	9	10
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

13. Valora de 1 a 10 la posibilidad de creación juegos complejos con OpenGL \*

Marca solo un óvalo.

	1	2	3	4	5	6	7	8	9	10	
Complicado	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sencillo

14. Valora de 1 a 10 la posibilidad de creación juegos complejos con herramientas que incluyan programación gráfica/visual \*

Marca solo un óvalo.

	1	2	3	4	5	6	7	8	9	10	
Complicado	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sencillo

Con la tecnología de  
 Google Forms

Fig. 21. Encuesta 3

### 4.3 Resultados y conclusiones

A continuación, se adjunta los resultados obtenidos en la encuesta adjuntada a los estudiantes.



Programación gráfica/visual. Se consideran aquellas técnicas que permiten definir parte de la funcionalidad o las interfaces mediante el uso de herramientas que reducen la necesidad de escritura de código

¿Estás familiarizado con la programación basada en código?

23 respuestas



Fig. 22. Resultados de la encuesta pregunta 1

En qué tipo de aplicaciones has usado software que te permite la creación de partes sin la escritura de código mediante la programación gráfica o visual

23 respuestas

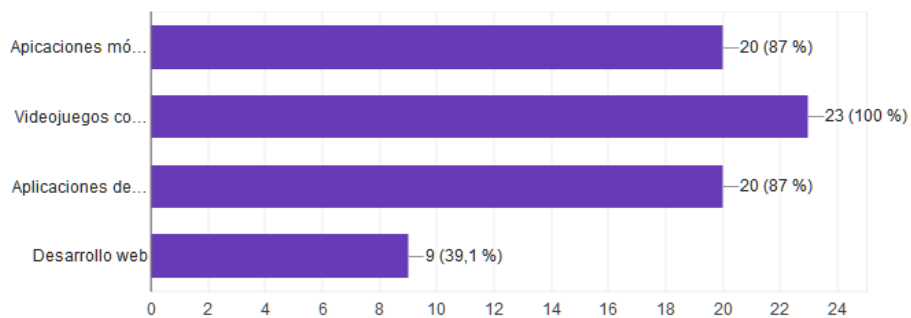
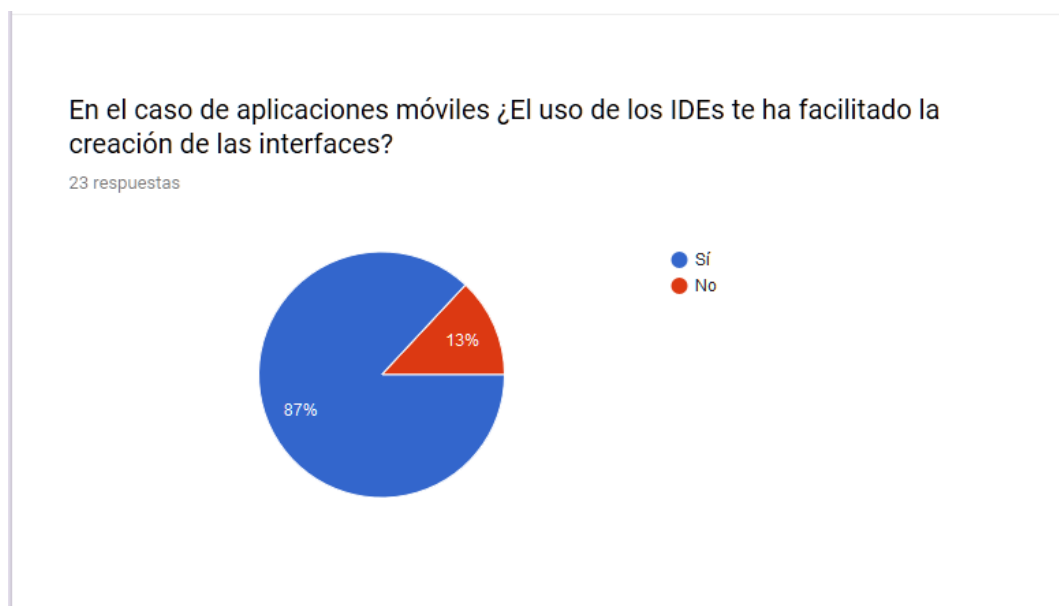


