



**UNIVERSIDADE
DE VIGO**

ESCOLA SUPERIOR DE ENXEÑARÍA INFORMÁTICA

Memoria do Traballo de Fin de Grao que presenta

D. Héctor Rodríguez Salgado

para a obtención do Título de Graduado en Enxeñaría Informática

Herramienta de modelado de comportamiento en sistemas espaciales



Xuño, 2016

Traballo de Fin de Grao Nº: EI 15/16 - 028

Titor/a: Javier Rodeiro Iglesias

Área de coñecemento: Linguaxes e Sistemas Informáticos

Departamento: Informática

Agradecimientos

En primer lugar, quiero agradecer a mis padres todo el esfuerzo realizado durante estos años para que pueda lograr mis objetivos y hacer lo que realmente me gusta, además de los ánimos que me han dado día a día para mirar siempre hacia delante y no rendirme nunca.

En segundo lugar, agradecerle a Javi la oportunidad de poder realizar este proyecto y de ayudarme a mejorar semana tras semana.

Por último, me gustaría agradecer a mi familia y compañeros la ayuda prestada. Pero, sobre todo, agradecer a mis amigos el apoyo incondicional a lo largo de la realización de este proyecto, que siempre han estado apoyándome en los peores momentos.

A todos ellos, muchas gracias.

Índice de contenidos

I. Memoria.....	1
1. Descripción del proyecto	1
1.1. Introducción	1
1.2. Objetivos	2
1.3. Descripción general de la aplicación.....	2
2. Descripción técnica	3
2.1. Metodología.....	3
2.2. Arquitectura técnica.....	4
2.3. Tecnologías y lenguajes de programación	4
2.4. Herramientas.....	5
3. Proceso de desarrollo y planificación	6
3.1. Descripción del proceso de desarrollo.....	6
3.2. Planificación y estimación temporal	7
3.3. Estimación de costes.....	8
3.4. Tiempos de ejecución y coste.....	10
3.5. Documentación entregada.....	12
4. Problemas y soluciones.....	13
4.1. Problemas encontrados	13
4.2. Soluciones	13
5. Posibles ampliaciones.....	14
6. Conclusiones.....	15
6.1. Tecnologías y herramientas	15
6.2. Planificación y gestión	15
6.3. Personal.....	16

II. Manual técnico	17
1. Introducción.....	17
1.1. Descripción del sistema	17
2. Documentación inicial.....	17
2.1. Redacción del anteproyecto	17
3. Análisis	18
3.1. Tecnologías a utilizar	18
3.2. Modelos existentes.....	19
3.3. Sistema a modelar y estrategia de desarrollo	23
3.4. Descripción de casos de uso.....	33
4. Diseño	36
4.1. Interfaz de usuario.....	36
4.2. Arquitectura de clases	39
4.3. Arquitectura general	40
4.4. Diagrama de secuencia del sistema	41
5. Implementación.....	42
5.1. Iteración 2	42
5.2. Iteración 3	65
5.3. Iteración 4	77
5.4. Iteración 5	92
6. Pruebas.....	99
7. Documentación final	102

III. Manual de usuario	103
1. Introducción	103
2. Requisitos del sistema	103
3. Instalación	103
4. Guía de uso.....	104
4.1. Interfaz de configuración	104
4.2. Interfaz de visualización	113
IV. Bibliografía	116
1. Artículos	116
2. Libros.....	116
3. Documentaciones técnicas.....	116
4. Otros	116

Índice de ilustraciones

<i>Ilustración 1 – Metodología iterativa e incremental.....</i>	3
<i>Ilustración 2 – Modelo cliente-servidor.....</i>	4
<i>Ilustración 3 – Planificación y estimación temporal</i>	7
<i>Ilustración 4 – Excepciones en el calendario.....</i>	8
<i>Ilustración 5 – Diagrama de Gantt</i>	8
<i>Ilustración 6 – Coste de hardware.....</i>	8
<i>Ilustración 7 – Coste de software</i>	9
<i>Ilustración 8 – Coste de personal</i>	9
<i>Ilustración 9 – Coste total.....</i>	9
<i>Ilustración 10 - Tiempos de ejecución real</i>	10
<i>Ilustración 11 - Diagrama de Gantt - Ejecución real</i>	11
<i>Ilustración 12 - Tiempos de ejecución - Documentación inicial.....</i>	17
<i>Ilustración 13 - Tiempos de ejecución - Tecnología a utilizar</i>	19
<i>Ilustración 14 – El juego de la vida - Formas fijas</i>	20
<i>Ilustración 15 – El juego de la vida - Formas deslizantes</i>	21
<i>Ilustración 16 - Tiempos de ejecución - Modelos existentes (1)</i>	23
<i>Ilustración 17 - Tiempos de ejecución - Modelos existentes (2)</i>	23
<i>Ilustración 18 - Etapa 1</i>	24
<i>Ilustración 19 - Etapa 2 – Primera versión</i>	25
<i>Ilustración 20 - Etapa 2 – Segunda versión</i>	26
<i>Ilustración 21 - Etapa 2 – Tercera versión</i>	27
<i>Ilustración 22 - Etapa 2 – Cuarta versión</i>	28
<i>Ilustración 23 - Primera versión - Prueba 1 - Visualización</i>	30
<i>Ilustración 24 - Primera versión - Prueba 1 - Registro.....</i>	30
<i>Ilustración 25 - Primera versión - Prueba 2 - Visualización</i>	30
<i>Ilustración 26 - Primera versión - Prueba 2 - Registro.....</i>	31
<i>Ilustración 27 - Ampliación primera versión.....</i>	31
<i>Ilustración 28 - Tiempos de ejecución - Sistema a modelar y estrategia de desarrollo (1).....</i>	32
<i>Ilustración 29 - Tiempos de ejecución - Sistema a modelar y estrategia de desarrollo (2).....</i>	32
<i>Ilustración 30 - Diagrama de casos de uso.....</i>	33
<i>Ilustración 31 - Descripción detallada - Escoger idioma</i>	33
<i>Ilustración 32 - Descripción detallada - Escoger sistema</i>	34
<i>Ilustración 33 - Descripción detallada - Escoger configurador</i>	34

<i>Ilustración 34 - Descripción detallada - Introducir parámetros de configuración</i>	34
<i>Ilustración 35 - Descripción detallada - Cargar archivo de configuración</i>	35
<i>Ilustración 36 - Descripción detallada - Cargar archivo de configuración</i>	35
<i>Ilustración 37 - Descripción detallada - Descargar ficheros</i>	35
<i>Ilustración 38 - Interfaz de usuario - Inicio</i>	36
<i>Ilustración 39 - Interfaz de usuario - Elección de sistema</i>	36
<i>Ilustración 40 - Interfaz de usuario - Carga de archivo de configuración</i>	37
<i>Ilustración 41 - Interfaz de usuario – Parametrización</i>	37
<i>Ilustración 42 - Interfaz de usuario – Visualización</i>	38
<i>Ilustración 43 - Tiempos de ejecución - Diseño de la interfaz de usuario</i>	38
<i>Ilustración 44 - Diagrama de clases inicial</i>	39
<i>Ilustración 45 - Diagrama de clases final</i>	39
<i>Ilustración 46 - Arquitectura general</i>	40
<i>Ilustración 47 - Diagrama de secuencia del sistema</i>	41
<i>Ilustración 48 - Diagrama de actividad - Creación del mundo y visualización del mundo</i>	43
<i>Ilustración 49 - Diagrama de actividad - Configuración de los estados del mundo</i>	43
<i>Ilustración 50 - Iteración 2 - Interfaz de usuario - Configuración y creación del mundo (1)</i>	44
<i>Ilustración 51 - Iteración 2 - Interfaz de usuario - Configuración y creación del mundo (2)</i>	44
<i>Ilustración 52 - Tiempos de ejecución - Iteración 2 - Configuración y creación del mundo</i>	45
<i>Ilustración 53 - Tiempos de ejecución - Iteración 2 - Configuración y creación del mundo</i>	45
<i>Ilustración 54 - Diagrama de actividad - Creación de un elemento</i>	46
<i>Ilustración 55 - Diagrama de actividad - Modificación y eliminación de un elemento</i>	47
<i>Ilustración 56 - Diagrama de actividad - Colocación de un elemento</i>	47
<i>Ilustración 57 - Iteración 2 - Diagrama de clases</i>	48
<i>Ilustración 58 - Iteración 2 - Interfaz de usuario – Gestión de elementos</i>	48
<i>Ilustración 59 - Tiempos de ejecución - Iteración 2 - Gestión de elementos</i>	49
<i>Ilustración 60 - Diagrama de actividad - Gestor de acciones</i>	51
<i>Ilustración 61 - Diagrama de actividad - Control de acciones por ciclo</i>	52
<i>Ilustración 62 - Diagrama de actividad - Acción – Ver</i>	53
<i>Ilustración 63 - Diagrama de actividad - Acción - Moverse</i>	54
<i>Ilustración 64 - Acción - Moverse - Método (puede moverse)</i>	54
<i>Ilustración 65 - Diagrama de actividad - Acción – Comer</i>	55
<i>Ilustración 66 - Diagrama de actividad - Acción – Dormir</i>	55
<i>Ilustración 67 - Diagrama de actividad - Acción – Olfatear</i>	56
<i>Ilustración 68 - Diagrama de actividad - Acción – Oír</i>	57

<i>Ilustración 69 - Diagrama de actividad - Acción – Reproducirse.....</i>	58
<i>Ilustración 70 - Iteración 2 - Interfaz de usuario - Acciones de los elementos (1)</i>	59
<i>Ilustración 71 - Iteración 2 - Interfaz de usuario - Acciones de los elementos (2)</i>	59
<i>Ilustración 72 - Iteración 2 - Interfaz de usuario - Acciones de los elementos (3)</i>	59
<i>Ilustración 73 - Iteración 2 - Interfaz de usuario - Acciones de los elementos (4)</i>	60
<i>Ilustración 74 - Iteración 2 - Diagrama de actividad - Principal</i>	61
<i>Ilustración 75 - Tiempos de ejecución - Iteración 2 - Acciones de los elementos</i>	62
<i>Ilustración 76 - Tiempos de ejecución - Iteración 2</i>	64
<i>Ilustración 77 - Tiempos de ejecución - Iteración 2 - Pruebas</i>	64
<i>Ilustración 78 - Tiempos de ejecución - Iteración 3 - Configuración y creación del mundo (1) ...</i>	66
<i>Ilustración 79 - Tiempos de ejecución - Iteración 3 - Configuración y creación del mundo (2) ...</i>	67
<i>Ilustración 80 - Diagrama de actividad - Colocación de elementos</i>	68
<i>Ilustración 81 - Diagrama de clases - Iteración 3</i>	68
<i>Ilustración 82 - Tiempos de ejecución - Iteración 3 - Gestión de elementos</i>	69
<i>Ilustración 83 - Diagrama de actividad - Acción - Moverse.....</i>	71
<i>Ilustración 84 - Diagrama de actividad - Acción – Dormir</i>	72
<i>Ilustración 85 - Diagrama de actividad - Acción - Reproducirse</i>	73
<i>Ilustración 86 - Iteración 3 - Diagrama de actividad - Principal</i>	74
<i>Ilustración 87 - Tiempos de ejecución - Iteración 3 - Acciones de los elementos</i>	75
<i>Ilustración 88 - Tiempos de ejecución – Iteración 3</i>	76
<i>Ilustración 89 - Tiempos de ejecución - Iteración 3 - Pruebas</i>	76
<i>Ilustración 90 - Diagrama de actividad - Almacenamiento de los parámetros iniciales</i>	77
<i>Ilustración 91 - Iteración 4 - Configuración y creación del mundo - Interfaz de usuario</i>	79
<i>Ilustración 92 - Tiempos de ejecución - Iteración 4 - Configuración y creación del mundo (1) ...</i>	79
<i>Ilustración 93 - Tiempos de ejecución - Iteración 4 - Configuración y creación del mundo (2) ...</i>	79
<i>Ilustración 94 - Diagrama de clases - Iteración 4</i>	81
<i>Ilustración 95 - Iteración 4 - Gestión de elementos - Interfaz de usuario</i>	82
<i>Ilustración 96 - Tiempos de ejecución - Iteración 4 - Gestión de elementos</i>	82
<i>Ilustración 97 - Diagrama de actividad - Acción - Mover.....</i>	84
<i>Ilustración 98 - Diagrama de actividad - Acción - Dormir</i>	85
<i>Ilustración 99 - Diagrama de actividad - Acción - Reproducirse</i>	86
<i>Ilustración 100 - Iteración 4 - Acciones de los elementos - Interfaz de usuario (1).....</i>	87
<i>Ilustración 101 - Iteración 4 - Acciones de los elementos - Interfaz de usuario (2).....</i>	87
<i>Ilustración 102 - Iteración 4 - Acciones de los elementos - Interfaz de usuario (3).....</i>	87
<i>Ilustración 103 - Iteración 4 - Acciones de los elementos - Interfaz de usuario (4)</i>	88

<i>Ilustración 104 - Iteración 4 - Acciones de los elementos - Interfaz de usuario (5)</i>	88
<i>Ilustración 105 - Iteración 4 - Diagrama de actividad - Principal</i>	89
<i>Ilustración 106 - Tiempos de ejecución - Iteración 4 - Acciones de los elementos (1)</i>	90
<i>Ilustración 107 - Tiempos de ejecución - Iteración 4 - Acciones de los elementos (2)</i>	90
<i>Ilustración 108 - Tiempos de ejecución – Iteración 4</i>	91
<i>Ilustración 109 - Tiempos de ejecución - Iteración 4 - Pruebas</i>	92
<i>Ilustración 110 - Tiempos de ejecución - Iteración 5 - Archivos de log</i>	94
<i>Ilustración 111 - Tiempos de ejecución - Iteración 5 - Multilenguaje (1)</i>	95
<i>Ilustración 112 - Tiempos de ejecución - Iteración 5 - Multilenguaje (2)</i>	96
<i>Ilustración 113 - Interfaz de usuario (1)</i>	96
<i>Ilustración 114 - Interfaz de usuario (2)</i>	97
<i>Ilustración 115 - Interfaz de usuario (3)</i>	97
<i>Ilustración 116 - Interfaz de usuario (4)</i>	98
<i>Ilustración 117 - Interfaz de usuario (5)</i>	98
<i>Ilustración 118 - Interfaz de usuario (6)</i>	98
<i>Ilustración 119 - Interfaz de usuario (7)</i>	99
<i>Ilustración 120 - Tiempos de ejecución - Iteración 5 - Interfaz de usuario</i>	99
<i>Ilustración 121 - Tiempos de ejecución – Pruebas.</i>	100
<i>Ilustración 122 - Tiempos de ejecución - Pruebas - Test 1</i>	100
<i>Ilustración 123 - Tiempos de ejecución - Pruebas - Test 2</i>	100
<i>Ilustración 124 - Tiempos de ejecución - Pruebas - Test 3</i>	100
<i>Ilustración 125 - Tiempos de ejecución - Pruebas - Test 4</i>	100
<i>Ilustración 126 - Tiempos de ejecución - Pruebas - Test 5</i>	100
<i>Ilustración 127 - Tiempos de ejecución - Pruebas - Test 6</i>	100
<i>Ilustración 128 - Tiempos de ejecución - Pruebas - Test 7</i>	101
<i>Ilustración 129 - Tiempos de ejecución - Pruebas - Test 8</i>	101
<i>Ilustración 130 - Tiempos de ejecución - Pruebas - Test 9</i>	101
<i>Ilustración 131 - Tiempos de ejecución - Pruebas - Test 10</i>	101
<i>Ilustración 132 - Tiempos de ejecución - Pruebas - Test 11</i>	101
<i>Ilustración 133 - Tiempos de ejecución - Pruebas</i>	101
<i>Ilustración 134 - Tiempos de ejecución - Documentación final</i>	102
<i>Ilustración 135 - Manual de usuario - Interfaz de configuración (1)</i>	104
<i>Ilustración 136 - Manual de usuario - Interfaz de configuración (2)</i>	105
<i>Ilustración 137 - Manual de usuario - Interfaz de configuración (3)</i>	105
<i>Ilustración 138 - Manual de usuario - Interfaz de configuración (4)</i>	106

<i>Ilustración 139 - Menú - Opción "Mundo"</i>	106
<i>Ilustración 140 - Menú – Opción “Elementos”</i>	107
<i>Ilustración 141 - Menú - Opción "Restricciones"</i>	108
<i>Ilustración 142 - Menú - Opción "Ciclos"</i>	109
<i>Ilustración 143 - Menú - Opción "Acciones"</i>	110
<i>Ilustración 144 - Menú - Opción "Rangos"</i>	112
<i>Ilustración 145 - Menú – Opción “Comportamiento”</i>	113
<i>Ilustración 146 - Manual de usuario - Interfaz de visualización (1)</i>	114
<i>Ilustración 147 - Manual de usuario - Interfaz de visualización (2)</i>	114
<i>Ilustración 148 - Manual de usuario - Interfaz de visualización (3)</i>	115

I. Memoria

1. Descripción del proyecto

1.1. Introducción

Se puede definir una simulación como un modelo que imita ciertos aspectos de la realidad. Esto permite trabajar en condiciones similares a las reales, pero con variables controladas y en un entorno que se asemeja al real pero que está creado o acondicionado artificialmente. La idea es que la simulación permita comprobar el comportamiento de una persona, de un objeto o de un sistema en ciertos contextos que, si bien no son idénticos a los reales, ofrecen el mayor parecido posible. De esta forma, es posible corregir fallos antes de que la experiencia se concrete en el plano de lo real.