Fig. 23. Resultados de la encuesta pregunta 2



**Fig. 24. Resultados de la encuesta pregunta 3**



**Fig. 25. Resultados de la encuesta pregunta 4**



**Fig. 26. Resultados de la encuesta pregunta 5**



**Fig. 27. Resultados de la encuesta pregunta 6**



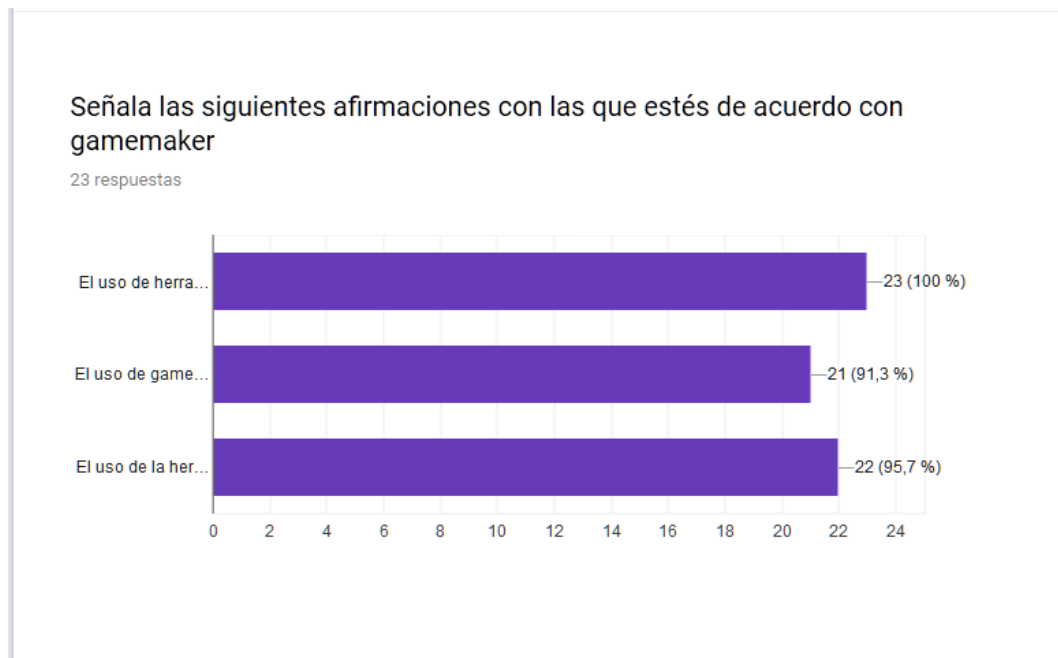


Fig. 28. Resultados de la encuesta pregunta 7

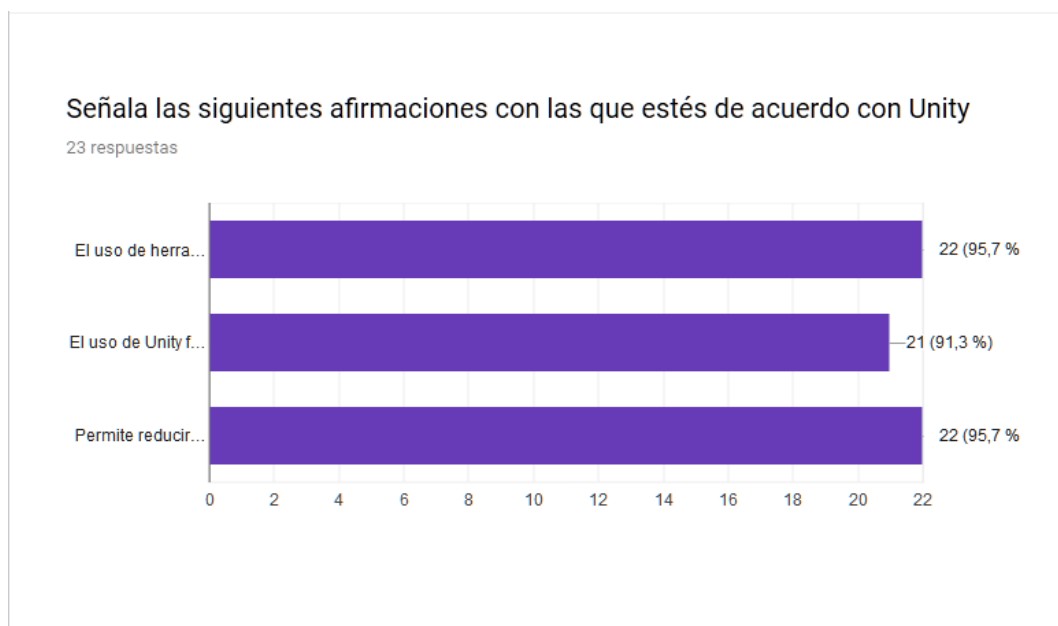
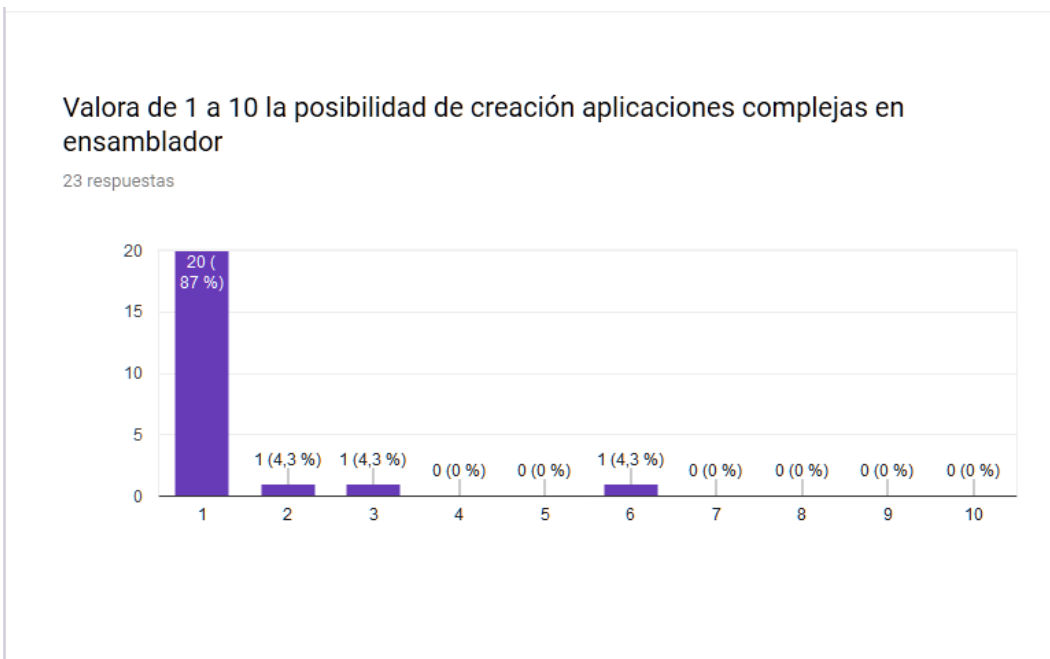