Por ello ha surgido la idea de este proyecto, para poder anticiparnos a lo que va a suceder en la realidad y poder mejorar nuestros sistemas, nuestras predicciones o, simplemente, “jugar” con la realidad.

El juego de la vida, de John H. Conway, es un claro ejemplo. Se trata de un autómata celular, en un principio, equivalente a una máquina universal de Turing. Esto es, todo lo que se puede computar algorítmicamente se puede computar en el juego de la vida. Se considera que el juego de la vida es un buen ejemplo de emergencia y autoorganización. Es de gran interés para científicos, matemáticos y economistas observar cómo patrones complejos pueden provenir de la implementación de reglas muy sencillas.

El juego de la vida se considera un juego de cero jugadores, debido a que su evolución está determinada por el estado inicial, sin la necesidad de ningún tipo de entrada de datos posterior.

Otro ejemplo similar, pero adaptado a una necesidad, es “A simulation in Emoji”. Una mala temporada de sequías, escasez de agua y una gran cantidad de incendios forestales en California hizo que surgiera esta idea, para poder comprender dicha situación. Esto ayudó a los bomberos forestales a hacer frente a los incendios de una forma más eficaz y en tiempo real, además de a los científicos a comprender mejor nuestro mundo. La filosofía de esta idea, al igual que el juego de la vida, es que los sistemas complejos surgen de reglas simples y que todos tienen ciertas similitudes.

1.2. Objetivos

El objetivo principal de este proyecto es el desarrollo de un entorno de simulación que se pueda ejecutar desde un navegador web, en el cual se podrá comprobar el comportamiento de los individuos en el sistema. Cabe destacar que no se trata de un comportamiento poblacional, sino que será a nivel de individuo.

Para ello, se modelarán dos sistemas, en los cuáles se mostrará la ejecución de cada uno, incluyendo gráficas y datos relevantes para la comprensión de lo ocurrido y poder realizar un estudio en base a los datos introducidos y los datos obtenidos. En el primer sistema, el comportamiento de los individuos estará configurado de forma predeterminada; en el segundo, será el usuario quién defina dicho comportamiento. En ambos sistemas se incorporarán los siguientes elementos: depredadores, presas, alimentos para las presas, elementos que permitirán la ocultación de las presas y elementos estáticos del entorno.

1.3. Descripción general de la aplicación

Cuando el usuario accede a la aplicación, se encontrará con una pantalla donde podrá escoger entre dos sistemas a parametrizar, así como el idioma de la aplicación. La principal diferencia entre los dos sistemas es la siguiente: en el primero, el comportamiento de los elementos está definido por el propio sistema; en cambio, en el segundo sistema, el usuario se encargará de decidir cómo se van a comportar los elementos dinámicos del sistema.

Una vez elegido el sistema, podrá seleccionar entre dos opciones: en la primera, cargará un fichero de configuración que se obtiene una vez ejecutada la aplicación con unos determinados parámetros; en la segunda, introducirá manualmente los parámetros de configuración iniciales. Después, procederá a ejecutar la aplicación con dichos parámetros y aparecerá una pantalla de carga. Una vez finalizada la ejecución, se mostrarán los resultados que contendrán un modelado del mundo generado, en el que el usuario podrá realizar un desplazamiento temporal para comprobar su evolución, tanto de forma automática como manual. Se incluirán además gráficas relativas a la ejecución del mundo para poder analizar y contrastar datos, además de la posibilidad de descargar diferentes archivos, como la configuración inicial o el registro de acontecimientos.

2. Descripción técnica

2.1. Metodología

Para el desarrollo de este proyecto se ha escogido una metodología iterativa e incremental debido a la naturaleza del mismo, siendo esta metodología la que mejor se adapta. El hecho de que se trate de un proyecto con una base investigadora, donde van a surgir cambios constantemente a lo largo del análisis, diseño y desarrollo, ha sido clave en la decisión de esta metodología.

Se trata de un proceso de desarrollo de software en el que se planifica un proyecto en distintos bloques temporales, denominado iteraciones, de forma que el sistema se desarrolla poco a poco y con un continuo “feedback” por parte del usuario. En cada iteración se repite un proceso similar para proporcionar un resultado completo sobre el producto final, de manera que el cliente pueda obtener los beneficios del proyecto de forma incremental.

Se debe distinguir entre el ciclo de vida incremental y el ciclo de vida iterativo. En el primero, se realizar un desarrollo por partes del producto software, para después integrarlas a medida que se completan. Se trata de agregar cada vez más funcionalidad al sistema. En el segundo, cada iteración revisa y mejora el producto.

Esta simbiosis permite la consecución de una versión más estable del software, de más calidad y que permite añadir más funcionalidades respecto a versiones anteriores.

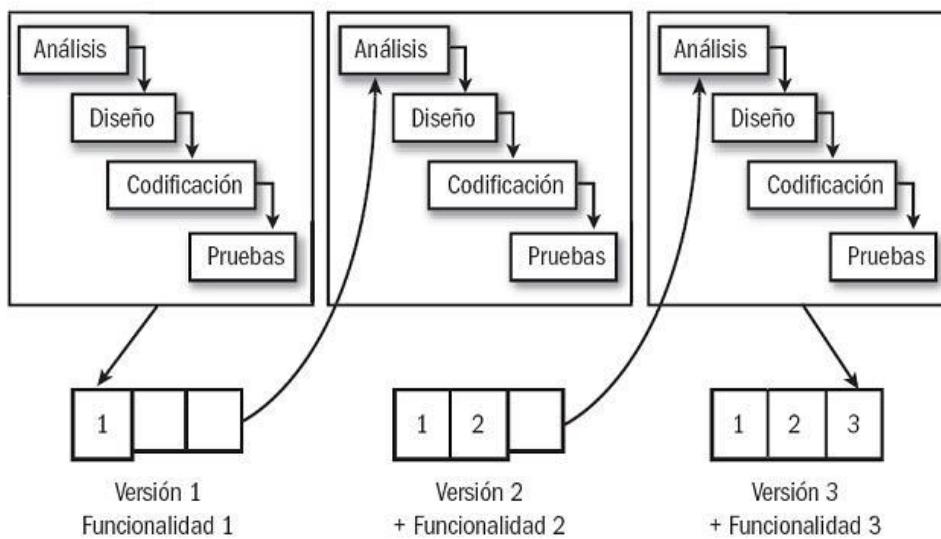


Ilustración 1 – Metodología iterativa e incremental

2.2. Arquitectura técnica

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los encargados de proveer recursos o servicios, denominados servidores, y los demandantes, llamados clientes.

En esta arquitectura, el proceso está repartido entre los clientes y los servidores. Se trata de una separación lógica, puesto que el servidor no necesariamente se ejecuta en una sola máquina ni es un único programa.

El cliente se define como el remitente de una solicitud, siendo este quién inicia las solicitudes o peticiones y espera y recibe las respuestas del servidor. Actúa como maestro en esta arquitectura.

El servidor es el receptor de la solicitud enviada por el cliente, por lo que desempeña un papel pasivo en la comunicación. Actúa como esclavo. Una vez que recibe una solicitud, la procesa y le envía la respuesta al cliente. Generalmente, acepta conexiones de un gran número de clientes.

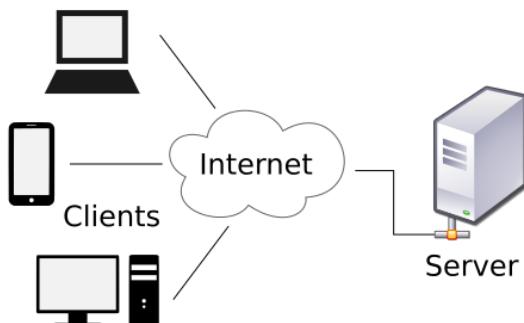


Ilustración 2 – Modelo cliente-servidor

2.3. Tecnologías y lenguajes de programación

- **HTML (HyperText Markup Language)**

Se trata de un lenguaje de marcado para la elaboración de páginas web. Es el elemento de construcción más básico y se usa para crear y representar visualmente una página web. Determina el contenido, pero no su funcionalidad.

- **CSS (Cascading Style Sheets)**

Es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML, que pretende separar la estructura de un documento de su presentación.

- **Javascript**

Es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente del lado del cliente, implementado como parte del navegador web, lo que permite mejoras en la interfaz de usuario y la creación de páginas web dinámicas. También tiene su uso del lado del servidor y en aplicaciones externas a la web, como por ejemplo las aplicaciones de escritorio.

- **PHP (Hypertext PreProcessor)**

Es un lenguaje de código abierto especialmente adecuado para el desarrollo web. El código es ejecutado en el servidor, generando HTML y enviándolo al cliente, que recibe el resultado de ejecutar el script.

- **Bootstrap**

Es un framework o conjunto de herramientas de código abierto para el diseño de sitios y aplicaciones web.

- **Apache**

Es un servidor web HTTP de código abierto que implementa el protocolo HTTP/1.1, cuya función es transferir documentos de hipertexto y comunicar un cliente con el servidor.

2.4. Herramientas

- **Microsoft Word**

Es una aplicación informática orientada al procesamiento de textos, creada por Microsoft e integrada en el paquete ofimático Microsoft Office.

- **Sublime Text 3**

Es un editor de texto y de código fuente, soportando múltiples lenguajes de programación.

- **Google Chrome**

Es una navegador web desarrollado por Google y compilado con base en varios componentes e infraestructuras de desarrollo de aplicaciones.

- **XAMPP**

Es un servidor que consiste principalmente en el sistema de gestión de bases de datos MySQL (en las últimas versiones, MariaDB), el servidor web Apache y los intérpretes para los lenguajes de script PHP y Perl.

- **Microsoft Project**

Es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, la asignación de recursos a tareas, dar seguimiento al progreso, administrar el presupuesto y analizar las cargas de trabajo.

- **Visual Paradigm**

Es una herramienta para el desarrollo de aplicaciones utilizando el modelado UML, que permite la elaboración de informes.

- **Adobe Photoshop CC**

Es una herramienta de edición de imágenes y diseño. Crea y mejora fotografías, diseños, ilustraciones en 3D, vídeos, entre otros.

3. Proceso de desarrollo y planificación

3.1. Descripción del proceso de desarrollo

Para llevar a cabo la ejecución de este proyecto, se ha distribuido el trabajo en las siguientes fases de desarrollo:

- **Documentación inicial**

En esta etapa inicial se realizará la documentación inicial, en la que se establece la definición del proyecto, los requisitos básicos de la aplicación así como la planificación de su desarrollo.

- **Análisis**

Se realizará un estudio de investigación acerca del proyecto a realizar, contemplando las distintas posibilidades de desarrollo, además de la familiarización con las tecnologías y herramientas a utilizar durante todo el desarrollo del mismo.

Durante dicho estudio se realizarán distintas implementaciones de la base de la aplicación para poder determinar cuál de ellas es la más óptima en cuanto a viabilidad y rendimiento.

- **Diseño**

Durante esta etapa se realizará un boceto del conjunto de páginas que formarán la aplicación web, formado por las interfaces de configuración y visualización del resultado obtenido.

- **Implementación**

A lo largo de esta etapa se realizará la implementación de la aplicación, además de las pruebas realizadas en cada etapa necesarias para la comprobación de su correcto funcionamiento.

Se tratará de un proceso cíclico e incremental, que se dividirá en cuatro iteraciones bien diferenciadas. Durante las tres primeras se repetirá el mismo proceso para cada una de ellas: creación del mundo, gestión de los elementos y gestión de las acciones que pueden realizar. En cambio, en la cuarta, se implementará la interfaz de usuario junto con las gráficas y los archivos de log.

- **Pruebas**

En esta última etapa se realizarán las pruebas globales de la aplicación, para comprobar su correcto funcionamiento.

- **Documentación**

Cabe destacar que la documentación de este proyecto no es una etapa como tal, sino que se realizará a lo largo de todo el proyecto durante las etapas mencionadas anteriormente.

3.2. Planificación y estimación temporal

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	« Documentación inicial	10 días	jue 01/10/15	mié 14/10/15	
2	Redacción anteproyecto	10 días	jue 01/10/15	mié 14/10/15	
3	« Análisis	40 días	jue 15/10/15	mié 09/12/15	
4	Tecnologías a utilizar	10 días	jue 15/10/15	mié 28/10/15	2
5	Modelos existentes	10 días	jue 29/10/15	mié 11/11/15	4
6	Sistema a modelar y estrategia de desarrollo	20 días	jue 12/11/15	mié 09/12/15	5
7	« Diseño	5 días	jue 10/12/15	mié 16/12/15	
8	Diseño: interfaz de usuario	5 días	jue 10/12/15	mié 16/12/15	6
9	« Implementación	85 días	jue 17/12/15	mié 11/05/16	
10	« Iteración 1	20 días	jue 17/12/15	mié 27/01/16	
11	Configuración y creación del mundo	5 días	jue 17/12/15	mié 23/12/15	8
12	Gestión de elementos	5 días	jue 07/01/16	mié 13/01/16	11
13	Acciones de los elementos	10 días	jue 14/01/16	mié 27/01/16	12
14	« Iteración 2	20 días	jue 28/01/16	mié 02/03/16	
15	Configuración y creación del mundo	5 días	jue 28/01/16	mié 03/02/16	13
16	Gestión de elementos	5 días	jue 11/02/16	mié 17/02/16	15
17	Acciones de los elementos	10 días	jue 18/02/16	mié 02/03/16	16
18	« Iteración 3	20 días	jue 03/03/16	mié 06/04/16	
19	Configuración y creación del mundo	5 días	jue 03/03/16	mié 09/03/16	17
20	Gestión de elementos	5 días	jue 10/03/16	mié 16/03/16	19
21	Acciones de los elementos	10 días	jue 17/03/16	mié 06/04/16	20
22	« Iteración 4	25 días	jue 07/04/16	mié 11/05/16	
23	Interfaz de usuario	10 días	jue 07/04/16	mié 20/04/16	21
24	Configuración de gráficas y archivos de log	15 días	jue 21/04/16	mié 11/05/16	23
25	« Pruebas	10 días	jue 12/05/16	mié 25/05/16	
26	Pruebas	10 días	jue 12/05/16	mié 25/05/16	24

Ilustración 3 – Planificación y estimación temporal

Excepciones Semanas laborales			
	Nombre	Comienzo	Fin
1	Navidad	24/12/2015	06/01/2016
2	Carnaval	04/02/2016	10/02/2016
3	Semana Santa	21/03/2016	27/03/2016

Ilustración 4 – Excepciones en el calendario

Nota: las jornadas de trabajo son de dos horas.

3.2.1. Diagrama de Gantt

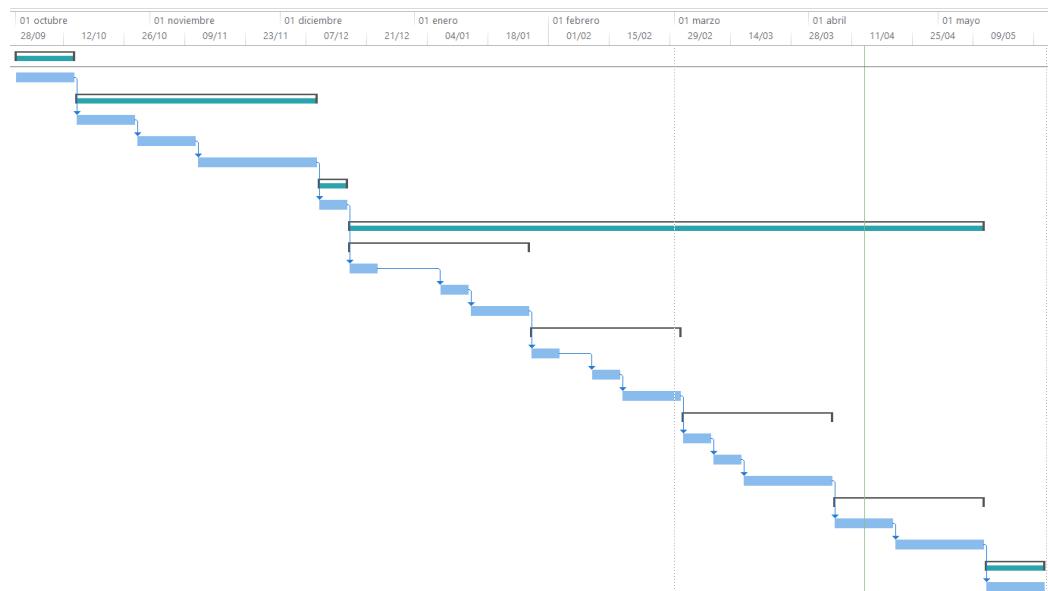


Ilustración 5 – Diagrama de Gantt

3.3. Estimación de costes

3.3.1. Costes de hardware

Se calcula al sumar el coste de los componentes hardware necesarios para la realización del proyecto. Se ha tenido en cuenta el precio del hardware en el momento de iniciar el proyecto.

Componente	Precio total	Amortización (años)	Precio final
PC	999 €	6	111 €

Ilustración 6 – Coste de hardware

3.3.2. Coste de software

Se calcula al sumar el coste de todas las licencias de Software de terceros requeridas para el desarrollo del proyecto. El precio de estas licencias puede variar durante el desarrollo del mismo, por lo que la estimación se realiza en base al coste de las licencias en el momento de la redacción de este documento.

Licencia	Precio / mes	Total
Office 365	7,00 €	56,00 €
Visual Paradigm Standard	31,07 €	248,56 €
Sublime Text 3		62,13 €
Adobe Photoshop CC	12,09 €	96,72 €
		463,41 €

Ilustración 7 – Coste de software

3.3.3. Coste de personal

Está calculado teniendo en cuenta el tiempo de desarrollo (en horas) estimado para un solo desarrollador.

Personal	Horas de trabajo	€ / hora	Total
Ingeniero de software	300	20 €	6.000 €

Ilustración 8 – Coste de personal

3.3.4. Coste total

Se calcula al sumar todos los costes anteriores.

Coste total
6.574,41 €

Ilustración 9 – Coste total

3.4. Tiempos de ejecución y coste

En los siguientes gráficos se muestran los tiempos de ejecución reales del proyecto:

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	« Documentación inicial	10 días	jue 01/10/15	mié 14/10/15	
2	Redacción anteproyecto	10 días	jue 01/10/15	mié 14/10/15	
3	« Análisis	40 días	jue 15/10/15	mié 09/12/15	
4	Tecnologías a utilizar	3 días	jue 15/10/15	lun 19/10/15	2
5	Modelos existentes	11 días	mar 20/10/15	mar 03/11/15	4
6	Sistema a modelar y estrategia de desarrollo	26 días	mié 04/11/15	mié 09/12/15	5
7	« Diseño	3 días	jue 10/12/15	lun 14/12/15	
8	Diseño: interfaz de usuario	3 días	jue 10/12/15	lun 14/12/15	6
9	« Implementación	87 días	mar 15/12/15	mié 11/05/16	
10	« Iteración 2	22 días	mar 15/12/15	mié 27/01/16	
11	Configuración y creación del mundo	8 días	mar 15/12/15	jue 07/01/16	8
12	Gestión de elementos	6 días	vie 08/01/16	vie 15/01/16	11
13	Acciones de los elementos	8 días	lun 18/01/16	mié 27/01/16	12
14	« Iteración 3	16 días	jue 28/01/16	jue 25/02/16	
15	Configuración y creación del mundo	3 días	jue 28/01/16	lun 01/02/16	13
16	Gestión de elementos	7 días	mar 02/02/16	mié 17/02/16	15
17	Acciones de los elementos	6 días	jue 18/02/16	jue 25/02/16	16
18	« Iteración 4	21 días	vie 26/02/16	vie 01/04/16	
19	Configuración y creación del mundo	6 días	vie 26/02/16	vie 04/03/16	17
20	Gestión de elementos	7 días	lun 07/03/16	mar 15/03/16	19
21	Acciones de los elementos	8 días	mié 16/03/16	vie 01/04/16	20
22	« Iteración 5	28 días	lun 04/04/16	mié 11/05/16	
23	Configuración de gráficas y archivos de log	18 días	lun 04/04/16	mié 27/04/16	21
24	Multilenguaje	5 días	jue 28/04/16	mié 04/05/16	23
25	Interfaz de usuario	5 días	jue 05/05/16	mié 11/05/16	24
26	« Pruebas	3 días	jue 12/05/16	lun 16/05/16	
27	Pruebas	3 días	jue 12/05/16	lun 16/05/16	25
28	« Documentación final	4 días	mar 17/05/16	vie 20/05/16	
29	Manual de usuario y redacción de la documentación	4 días	mar 17/05/16	vie 20/05/16	27

Ilustración 10 - Tiempos de ejecución real

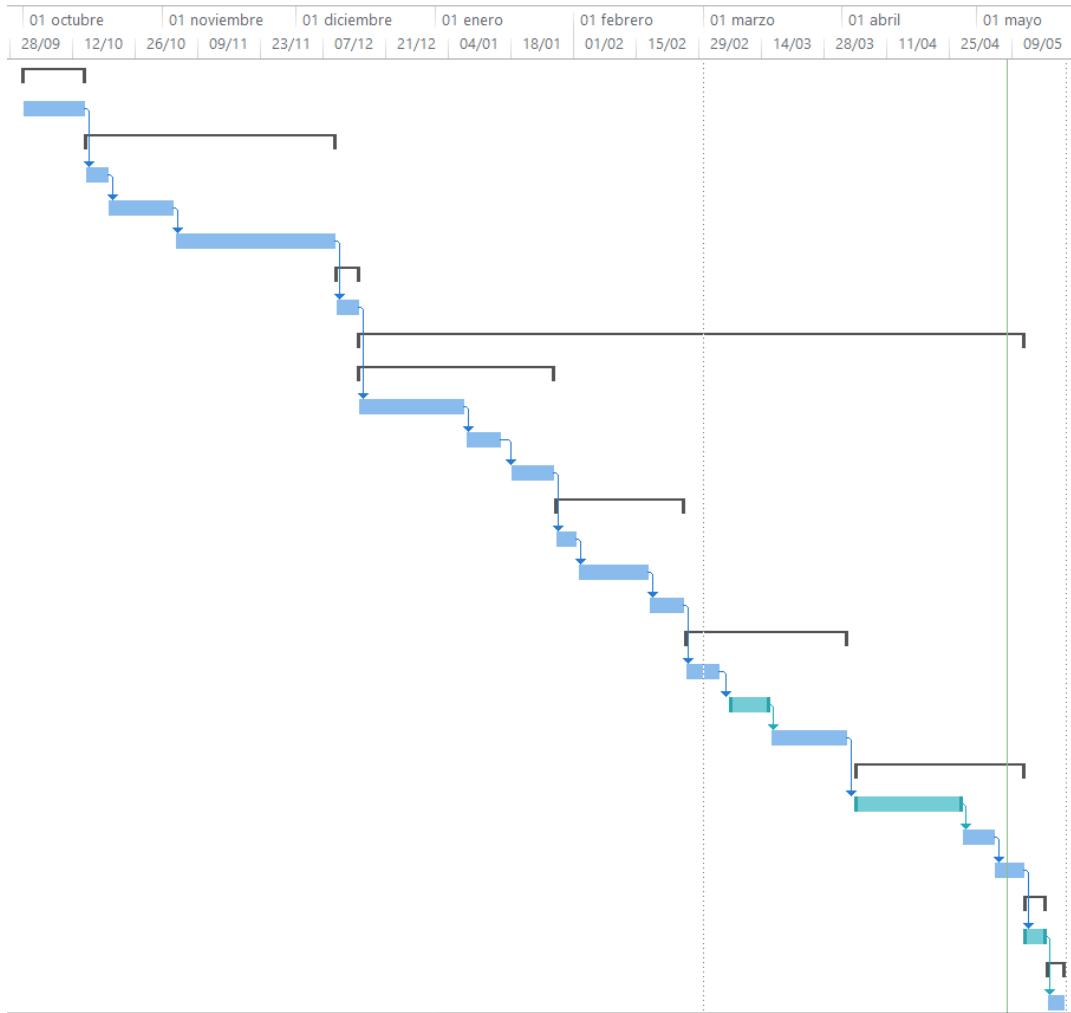


Ilustración 11 - Diagrama de Gantt - Ejecución real

Realizando una comparación con los tiempos planificados, se han producido algunos cambios, bien porque se ha necesitado más o menos tiempo para la realización de algunas actividades o bien porque se han añadido otras nuevas.

En el caso de las nuevas, se ha añadido la actividad “Multilenguaje”, ya que se ha implementado la opción multilingüe en la aplicación, tanto castellano, como gallego, como inglés. Además, se ha necesitado una actividad a mayores para terminar de redactar la documentación, junto con el manual de usuario. Es cierto que la documentación relativa a cada actividad se ha ido realizando dentro de la misma, pero se ha necesitado tiempo para darle un correcto formato y terminar algunos aspectos que faltaban por hacer.

En cuanto a las iteraciones, se ha hecho una pequeña modificación: durante la fase de investigación, puesto que se ha desarrollado la base de la aplicación, se considera este desarrollo como primera iteración. Con lo cual, en la implementación se comienza en la segunda iteración, hasta un total de cinco.

Cabe mencionar que, a pesar de que cada jornada de trabajo se había establecido en dos horas, en la práctica no se ha podido cumplir debido a la falta de disponibilidad del personal, por lo que en algunas jornadas se han incrementado las horas y en otras se han reducido. A pesar de estos pequeños desvíos, se ha podido finalizar el proyecto 5 días antes de lo previsto.

3.5. Documentación entregada

3.5.1. Memoria

Documento que contiene información relativa a la identificación del proyecto, visión general del sistema, objetivos, descripción del sistema, tecnologías utilizadas, planificación, incluyendo las desviaciones o problemas surgidos durante la elaboración del mismo, así como la bibliografía.

3.5.2. Manual técnico

Incluye los documentos obtenidos de cada una de las fases de desarrollo: planificación, especificación, diagramas de diseño y pruebas realizadas. Esta documentación será de gran utilidad para posteriores modificaciones realizadas por personas ajenas al proyecto.

3.5.3. Manual de usuario

Documento con la información necesaria para llevar a cabo la instalación del sistema y la explicación de forma sencilla de las diferentes funcionalidades de la aplicación desarrollada con el objetivo de que cualquier persona pueda utilizarla.

3.5.4. Código fuente del proyecto

Se adjuntará el código fuente de la aplicación junto con la documentación descrita correspondientes a la aplicación, con la siguiente estructura:

- /src : directorio que contendrá el código fuente de la aplicación.
- /docs : directorio que contendrá la documentación relativa a la aplicación.
- /old_versions: directorio que contendrá el código fuente de las distintas versiones desarrolladas durante la implementación de la aplicación.

4. Problemas y soluciones

4.1. Problemas encontrados

4.1.1. Arquitectura

El principal problema que ha surgido durante el desarrollo del proyecto ha sido la elección de la arquitectura y las tecnologías a utilizar. Se ha invertido tiempo en este aspecto debido a que, en un principio, se desconocía cuál sería la mejor arquitectura y lenguajes de programación para desarrollar esta aplicación.

4.1.2. Rendimiento

En las primeras pruebas realizadas, el tiempo de ejecución obtenido para unos determinados parámetros de entrada era muy superior al tiempo esperado. De esa forma, se convertía en una aplicación que ningún usuario querría utilizar debido a su extrema lentitud.

4.1.3. Almacenamiento interno

La elección de cómo gestionar el almacenamiento de datos dentro del script ha supuesto un problema debido a la insostenible gestión del sistema escogido en un principio. La utilización constante de ‘arrays’ sin organización y su consecuente gestión (añadir, eliminar, modificar, buscar), ha provocado diversos problemas a la hora de implementar las distintas funcionalidades de la aplicación.

4.1.4. Familiarización con Javascript

Una vez definido el núcleo de aplicación y sus funcionalidades, faltaba mostrar de forma visual el funcionamiento de la misma. Para ello, Javascript era indispensable. Debido a los escasos conocimientos en la materia, surgieron algunas dificultades a la hora de enfrentarse al ‘front-end’ de la aplicación.

4.2. Soluciones

4.2.1. Arquitectura

Se ha tenido que evaluar con exactitud el rendimiento de varias arquitecturas y tecnologías para, al final, optar por la que mejor se adaptase al sistema a modelar. Esto ha formado parte del período de investigación y análisis. Se ha conseguido una notable mejoría desde la primera arquitectura definida hasta la implantada finalmente.

4.2.2. Rendimiento

Los problemas de rendimiento se han ido solucionando a lo largo de la codificación de la aplicación. Cuando se detectaba cuál podría ser el problema de esta pérdida, se solventaba modificando ciertos aspectos que no alteraban la arquitectura de la aplicación, pero que sí mejoraban con creces el rendimiento de la misma. Esta modificación se ha hecho a lo largo de todo el desarrollo debido a que han ido surgiendo mejoras respecto a ello.

4.2.3. Almacenamiento interno

Para solventar este problema, se ha hecho una investigación sobre las posibilidades de almacenamiento interno que ofrece el lenguaje de programación PHP. Una vez decidida cuál sería la mejor opción, se reestructuró dicha parte. Esto facilitó el desarrollo de las demás funcionalidades que quedaban por implementar y también afectó para bien en el rendimiento de la aplicación.

4.2.4. Familiarización con Javascript

Después de conocer los fundamentos básicos del lenguaje y de realizar algunas pruebas con los elementos que se necesitarían implementar en la aplicación, el punto de vista sobre este lenguaje ha cambiado radicalmente. No sólo a la hora de poder realizar lo planificado en cuanto a interfaz, sino que dicho aprendizaje ha permitido adquirir conocimientos muy importantes de cara a la realización de futuras aplicaciones.

5. Posibles ampliaciones

Cuando se ha tomado la decisión de la realización de este proyecto, ya se sabía que se podría convertir en un proyecto a largo plazo. Se puede considerar como una primera fase lo realizado hasta el momento porque las posibilidades que nos ofrece son mucho mayores. Lo que se ha conseguido ha sido un sistema en el cual se configura un mundo y se crean elementos dentro de él que realizan acciones en base a un comportamiento establecido. Todo esto se realiza durante la misma ejecución, sin posibilidad de modificar de forma manual lo que está ocurriendo en cada ejecución.

Como primera ampliación, se mejoraría la seguridad del sistema, puesto que permitir que un usuario introduzca código en la aplicación siempre es una acción un tanto peligrosa. Se trasladaría ese control a la parte del servidor, combinando la búsqueda de palabras reservadas con la prohibición de la ejecución de ciertas funciones en la configuración del servidor.

Como segunda ampliación, que varios usuarios puedan introducir componentes en distintos momentos de la ejecución, cada uno con su respectivo comportamiento. Esto abriría un buen abanico de posibilidades de cara a proyectos de investigación de entornos biológicos como a la posibilidad trasladar este proyecto de investigación al mundo de los videojuegos web.

Por otra parte, se podría crear una gestión de usuarios de la aplicación, con su correspondiente alta, baja, modificación y consulta, y una serie de estadísticas personales y rankings globales para mejorar la experiencia del usuario en la aplicación y poder realizar competiciones con otros usuarios.

6. Conclusiones

6.1. Tecnologías y herramientas

PHP es un lenguaje muy potente, sobre todo después de la última versión (PHP 7), que ha visto a la luz a finales de 2015, ya que ha supuesto una gran cantidad de mejoras respecto de la versión anterior (mejora en el rendimiento, el doble de rápido, mejor manejo en el control de errores, etc). En un principio se desconocía gran parte de las aplicaciones que puede tener este lenguaje y eso ha hecho que la elección de este lenguaje de programación para la realización de la aplicación haya sido un poco más difícil. Después de la experiencia, cabe destacar que ha sido una buena elección.

En cuanto a Javascript, es otro lenguaje muy potente e indispensable en el desarrollo web. Tiene una infinidad de aplicaciones que desconocía. Permite conseguir una interfaz de usuario mucho más amigable.

Lo mismo se podría decir del framework Bootstrap, hoy en día considerado casi indispensable para la creación de una página web, debido a las facilidades que da a los desarrolladores y a la estética que se consigue finalmente. Debido a la multitud de dispositivos que nos encontramos hoy en día, conseguir adaptar una página web a prácticamente cualquier dispositivo no es una tarea nada fácil. Esto es algo que se consigue con el framework Bootstrap.

6.2. Planificación y gestión

En un principio, ha sido un poco costoso realizar la planificación del proyecto debido a la escasa experiencia en planificación y gestión. A pesar de ello, se ha conseguido sacar adelante y se han obtenido buenos conocimientos para futuros proyectos.