Fig. 29. Resultados de la encuesta pregunta 8



**Fig. 30. Resultados de la encuesta pregunta 9**



**Fig. 31. Resultados de la encuesta pregunta 10**

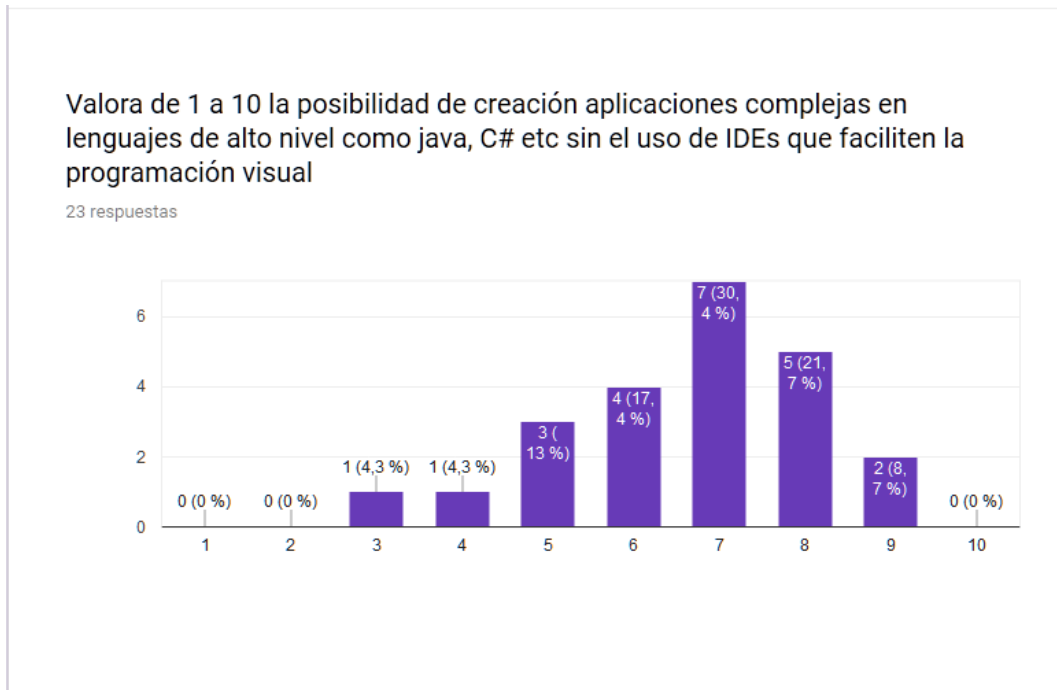