6.3. Personal

En cuanto al aspecto personal, ha sido una buena experiencia de cara al futuro laboral, ya que se han aprendido los conocimientos necesarios para llevar a cabo la realización y gestión de un proyecto de esta magnitud.

En el ámbito que se ha realizado el proyecto, he aprendido el manejo de nuevas tecnologías web que antes no conocía y me ha permitido profundizar en los aspectos/conceptos más importantes del desarrollo web.

Como conclusión personal, ha sido un proyecto muy gratificante, a pesar de las dudas que han surgido en un principio. Además, gracias al tutor que he tenido (el cuál he elegido por voluntad propia), he mejorado mis habilidades autodidácticas, muy importantes de cara al futuro laboral para lograr grandes objetivos.

II. Manual técnico

1. Introducción

1.1. Descripción del sistema

El sistema a modelar consistirá en la simulación de un mundo dónde los elementos interactúan entre sí y con el mundo, de forma que se pueda realizar un estudio del comportamiento del mundo en base a unos parámetros establecidos previamente.

Para ello, se realizará una configuración inicial del mundo y sus elementos y, después de su ejecución, se mostrarán los datos obtenidos. El resultado será en forma de gráficas y mediante una representación espacial. Además, se podrán descargar los archivos de configuración, el mundo generado y las acciones que en él han transcurrido.

2. Documentación inicial

2.1. Redacción del anteproyecto

Como paso previo a la ejecución de un proyecto, es necesario realizar la redacción de un anteproyecto donde se definan los aspectos principales del proyecto a realizar: una breve descripción, objetivos a cumplir, proceso de desarrollo y planificación, entre otros.

A continuación se muestra el detalle de los tiempos de ejecución para esta fase:

Octubre 2015						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
				1 0h	2 Introducción (5h)	3
4	5 Introducción y objetivos (4,5h)	6 Descripción técnica (1,5h)	7 Proceso de desarrollo (1h)	8 0h	9 Planificación (3,5h)	10
11	12 Medios materiales (0,25 h)	13 0h	14 Bibliografía (0,25 h)	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

■ Documentación inicial - Redacción anteproyecto

Ilustración 12 - Tiempos de ejecución - Documentación inicial

3. Análisis

3.1. Tecnologías a utilizar

Para el desarrollo del proyecto es necesario conocer a qué tecnologías hay que enfrentarse a la hora de la implementación. Para ello se han tenido en cuenta dos aspectos: tecnologías ya conocidas previamente en las que hay un cierto dominio y tecnologías recomendadas para este tipo de proyectos, las cuáles aportarán un mayor rendimiento a la aplicación.

De lo aprendido hasta el momento, los conocimientos en lenguajes de programación como Java, C/C++ y PHP se consideran suficientes como para iniciar la implementación de este proyecto, sumando el gestor de bases de datos MySQL, como una opción para acompañar a dichos lenguajes. A esto habría que sumarle los lenguajes orientados a la programación web como son HTML, CSS y Javascript.

En cuanto a los lenguajes más recomendados para la programación, se recomienda el uso de PHP o Python, junto con HTML, CSS y Javascript.

Dado que Python es un lenguaje del cual no hay ningún conocimiento previo, se ha descartado para la implementación. En cuanto a Java y C/C++, se tiene preferencia por Java puesto que es un lenguaje que ha estado presente durante gran parte de la carrera y del cual se tiene una mayor fluidez a la hora de desarrollar una aplicación. Por tanto, la alternativa C/C++ también se descarta.

Dicho esto, como lenguaje de programación para el núcleo o ‘back-end’ de la aplicación quedarían dos alternativas: Java o PHP, junto con la posibilidad de agregar MySQL. Para la vista o ‘front-end’, se utilizará HTML, CSS y Javascript.

Por otra parte, se necesita un servidor para que la aplicación a crear pueda funcionar. En este caso, se ha elegido el servidor HTTP Apache.

A continuación se detallan los tiempos de ejecución para esta fase:

Octubre 2015						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15 0h	16 0h	17
18	19 Tecnologías a utilizar (3h)	20	21	22	23	24
25	26	27	28	29	30	31

 Análisis - Tecnologías a utilizar

Ilustración 13 - Tiempos de ejecución - Tecnología a utilizar

3.2. Modelos existentes

Cuando se decide realizar un proyecto, es conveniente investigar acerca de otros proyectos o, en este caso, aplicaciones similares al proyecto a realizar para poder contrastar datos, buscar los puntos débiles que puedan tener dichos productos o aplicaciones y poder realizar un producto final que supere con creces lo que hay actualmente en el mercado.

Una vez que la idea de este proyecto ha ido cogiendo forma, las preguntas acerca de lo que hay hecho similar hasta este momento han ido creciendo. Por tanto, se ha realizado una investigación que ha dado sus frutos en forma un juego matemático, ‘El juego de la vida’ y una aplicación web, ‘A simulation in Emojí’.

Este proyecto, una vez descubiertas estas dos vías, se ha basado en ellas para mejorar la idea inicial y conseguir un producto de mayor calidad y con mejor experiencia para el usuario. A continuación se realizará una explicación de ambos resultados de la investigación:

El juego de la vida

En el año 1970, el británico John Horton Conway ha diseñado un autómata celular llamado El juego de la vida. Desde un punto de vista teórico es equivalente a una máquina universal de Turing, esto es, todo lo que se puede computar algorítmicamente se puede computar en el juego de la vida.

Su funcionamiento es muy simple, pero a la vez puede dar lugar a una combinatoria muy compleja. Se trata de un tablero bidimensional poblado por células que están regidas por turnos, siendo el proceso de ejecución el siguiente: en cada paso se creará una nueva célula en cada casilla vacía que sea adyacente a tres células, mientras que aquellas células que sean adyacentes a cuatro o más morirán debido a la superpoblación de su hábitat. Del mismo modo, morirán aquellas que sean adyacentes a menos de dos, debido a la soledad de las mismas.

Se puede decir por tanto que, más que un juego, es un entretenimiento. Se trata de un juego de cero jugadores, donde una vez establecidas las condiciones de población celular iniciales, todo el desarrollo tiene lugar de forma autónoma y no se puede intervenir para cambiar su estado. Tampoco se puede definir una victoria o una derrota, ya que la actividad puede finalizar con la muerte de las células, se puede repetir en bucle o puede vagar por estados aleatorios de forma indefinida sin llegar a estabilizarse.

Desde su publicación ha atraído mucho interés por parte de científicos, matemáticos y economistas, entre otros, observar cómo patrones complejos pueden provenir de la implementación de reglas muy simples, ya que la vida se considera un buen ejemplo de emergencia y autoorganización.

Durante la ejecución de este ‘juego’, se van obteniendo diferentes figuras que dan lugar a múltiples combinaciones, entre las que destacan:

- Formas fijas:

Son formas estáticas que ni mueren ni crecen debido a que cada una de sus células está en contacto con otras dos o tres, con lo cual se asegura su existencia eterna.

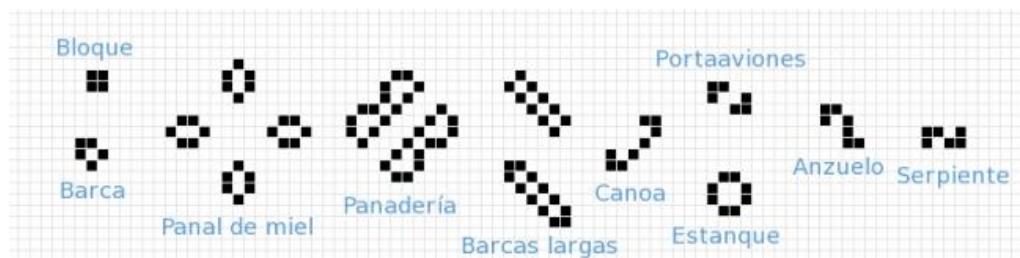


Ilustración 14 – El juego de la vida - Formas fijas

- Formas oscilantes

Son similares a las formas fijas ya que no se desplazan por el tablero, pero sí que varían su forma. Lo más habitual es que recuperen su forma original en tan sólo dos pasos.

- Formas deslizantes

Son conjuntos de células que en sus distintas generaciones evolucionan trasladándose por el tablero.

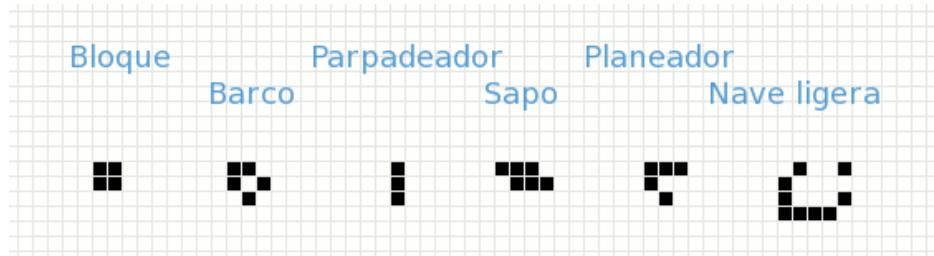


Ilustración 15 – El juego de la vida - Formas deslizantes

Es cierto que todo esto se trata únicamente de un juego, pero lo realmente interesante está en averiguar el algoritmo de transacción de estados dada una configuración inicial y, una vez conseguido, poder alcanzar nuevos retos. Además, siendo El juego de la vida un ejemplo de máquina universal de Turing, se puede utilizar para ejecutar sobre él algoritmos simples.

El concepto básico de El juego de la vida lo convierte en un sistema determinista, lo que quiere decir que ante una misma entrada de datos, siempre se va a obtener la misma salida. Lo realmente interesante para el desarrollo de este proyecto es evolucionar esta teoría añadiendo aleatoriedad al sistema, pasando de ser determinista a indeterminista. De esta forma se puede hacer que dicho modelo pueda ayudar a comprender mejor la realidad a la que representa, con lo cual podrá ayudar a estudiar modelos naturales.

'A simulation in emoji'

En una mala temporada de sequías, escasez de agua e incendios forestales, debido al caos generado, surgió la idea de entender uno de los sistemas complejos que baraja nuestro mundo, para poder entenderlo, aprender y pensar en los sistemas. Es mucho más fácil entender un sistema si se puede ver e, incluso, jugar con él.

En este punto, entra el juego el mundo de las simulaciones, para poder hacerle frente a esta ola de caos infernal en forma de incendios. Para ello, se hicieron diferentes simulaciones del mundo en ‘emoji’ que ayudaron a los bomberos a apagar los incendios y solventar el problema.

Los sistemas complejos pueden, a menudo, surgir de reglas simples. Además, todos los sistemas complejos tienen cosas en común, ya que se pueden aprender algunas lecciones básicas sobre todos los sistemas de cualquier sistema.

Inicialmente se creó un mundo donde únicamente había árboles y, en caso de generarse, fuego. Esta simulación comprobó cómo un simple foco de fuego puede llegar a quemar un bosque entero. Esto se debe a que no se puede pensar en una causa – efecto lineal, sino que se debe tratar como un ciclo de retroalimentación. De ahí que un único foco no quemé a un único árbol, sino que se pueda propagar por más árboles hasta llegar a quemar el bosque entero. Cuantos más árboles haya, más fácil es que se pueda propagar el fuego y viceversa, a menor número de árboles, menor probabilidad de propagación del fuego.

Esta sería la explicación de por qué es tan difícil el manejo de problemas complejos. Para solucionar un problema, se quiere llegar a la raíz del mismo para poder erradicarlo. Pero, ¿qué pasa cuando no se trata de un problema lineal? Llegar al objetivo es una tarea mucho más complicada.

Aplicado a los incendios forestales, a veces la solución no está únicamente en apagar el fuego. Si es cierto que lo que se pretende además de apagar el fuego es salvar la mayor cantidad de árboles posible. Pero en algunos casos se tiene que pensar de forma contraria, es decir, eliminar algunos árboles (realizando cortafuegos), para llegar a evitar futuros incendios y, de esta forma, solucionar el problema anticipándose a él. Con la simulación de este mundo se consigue realizar predicciones de cómo actuaría el fuego en el caso de que se produjese un incendio y poder actuar de la mejor manera posible.

Hay que tener en cuenta que en el caso anterior intervienen muchos más factores de los mencionados, por lo que, al enfrentarse a un problema complejo, nunca habrá una mejor solución. Por esa razón es por la que hay que pensar en los sistemas, para hacerle frente no sólo a los incendios forestales, sino a cualquier desafío global al que haya que enfrentarse.

A continuación se muestra el detalle de los tiempos de ejecución para esta fase:

Octubre 2015						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
	Juego de la vida (3h)	Juego de la vida (1,5h)	Búsqueda y análisis (4h)	Búsqueda y análisis (3,5h)	0h	0h
				A simulation in Emoji (1h)	0h	0h

 Análisis - Modelos existentes

Ilustración 16 - Tiempos de ejecución - Modelos existentes (1)

Noviembre 2015						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
1	2	3	4	5	6	7
		A simulation in emoji (2h)	A simulation in emoji (1,5h)			
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

 Análisis - Modelos existentes

Ilustración 17 - Tiempos de ejecución - Modelos existentes (2)

3.3. Sistema a modelar y estrategia de desarrollo

Dado que se trata de un proyecto en su inicio de investigación, es necesario contemplar varias alternativas para su desarrollo. En primer lugar, a partir de la idea por la cual ha surgido este proyecto, es necesario diseñar el sistema que mejor se adapte a ella.

Para ello, se ha escogido una arquitectura cliente-servidor que refleja perfectamente el sistema deseado: un cliente establece unos parámetros de configuración, se los envía al servidor, este los procesa y le devuelve el mundo generado al cliente.

Una vez definida la arquitectura, es necesario establecer una estrategia para el desarrollo e implementación de la misma. Para ello, se han realizado varias etapas de investigación para obtener la mejor implementación posible para esta arquitectura.

Primera etapa

Como se ha comentado anteriormente (3.1 Tecnología a utilizar), en un primer momento, se ha decidido implementar dicha arquitectura utilizando Java y MySQL para el ‘back-end’ y HTML, CSS, Javascript y PHP para el ‘front-end’ de la aplicación. El objetivo de esta primera etapa era conseguir hacer una pequeña aplicación que implementase este sistema para comprobar su viabilidad. Dicha aplicación consiste en lo siguiente: se tiene un tablero con una entrada, una salida y un jugador; el jugador es colocado en la entrada del tablero y tiene que buscar la salida de forma que no puede pisar más de una vez en la misma casilla. Si lo consigue, ha ganado; en caso contrario, pierde. El proceso de búsqueda de la salida se hace de forma aleatoria, por lo que el jugador carece de algún tipo de inteligencia.

El objetivo de esta primera etapa se ha cumplido, tal como se muestra a continuación:

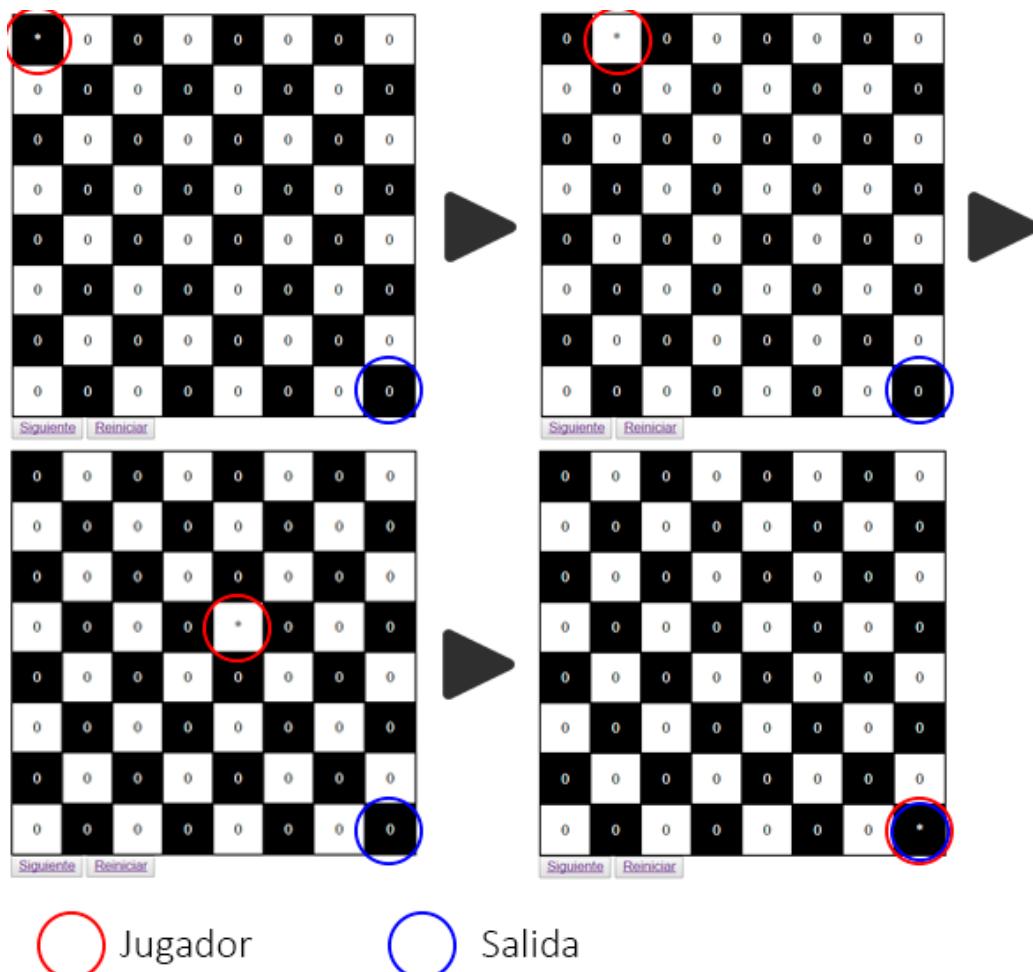


Ilustración 18 - Etapa 1

Tal y como se puede apreciar en las imágenes anteriores, el jugador se desplaza a través del tablero hasta, en este caso, encontrar la salida. El botón ‘Siguiente’ permite ver los desplazamientos que ha realizado el jugador y el botón ‘Reiniciar’ empieza la misma secuencia de nuevo. Únicamente se han mostrado algunas secuencias de la ejecución.

Una vez realizada con éxito esta etapa se han extraído dos conclusiones: la primera, que se trata de una alternativa viable para desarrollar la aplicación final y la segunda, que a pesar de conseguir unos resultados aceptables, se necesitan recursos extra para la ejecución de la aplicación. Esto es, se necesita ejecutar el código que forma el ‘back-end’ de la aplicación en una máquina virtual de Java (JVM) y almacenar los datos extraídos en un sistema gestor de bases de datos como es MySQL, lo que también obliga a instalar un conector que sea el puente de conexión entre Java y MySQL.

Segunda etapa

Durante la segunda etapa se ha decidido investigar la otra alternativa a la combinación Java & MySQL, pasando a codificar el ‘back-end’ únicamente en PHP, manteniendo el almacenamiento de datos necesario en el almacenamiento interno de PHP mediante las estructuras de datos llamadas arrays.

Primera versión

En este caso, se ha decidido implementar una aplicación muy similar a la mencionada en la etapa 1, pero con alguna variación. Para esta aplicación se han desarrollado varias versiones, cada una con algún matiz añadido respecto a la anterior, utilizando únicamente PHP como lenguaje de programación para el ‘back-end’ de la aplicación.

En la primera versión, se ha realizado la misma aplicación que en la etapa anterior pero sin ningún tipo de restricción. El jugador se coloca en la entrada del laberinto y tiene que encontrar la salida, pudiendo pisar la misma casilla las veces que deseé, como se muestra a continuación:

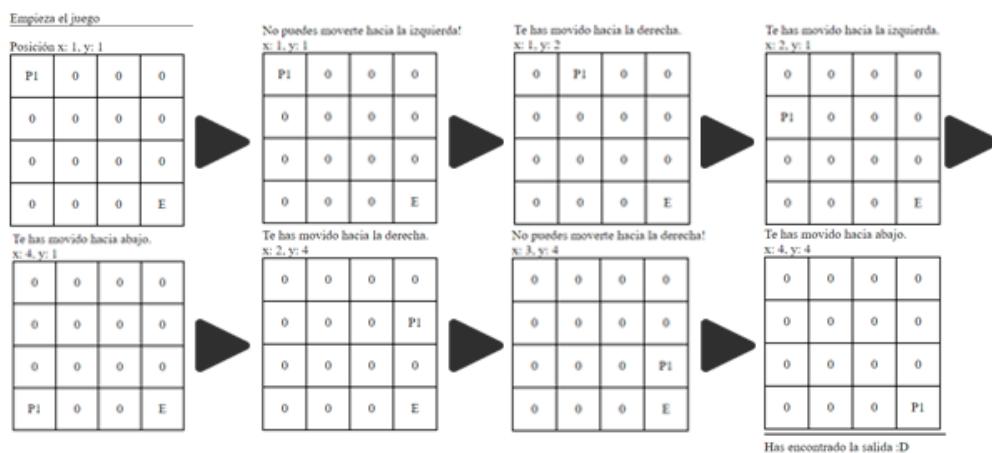


Ilustración 19 - Etapa 2 – Primera versión

Donde 'P1' representa al jugador y 'E' representa a la salida.

Como primera impresión, el rendimiento de la aplicación ha sido bueno. Además, cuenta con una ventaja respecto a lo implementado en la anterior etapa, ya que no se necesita una máquina virtual Java para ejecutar la aplicación, puesto que se ejecuta directamente en el servidor. También se ha suprimido el uso de MySQL para el almacenamiento de datos, se hace directamente utilizando la memoria interna de PHP.

Dados los buenos resultados obtenidos, se ha decidido continuar con esta vía y realizar algunas ampliaciones a esta aplicación de prueba para comprobar que los resultados obtenidos siguen siendo los esperados en cuanto a rendimiento.

Segunda versión

En la segunda versión se han añadido algunas restricciones: el jugador no puede volver a pasar por la misma casilla y se han incluido obstáculos, los cuales el jugador no puede atravesar.

A continuación se muestra el resultado obtenido de la ejecución de esta versión:

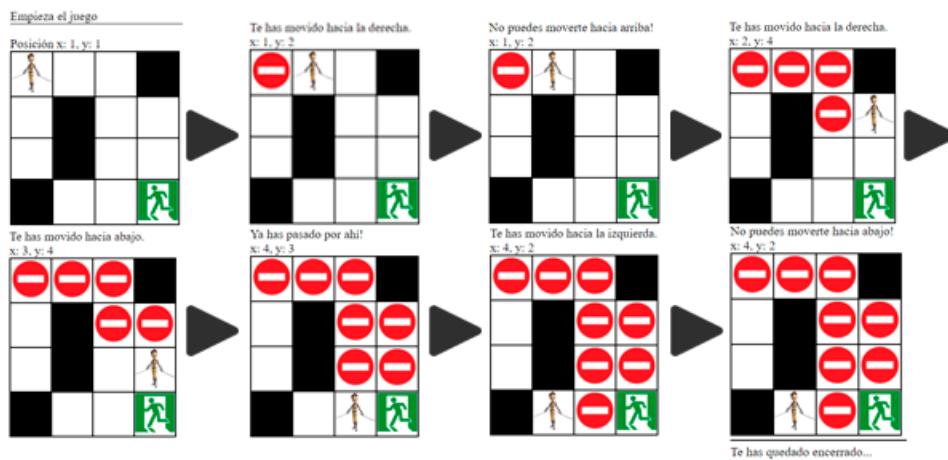


Ilustración 20 - Etapa 2 – Segunda versión

En este caso, el jugador no ha encontrado la salida porque se ha quedado encerrado: no puede volver sobre sus pasos ni puede atravesar los muros.

Tercera versión

Hecho esto, se ha decidido aumentar el número de jugadores, pero eliminando los muros. Únicamente existe la restricción de no poder pasar más de una vez por el mismo sitio.

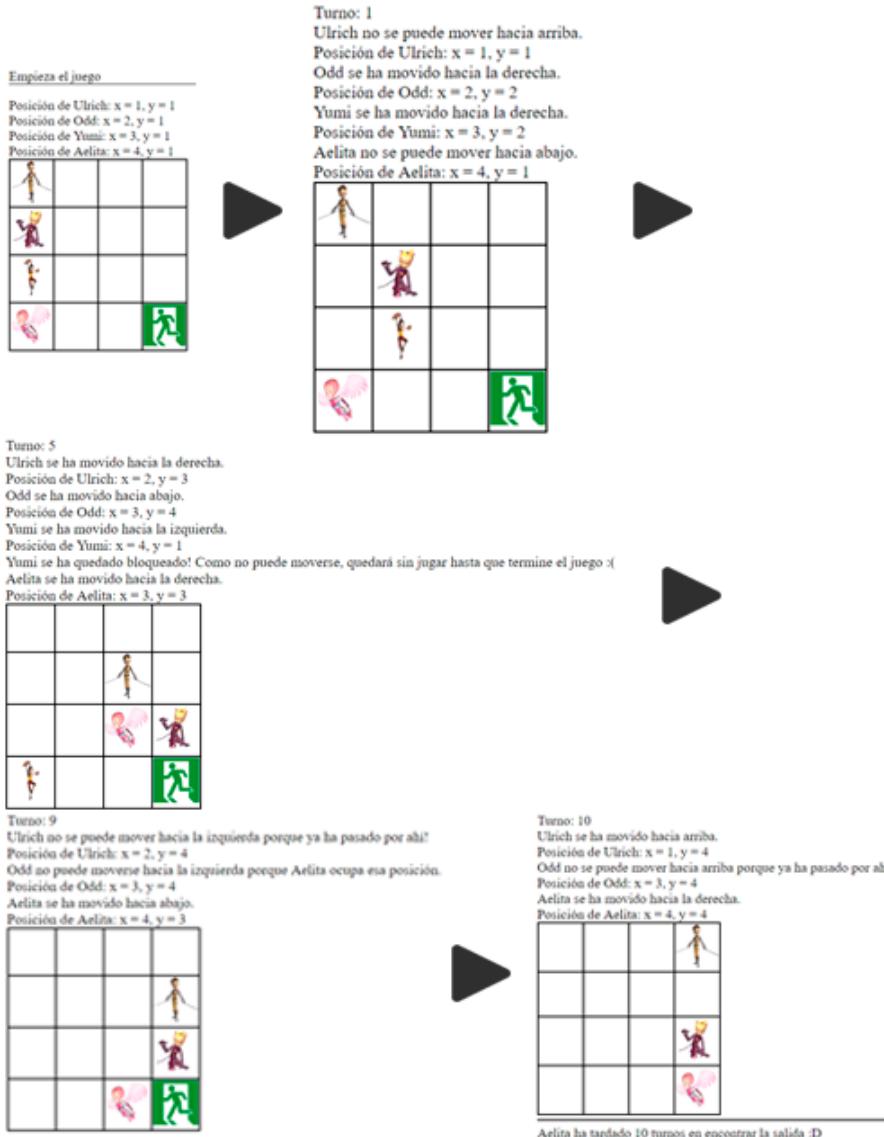


Ilustración 21 - Etapa 2 – Tercera versión

Cuarta versión

Posteriormente, se ha realizado el mismo experimento añadiendo muros al laberinto.

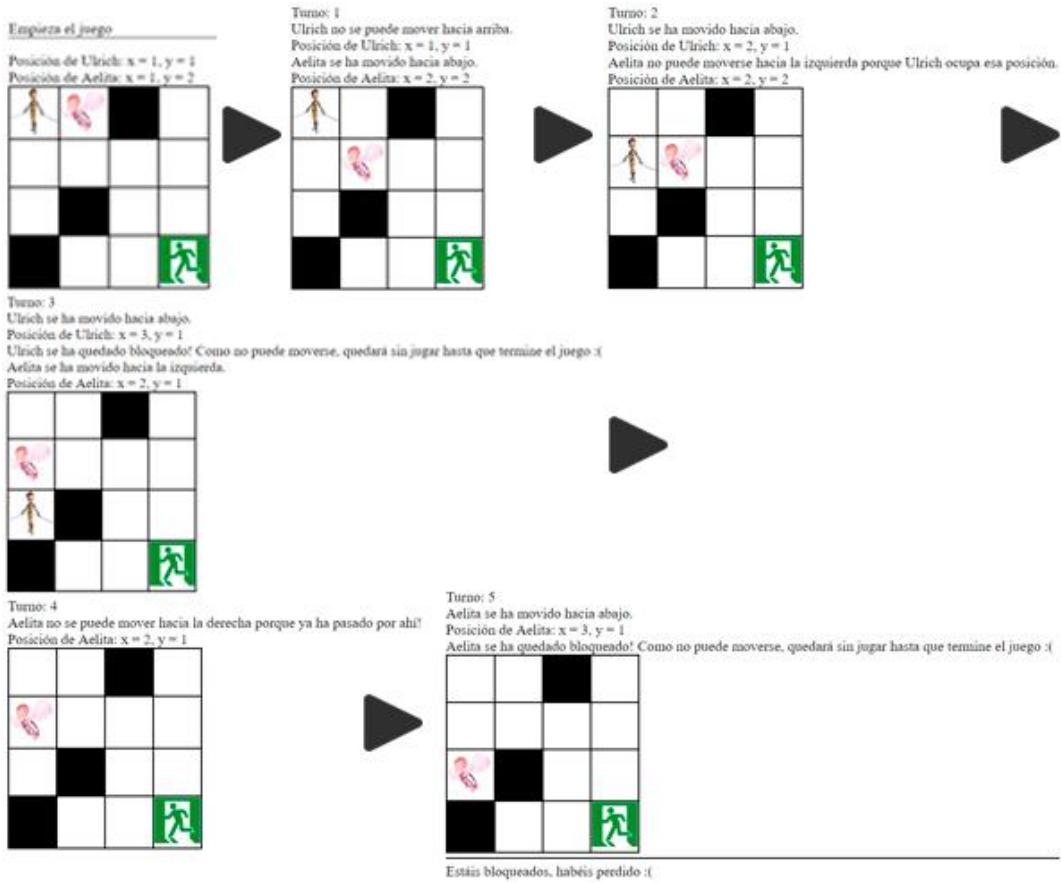


Ilustración 22 - Etapa 2 – Cuarta versión

El resultado de ambas pruebas ha resultado satisfactorio.

Una vez realizados estos experimentos, se ha llegado a la conclusión de que con la ejecución de una aplicación de esta escala, los resultados conseguidos son los esperados.

Puesto que un principio se manejaban dos alternativas para la implementación de la aplicación final, después de estas pruebas se ha optado por la segunda opción: PHP ('back-end') y HTML, CSS y Javascript ('front-end'). Ambas ofrecían un rendimiento similar, pero la segunda opción simplificaba el desarrollo de la aplicación, ya que implicaba el uso de menos tecnologías, obteniendo un resultado similar.

Inicio de la aplicación final

Llegado a este punto y con suficientes pruebas realizadas para decidir la implementación del sistema final, se ha decidido comenzar con la misma realizando una aproximación a pequeña escala de lo que sería la aplicación final.

Para ello, se han marcado las siguientes pautas a seguir para esta primera toma de contacto: diseño de la arquitectura del mundo y la interacción entre él y los elementos, implementación de una primera versión del mundo e introducción de una cantidad mínima de elementos para comprobar que la comunicación entre ellos y el mundo es correcta y que se trata de una arquitectura adecuada para el resto de la implementación.

Se han barajado dos opciones desde un primer momento:

- Una estructura de clases, donde el mundo y cada elemento es una clase, además de un script principal que se encargará de manejar todo el sistema.
- Un script central mundo y una arquitectura de clases para los elementos. En este caso, el mundo será quién maneje todo el sistema, estando los elementos incluidos en él.

Las pruebas descritas anteriormente se han realizado siguiendo la primera opción y, puesto que han sido satisfactorias, se ha decidido continuar con la misma para las siguientes implementaciones.

Primera versión

En una primera aproximación, se ha marcado como objetivo lo siguiente:

- Modelado del mundo
- Modelado de los elementos
- Interacción de los elementos con el mundo
- Creación de un registro para comprobar el correcto funcionamiento de la estructura.

Durante el modelado del mundo se establecido la gestión interna del mismo (límites, estados como el tiempo o el clima) y la gestión de los elementos (colocación dentro del mundo y eliminación, búsqueda, gestión de acciones). Dicha estructura se ha gestionado mediante un único script en el cual se han definido las funciones necesarias para dicha gestión.

En cuanto a los elementos, se han gestionado mediante una estructura de clases en la que cada elemento es una clase con atributos propios. A su vez, todo ellos derivan de una clase principal que se utiliza para identificar a cada elemento.