Fig. 32. Resultados de la encuesta pregunta 11

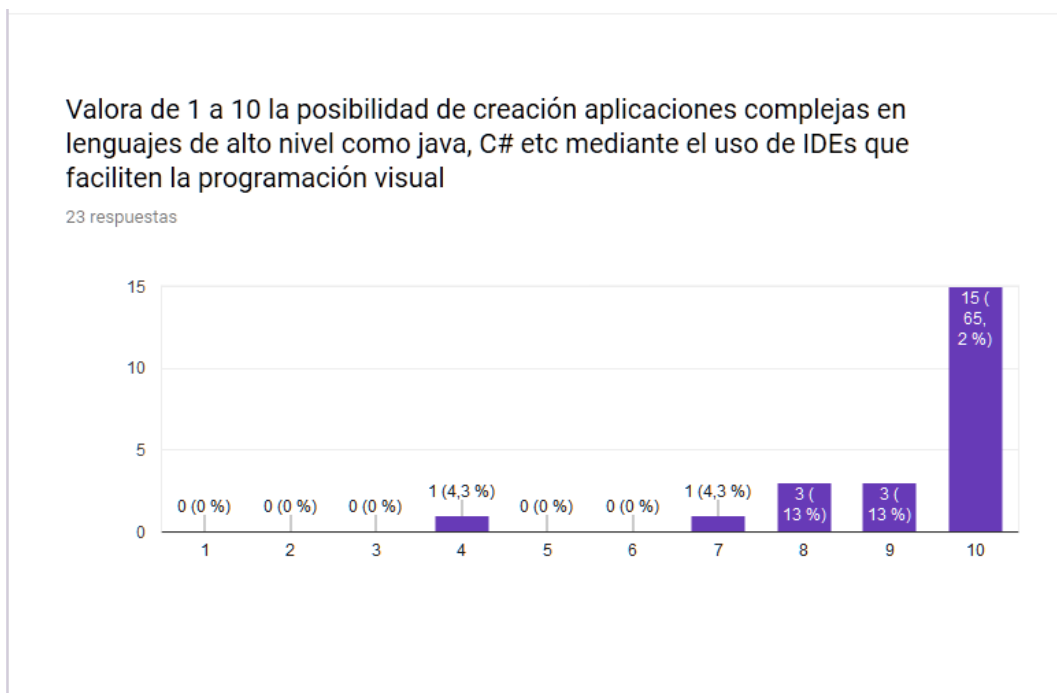
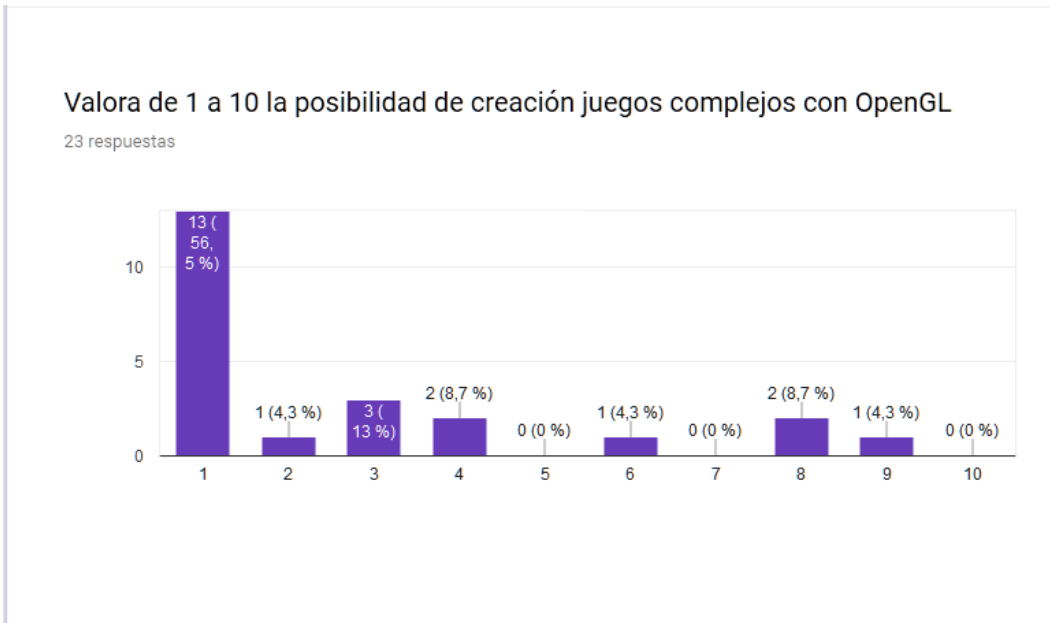
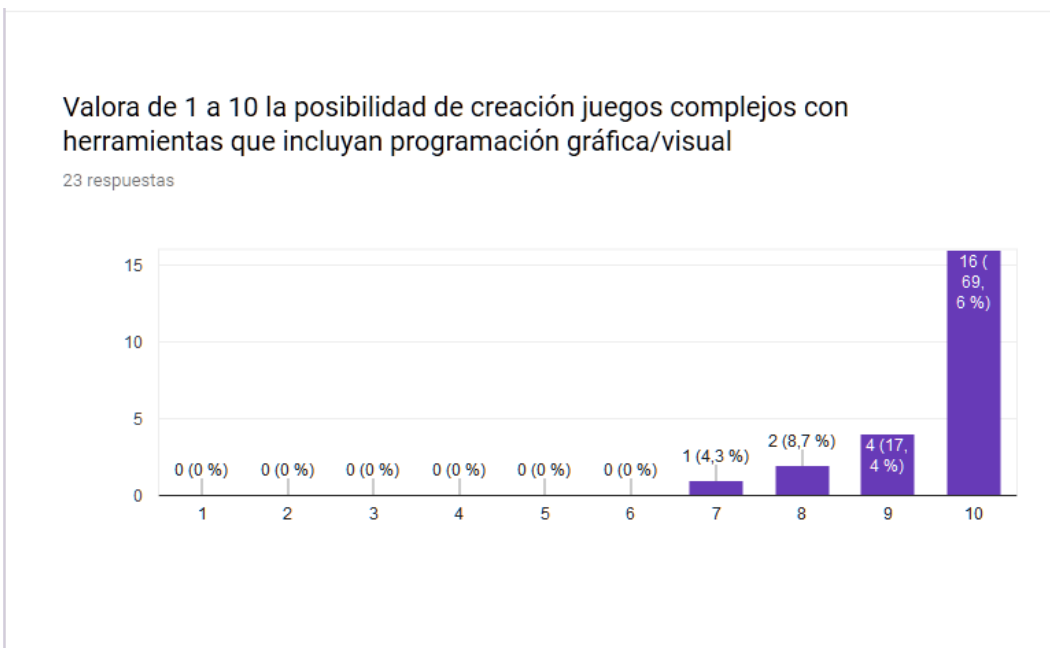