Como resultado de esta implementación, se ha generado lo siguiente:

- Primera prueba:

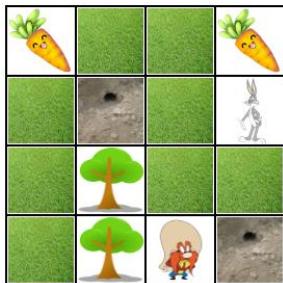


Ilustración 23 - Primera versión - Prueba 1 - Visualización

```
**** Fichero de registro ****
--Mundo
Dimensiones: 4 x 4
Mundo creado
Césped           [ n , n ]
Árbol 16         [ 3 , 1 ]
Árbol 17         [ 2 , 1 ]
Madriguera 18    [ 1 , 1 ]
Madriguera 19    [ 3 , 3 ]
Zanahoria 20    [ 0 , 0 ]
Zanahoria 21    [ 0 , 3 ]
Conejo 22        [ 1 , 3 ]
Lobo 23          [ 3 , 2 ]

--Acciones
--Turno 1
rabbit 22       Mirar      up
wolf 23          Mirar     right
wolf 23          Mover     left
--Turno 2
rabbit 22       Mirar      up
wolf 23          Mirar     down
--Turno 3
rabbit 22       Mirar      up
wolf 23          Mirar     right
wolf 23          Mover     down
```

Ilustración 24 - Primera versión - Prueba 1 - Registro

- Segunda prueba:

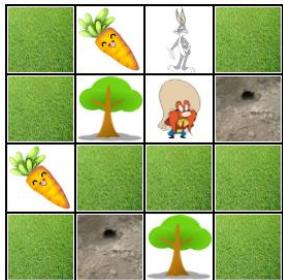


Ilustración 25 - Primera versión - Prueba 2 - Visualización

```

**** Fichero de registro ****
--Mundo
Dimensiones: 4 x 4
Mundo creado
Césped      [ n , n ]
Árbol 16    [ 1 , 1 ]
Árbol 17    [ 3 , 2 ]
Madriguera 18 [ 1 , 3 ]
Madriguera 19 [ 3 , 1 ]
Zanahoria 20 [ 2 , 0 ]
Zanahoria 21 [ 0 , 1 ]
Conejo 22   [ 0 , 2 ]
Lobo 23     [ 1 , 2 ]

--Acciones
--Turno 1
rabbit 22   Mirar      right
wolf 23     Mirar      down
--Turno 2
rabbit 22   Mirar      up
rabbit 22   Mover      right
wolf 23     Mirar      left
--Turno 3
rabbit 22   Mirar      down
wolf 23     Mirar      down

```

Ilustración 26 - Primera versión - Prueba 2 - Registro

En ambas pruebas se puede observar una recreación del mundo y un registro de acciones formado por: las dimensiones del mundo, los elementos que se han colocado en el mundo junto con su posición y las acciones que pueden realizar (en este caso, dos acciones básicas como mirar o moverse) determinados elementos durante el tiempo de ejecución, dividido en ciclos.

A pesar de que se ha diseñado un pequeño ejemplo, se ha implantado la base para todo el desarrollo de la aplicación.

Ampliación de la primera versión

Continuando con lo realizado en la primera versión, se ha añadido una nueva funcionalidad para poder realizar un seguimiento gráfico de los movimientos de cada elemento en el mundo.

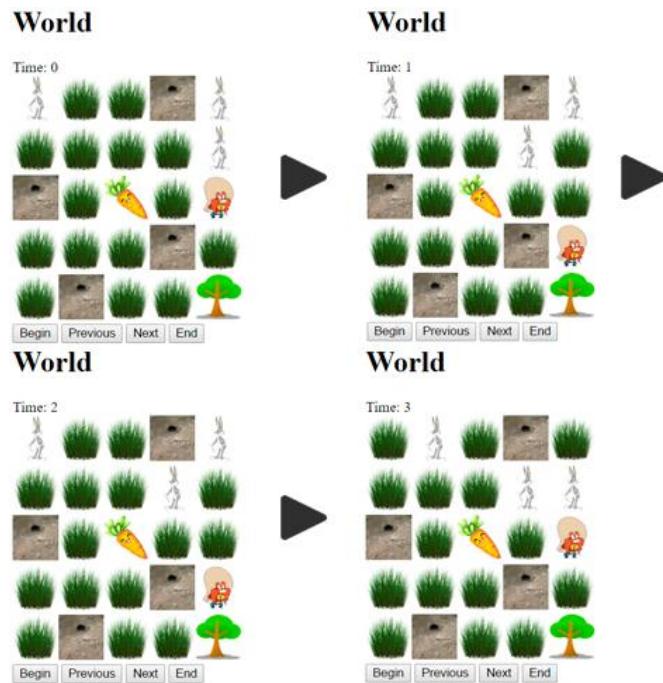


Ilustración 27 - Ampliación primera versión

Alcanzado este punto, se considera que el período de investigación ha finalizado puesto que se ha conseguido una base sólida para el desarrollo de la aplicación.

Además, esta primera versión es considerada como la primera iteración en lo que se refiere a la implementación del proyecto. Es cierto que se encuentra enmarcada en el período de investigación porque ha surgido de esa fase, pero ya se han implementado la base de la aplicación final, siendo este el motivo de su nombramiento como tal.

A continuación se muestra el detalle de los tiempos de ejecución para esta fase:

Noviembre 2015						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
1	2	3	4 Arquitectura (2h)	5 0h	6 0h	7
8	9 0h	10 Primera etapa (5h)	11 Primera etapa (3h)	12 0h	13 Primera etapa (1h)	14
15	16 Segunda etapa - Primera versión (3h)	17 Segunda etapa - Primera versión (2,5h)	18 0h	19 0h	20 0h	21
22	23 Segunda etapa - Primera versión (4,5h)	24 Segunda etapa - Segunda versión (1,5h)	25 Segunda etapa - Segunda versión (2,5h)	26 0h	27 0h	28
29	30 Segunda etapa - Tercera versión (2,5h)					

■ Análisis - Sistema a modelar y estrategia de desarrollo

Ilustración 28 - Tiempos de ejecución - Sistema a modelar y estrategia de desarrollo (1)

Diciembre 2015						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
		1 Segunda etapa - Tercera y cuarta versión (3h)	2 Segunda etapa - Cuarta versión (3h)	3 0h	4 Aplicación final - Primera versión (7h)	5
6	7 Aplicación final - Primera versión (6h)	8 Aplicación final - Primera versión (6h)	9 Aplicación final - Primera versión (5h)	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

■ Análisis - Sistema a modelar y estrategia de desarrollo

Ilustración 29 - Tiempos de ejecución - Sistema a modelar y estrategia de desarrollo (2)

3.4. Descripción de casos de uso

3.4.1. Diagrama de casos de uso

En el siguiente diagrama se muestran las interacciones del usuario con el sistema:

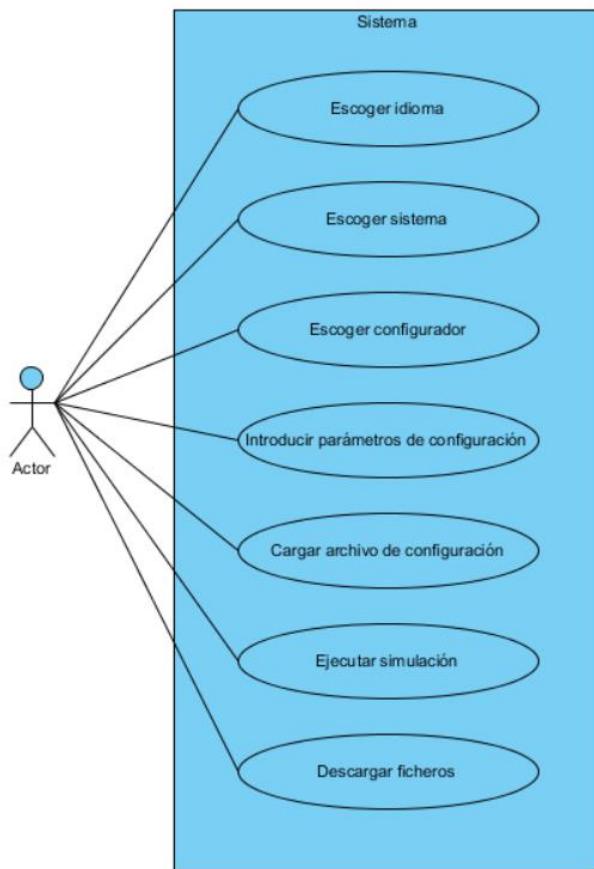


Ilustración 30 - Diagrama de casos de uso

3.4.2. Descripción detallada de casos de uso

A continuación se realiza una descripción detallada de los casos de uso, incluyendo los siguientes aspectos: breve descripción, precondiciones, post-condiciones, flujo de eventos y flujo alternativo (en el caso de que sea necesario).

- Escoger idioma

Breve descripción	El actor selecciona el idioma pulsando el botón del idioma deseado	
Precondiciones	-	
Post-condiciones	Se modifica el idioma de la aplicación	
Flujo de eventos	Actor	Sistema
	1 Selección del idioma deseado	Cambia el idioma de la aplicación
Flujo alternativo [A1]	Actor	Sistema
	1	Si el actor no selecciona un idioma, se establece el idioma por defecto

Ilustración 31 - Descripción detallada - Escoger idioma

- Escoger sistema

Breve descripción	El actor selecciona el tipo de sistema a modelar pulsando en el botón correspondiente	
Precondiciones	-	
Post-condiciones	Se escoge el sistema a modelar	
Flujo de eventos	Actor	Sistema
	1 Selecciona el sistema a modelar	
	2	Muestra la pantalla de configuración para el sistema seleccionado.

Ilustración 32 - Descripción detallada - Escoger sistema

- Escoger configurador

Breve descripción	El actor elige entre dos opciones de configuración del sistema	
Precondiciones	El sistema a modelar tiene que estar escogido	
Post-condiciones	Se muestra la pantalla de parametrización	
Flujo de eventos	Actor	Sistema
	1 Selecciona la opción de configuración	
	2	Muestra la pantalla de configuración seleccionada.

Ilustración 33 - Descripción detallada - Escoger configurador

- Introducir parámetros de configuración

Breve descripción	El actor rellena todos los campos de la interfaz de parametrización	
Precondiciones	El tipo de configuración tiene que estar escogida	
Post-condiciones	Comienza la simulación del sistema configurado	
Flujo de eventos	Actor	Sistema
	1 Rellena cada campo de la interfaz de parametrización	
	2	Comienza la simulación del sistema
Flujo alternativo [A1]	Actor	Sistema
	1	Si los datos introducidos por el actor no se corresponden con los datos esperados por el sistema, vuelve al paso 1

Ilustración 34 - Descripción detallada - Introducir parámetros de configuración

- Cargar archivo de configuración

Breve descripción	El actor elige un archivo de configuración que haya guardado previamente para cargarlo en el sistema	
Precondiciones	El tipo de configuración tiene que estar escogida	
Post-condiciones	Se accede a la pantalla de parametrización	
Flujo de eventos	Actor	Sistema
	1 Selecciona un archivo	
Flujo alternativo [A1]		Muestra la pantalla de parametrización
	Actor	Sistema
Flujo alternativo [A1]	1	Si el archivo seleccionado no se corresponde con el tipo de archivo requerido, vuelve al paso 1

Ilustración 35 - Descripción detallada - Cargar archivo de configuración

- Ejecutar simulación

Breve descripción	Se inicia la simulación del mundo configurado. Mientras la simulación se está realizando, se muestra una pantalla de carga. Una vez que finaliza la simulación, se muestran los datos recogidos de la misma	
Precondiciones	La interfaz de configuración tiene que estar completada	
Post-condiciones	Se muestran los datos recogidos de la simulación del sistema	
Flujo de eventos	Actor	Sistema
	1 Pulsa el botón para iniciar la simulación del sistema modelado	
Flujo alternativo [A1]	2	Ejecuta la simulación del sistema modelado
	3	Muestra una pantalla de carga mientras se ejecuta la simulación
Flujo alternativo [A1]	4	Muestra los datos recogidos de la simulación
	Actor	Sistema
Flujo alternativo [A1]	1	Si se ha producido algún fallo durante la ejecución, muestra una pantalla de error

Ilustración 36 - Descripción detallada - Cargar archivo de configuración

- Descargar ficheros

Breve descripción	Descarga de archivos referentes a la simulación del sistema	
Precondiciones	La simulación ha sido ejecutada	
Post-condiciones	-	
Flujo de eventos	Actor	Sistema
	1 Selecciona el archivo a descargar	
Flujo alternativo [A1]		Transmite al usuario el archivo seleccionado

Ilustración 37 - Descripción detallada - Descargar ficheros

Los tiempos de ejecución de esta fase se ven reflejados a lo largo de **5. Implementación**, ya que es documentación relativa a la implementación y se ha realizado en ese instante.

4. Diseño

4.1. Interfaz de usuario

La interfaz de usuario de la aplicación está formada por las siguientes pantallas:

- Inicio de la aplicación

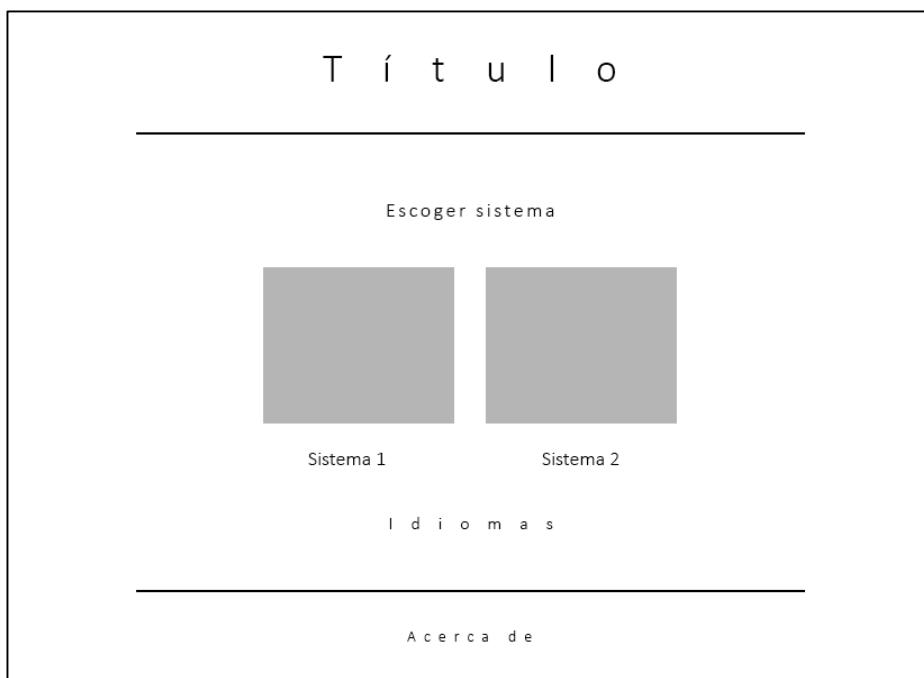


Ilustración 38 - Interfaz de usuario - Inicio

- Elección de sistema

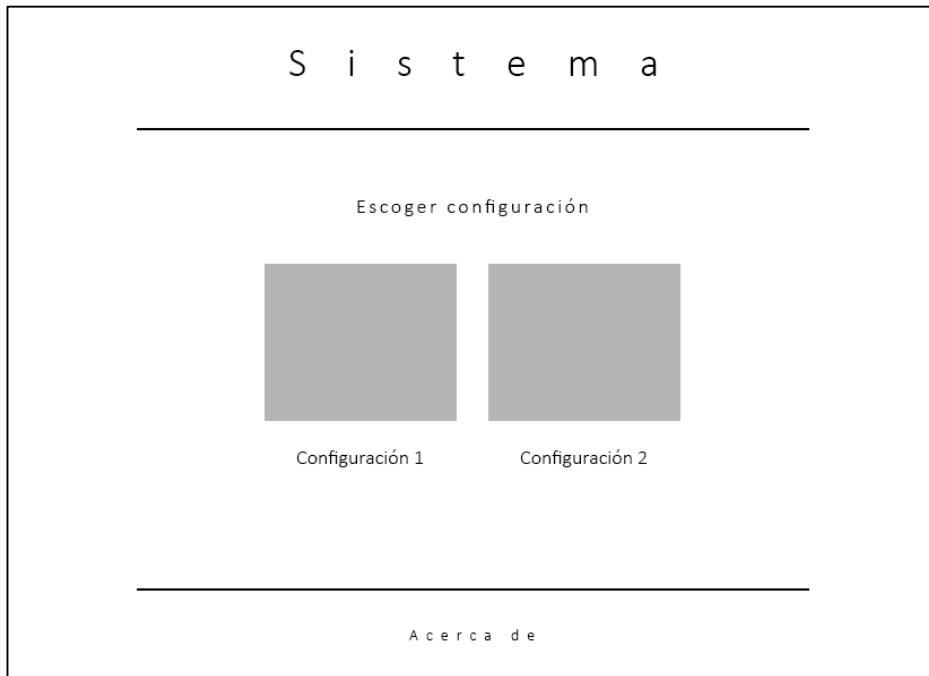


Ilustración 39 - Interfaz de usuario - Elección de sistema

- Carga de archivo de configuración

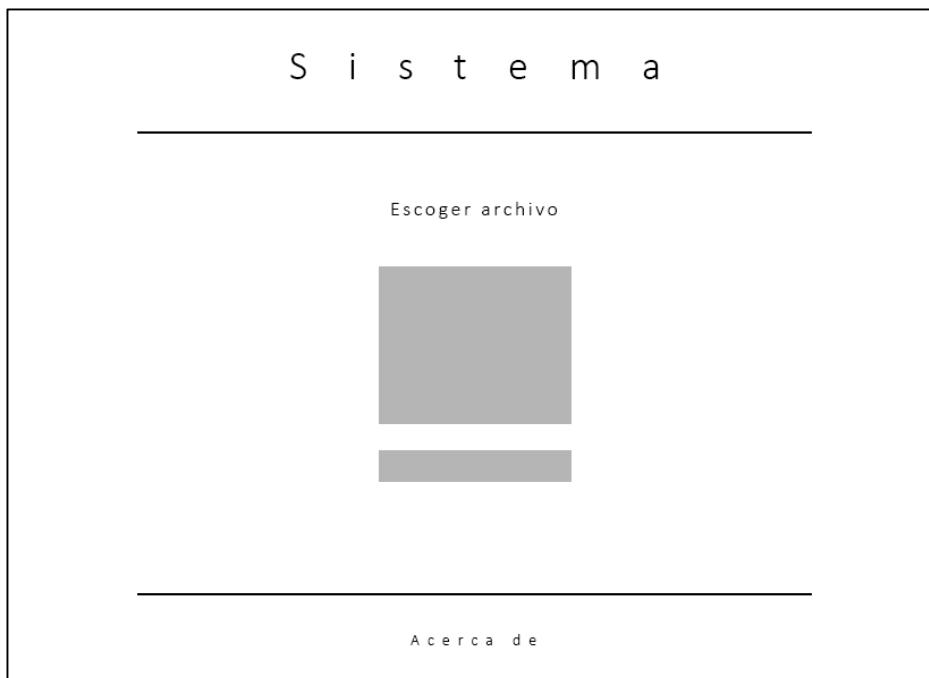


Ilustración 40 - Interfaz de usuario - Carga de archivo de configuración

- Parametrización

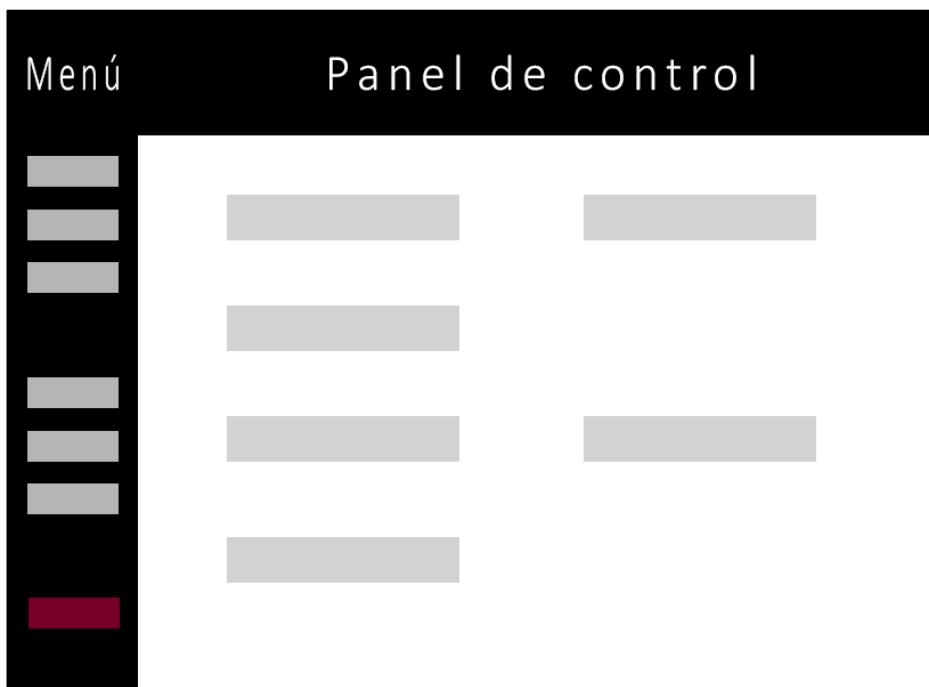


Ilustración 41 - Interfaz de usuario – Parametrización

- Visualización

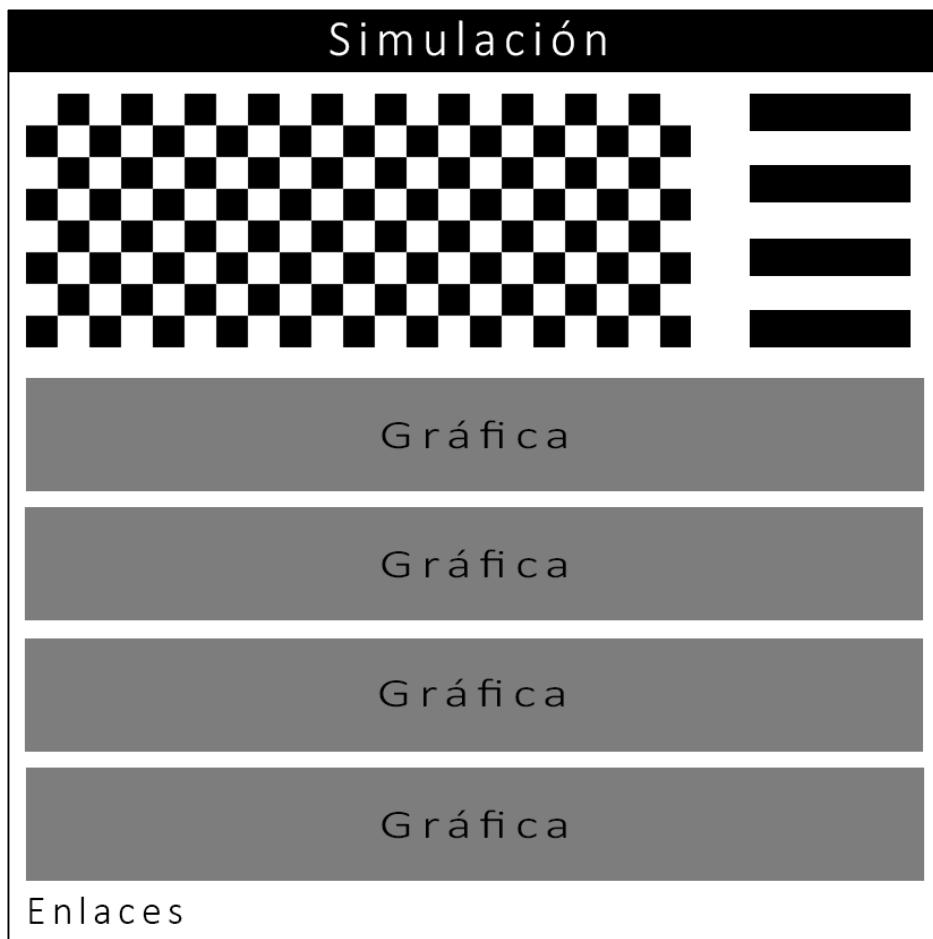


Ilustración 42 - Interfaz de usuario – Visualización

A continuación se detallan los tiempos de ejecución para esta fase:

Diciembre 2015						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Diseño - Interfaz de usuario

Ilustración 43 - Tiempos de ejecución - Diseño de la interfaz de usuario

4.2. Arquitectura de clases

En un primer momento y en base a lo concluido en la fase de análisis, se ha realizado el siguiente diseño de la arquitectura de clases de la aplicación:

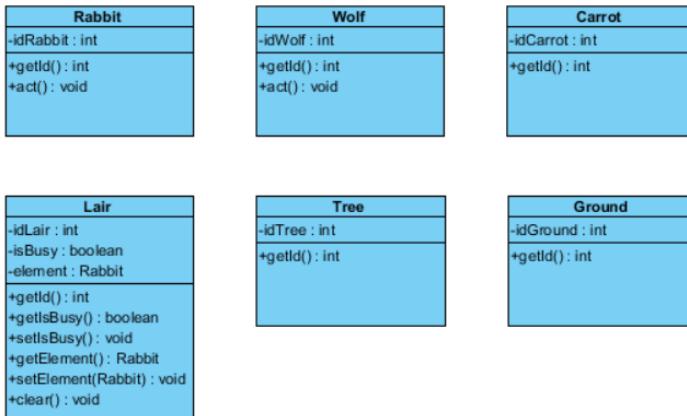


Ilustración 44 - Diagrama de clases inicial

Aunque debido a que han ido surgiendo mejoras y modificaciones en la arquitectura, el diseño final se refleja a continuación:

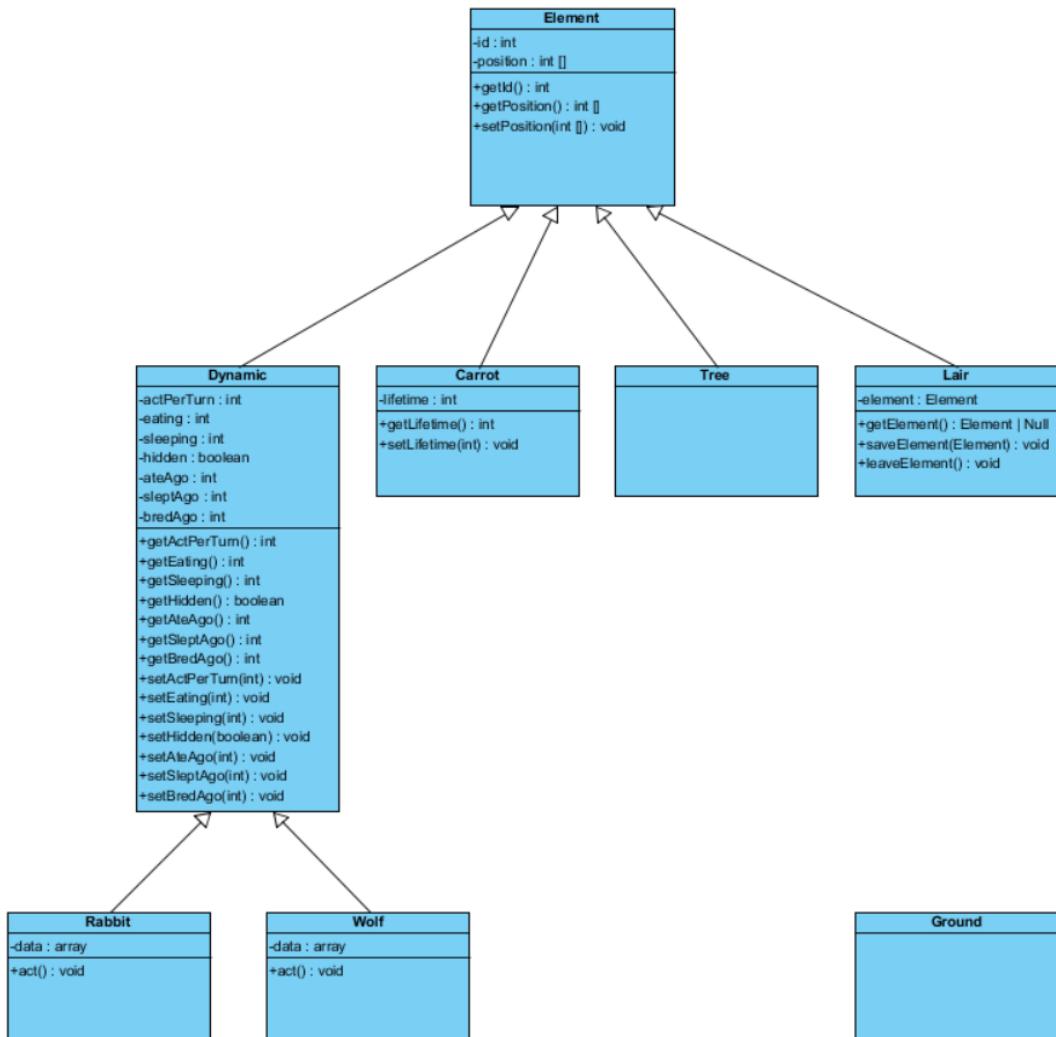


Ilustración 45 - Diagrama de clases final

Los tiempos de ejecución de esta fase se ven reflejados a lo largo de **5. Implementación**, ya que es documentación relativa a la implementación y se ha realizado en ese instante.

4.3. Arquitectura general

Finalmente, se puede definir la estructura final del sistema de la siguiente forma: un script “mundo” se encargará de gestionar todas las acciones que ocurren durante la ejecución dentro del mundo, conteniendo las clases necesarias para la gestión de los elementos.

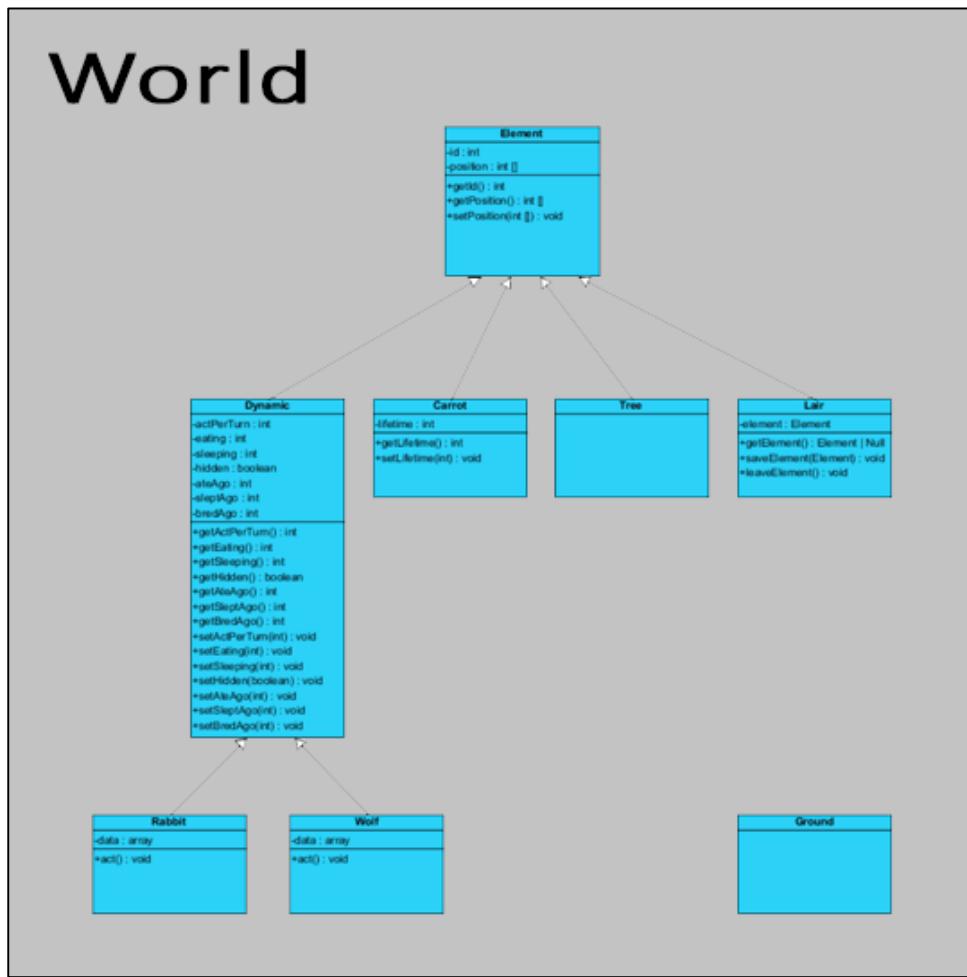


Ilustración 46 - Arquitectura general

Los tiempos de ejecución de esta fase se ven reflejados a lo largo de **5. Implementación**, ya que es documentación relativa a la implementación y se ha realizado en ese instante.