Fig. 33. Resultados de la encuesta pregunta 12



**Fig. 34. Resultados de la encuesta pregunta 13**



**Fig. 35. Resultados de la encuesta pregunta 14**

De los resultados de la encuesta se puede ver que los alumnos tienen una percepción positiva sobre el uso de programación gráfica para el desarrollo de las aplicaciones tal y como se puede ver en las preguntas 12 y 14.

En la pregunta 7 y 8 se puede ver que el uso de las herramientas manejadas en el proyecto de innovación docente ha facilitado a los alumnos el desarrollo de la funcionalidad y reducido el tiempo de desarrollo mejorando así los resultados finales.



Anteriormente a la elaboración del nuevo material docente en Animación digital se usaba para el desarrollo del software bibliotecas básicas para el desarrollo de aplicaciones 3D como OpenGL, DirectX y Java3D y bibliotecas de más alto nivel que no incorporaban herramientas de programación visual como OGRE. Este contenido estaba más alineado con las técnicas de programación clásicas que se dan a lo largo de la carrera.

Con la incorporación de las herramientas de programación gráfica/visual se ha conseguido reducir el tiempo de desarrollo y se ha incrementado considerablemente la calidad de los resultados finales obtenidos. Este aspecto es muy importante para aumentar la motivación y el interés de los estudiantes por la asignatura.

Durante el examen de la asignatura de Animación digital se dio la oportunidad de elegir la realización de un ejercicio entre dos existentes. El primero de los ejercicios se podía realizar casi íntegramente mediante programación visual, esta fue la opción elegida por 45 de los 47 alumnos examinados, cabe destacar que los alumnos redujeron el uso del código al mínimo posible. Los dos alumnos restantes optaron por la realización del ejercicio que requería el uso tanto de programación gráfica/visual y de programación. En base a estos resultados se comprobó que para la realización de ejercicios similares los alumnos optaban por la aplicación de técnicas de programación gráfica/visual para su resolución.