4.4. Diagrama de secuencia del sistema

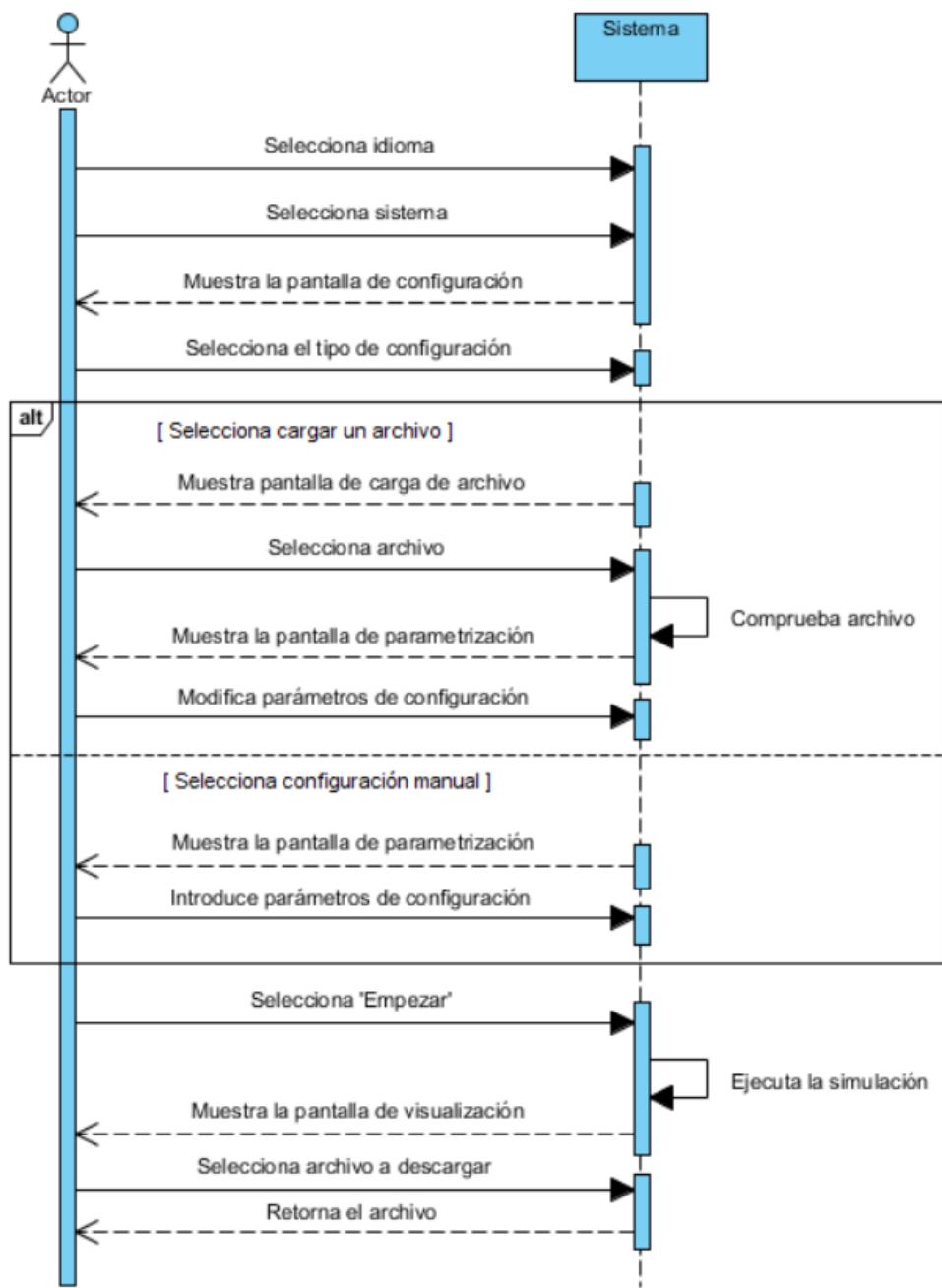


Ilustración 47 - Diagrama de secuencia del sistema

Los tiempos de ejecución de esta fase se ven reflejados a lo largo de **5. Implementación**, ya que es documentación relativa a la implementación y se ha realizado en ese instante.

5. Implementación

Para la implementación de la aplicación se ha decidido dividir el trabajo en varias iteraciones, en las cuales se van ir incrementando las funcionalidades respecto a la iteración anterior y solucionando los problemas que puedan ir surgiendo. A esto se le suma la realización de una serie de pruebas para comprobar el correcto funcionamiento de lo implementado en cada iteración.

Cabe destacar que se comienza en la iteración 2, puesto que la iteración 1 se ha realizado durante la fase de investigación.

5.1. Iteración 2

5.1.1. Configuración y creación del mundo

En cuanto a la configuración y creación del mundo, en esta segunda iteración se han realizado las siguientes tareas:

- **Definición de los parámetros iniciales del mundo.**

Se ha hecho un recuento de los parámetros necesarios para la configuración del mundo, obteniendo los siguientes:

- Ciclos de ejecución: número de ciclos durante los cuales se realizará la simulación del mundo.
- Duración del día: número de ciclos que durará un día (día + noche).
- Duración de la noche: número de ciclos que durará una noche.
- Tamaño (largo y ancho): número de casillas, tanto a lo largo como a lo ancho que compondrán el mundo.
- Tiempo atmosférico inicial: estado del tiempo atmosférico con el que se empezará la ejecución.
- Estado inicial del día (día o noche): estado que indica si al inicio de la ejecución es de día o de noche.
- Cambiar tiempo atmosférico cada x ciclos: indica cada cuántos ciclos se modificará el estado Tiempo atmosférico.

- **Definición, creación, modificación y visualización del mundo.**

Se ha definido cómo va a ser el mundo en base al tamaño establecido, así como los métodos para poder modificar el mundo (el contenido de cada posición) y para realizar un renderizado del mismo.

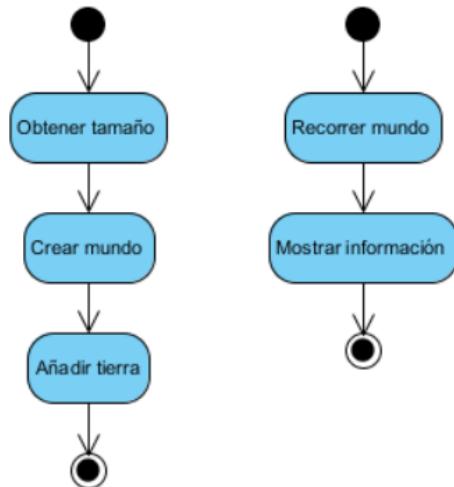


Ilustración 48 - Diagrama de actividad - Creación del mundo y visualización del mundo

Este apartado se puede ver reflejado en las siguientes funciones: `getSize()`, `createWorld()`, `getWorld()`, `writeWorld()` y `setWorld()`.

- **Configuración de los estados del mundo (tiempo, día/noche y tiempo atmosférico) y sus correspondientes cambios dentro de la ejecución del mismo.**

Dentro del mundo, las unidades de tiempo van a ser los ciclos. En cada ciclo sucederán una serie de acciones que se explicarán más adelante.

En cuanto a los estados del mundo, se van a definir dos: por una parte, el período día / noche, que irá cambiando de uno a otro cada ciertos ciclos; por otra, el tiempo atmosférico, que también variará conforme avancen los ciclos.

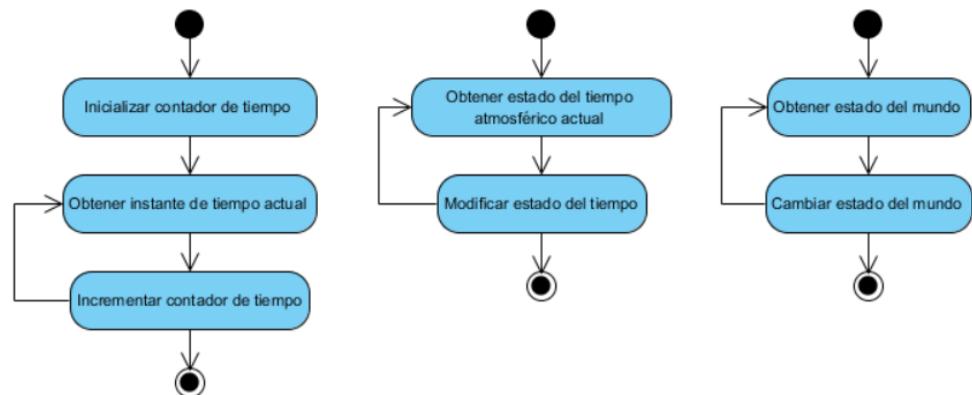


Ilustración 49 - Diagrama de actividad - Configuración de los estados del mundo

Para llevarlo a cabo, se han implementado las siguientes funciones: `getWeather()`, `setWeather()`, `isDaylight()`, `setDaylight()`, `getTime()` y `setTime()`.

- **Creación de archivos de log.**

Para realizar pruebas con mayor exactitud, se han creado una serie de logs que reflejan lo que ocurre en el mundo en cada momento. Además, estos servirán en un futuro para informar al usuario de lo ocurrido dentro del mismo.

- Mundo: se guarda el contenido de cada posición del mundo en cada ciclo de ejecución.
- Acciones: almacena los parámetros de configuración inicial del mundo, mencionados en el apartado ‘Definición de los parámetros iniciales del mundo’ de esta etapa. Además guarda los cambios en los distintos estados del mundo durante cada ciclo junto con un contador de ciclos.
- Debug: se utiliza para depurar la aplicación durante la fase de desarrollo.

La creación de los distintos archivos de log se realiza mediante las siguientes funciones: *openFileActions()*, *openFileWorld()*, *openFileDebug()* y *closeFile()*.

- **Creación de una pequeña interfaz con los parámetros de configuración que el usuario introducirá por pantalla.**

Para que la configuración sea personalizable, se ha implementado una pequeña interfaz con los parámetros necesarios para la creación del mundo, mencionados en el apartado ‘Definición de los parámetros iniciales del mundo’ de esta etapa.

The screenshot shows a user interface for configuration. At the top, there is a section labeled 'Turnos de ejecución' with a text input field containing '200'. Below this are two sections: 'Duración de un día' with a text input field containing '100' and 'Duración de la noche' with a text input field containing '20'. At the bottom, there is a 'Tamaño' section with two dropdown menus: 'Largo' and 'Ancho', each followed by a text input field containing '20'.

Ilustración 50 - Iteración 2 - Interfaz de usuario - Configuración y creación del mundo (1)

The screenshot shows a user interface for configuration. At the top, there is a section labeled 'Tiempo atmosférico inicial' with a dropdown menu showing 'soleado'. Below this are two sections: 'Estado inicial del día' with a dropdown menu showing 'día' and 'Cambiar tiempo cada (turnos)' with a text input field containing '10'.

Ilustración 51 - Iteración 2 - Interfaz de usuario - Configuración y creación del mundo (2)

A continuación se muestran los tiempos de ejecución para esta fase:

Diciembre 2015						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15 0h	16 0h	17 0h	18 Definición de los parámetros iniciales del mundo (0,5h)	19
20	21 Definición, creación, modificación y visualización (2h)	22 Configuración de los estados del mundo (1h)	23 Creación de archivos de log (1h)	24	25	26
27	28	29	30	31		

■ Implementación - Iteración 2

Ilustración 52 - Tiempos de ejecución - Iteración 2 - Configuración y creación del mundo

Enero 2016						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
					1	2
3	4	5	6	7 Pequeña interfaz de usuario (0,5h)	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

■ Implementación - Iteración 2

Ilustración 53 - Tiempos de ejecución - Iteración 2 - Configuración y creación del mundo

5.1.2. Gestión de elementos

Para la gestión de los elementos que estarán presentes en el mundo, se han realizado las siguientes tareas durante esta iteración:

- **Definición de los parámetros iniciales de los elementos.**

Además del mundo, los elementos también tienen unos parámetros de configuración, que se reflejan a continuación:

- Cantidad elementos de cada tipo: se indica cuántos elementos de cada tipo se van a colocar en el mundo en el estado inicial del mismo.
- Generación de comida para las presas: se indica cada cuántos turnos se van a generar los elementos ‘comida para las presas’ y la cantidad de los mismos.

- Generación de depredadores: se indica cada cuántos turnos se van a generar depredadores y la cantidad de los mismos.
- **Creación, modificación, eliminación y búsqueda de elementos dentro de la estructura de almacenamiento interno para los mismos.**

Se han definido los métodos necesarios para crear, eliminar y buscar elementos en el mundo, además de poder modificar los atributos de cada uno. Para gestionar los elementos, se han creado una estructura interna de almacenamiento, que se divide de la siguiente forma:

- Estáticos: contiene los elementos estáticos del mundo, es decir, aquellos que nunca cambian su posición durante la ejecución.
- Dinámicos: contiene los elementos que realizan acciones durante la ejecución, en este caso, depredadores y presas.
- Premio: contiene los alimentos para las presas

Esta división se gestiona internamente mediante el uso de arrays, uno para cada tipo de elementos.

Cada elemento se crea de la siguiente forma:

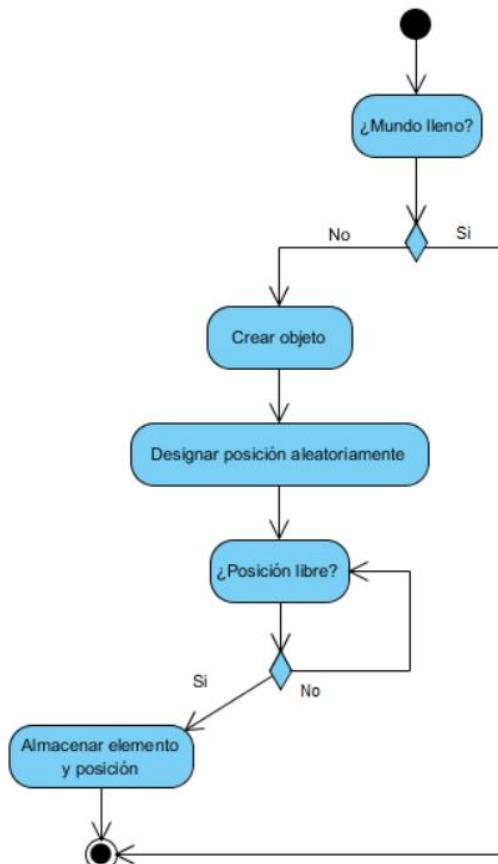


Ilustración 54 - Diagrama de actividad - Creación de un elemento

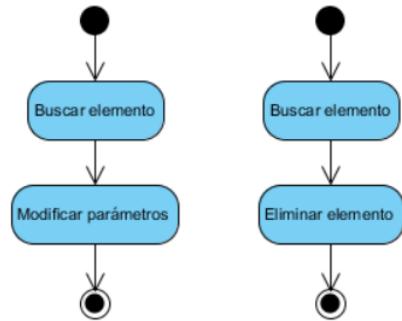


Ilustración 55 - Diagrama de actividad - Modificación y eliminación de un elemento

Las siguientes funciones se corresponden con la modificación, eliminación y búsqueda de elementos dentro de esta estructura: `getDynamic()`, `setDynamic()`, `getNoDynamic()`, `setNoDynamic()`, `getPrize()`, `setPrize()` y `search()`.

- **Colocación de los elementos en el mundo.**

Se han definido los métodos necesarios para la colocación de los elementos en el mundo, detallados a continuación:

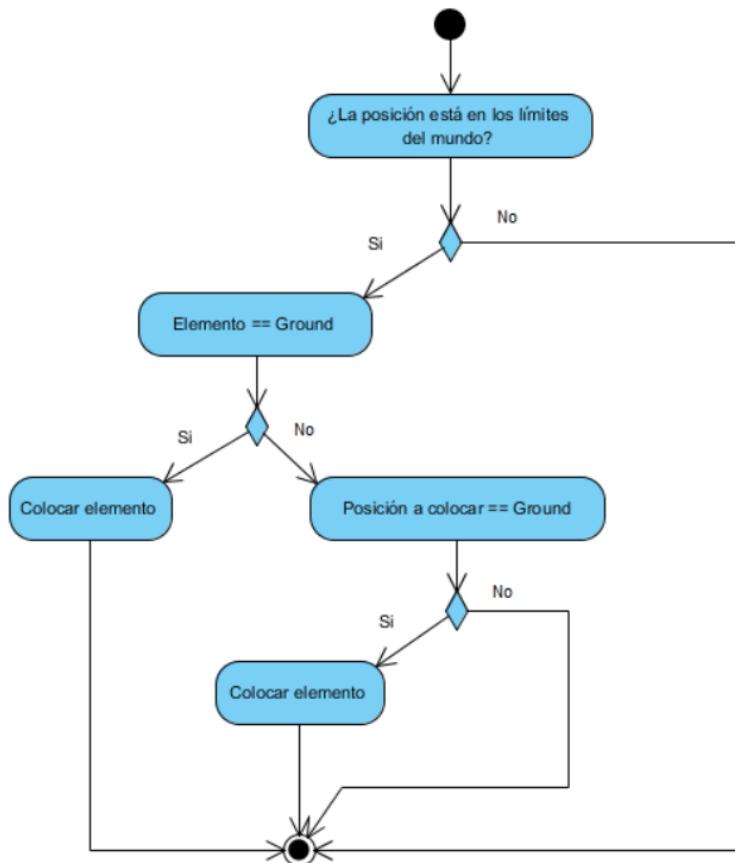


Ilustración 56 - Diagrama de actividad - Colocación de un elemento

A la hora de la implementación, se han realizado las siguiente función para este propósito: `putElement()`.

- **Acceso y modificación de los atributos de los elementos.**

Se han configurado las clases que forman los elementos para poder acceder y modificar sus atributos, necesarios a lo largo de la ejecución.

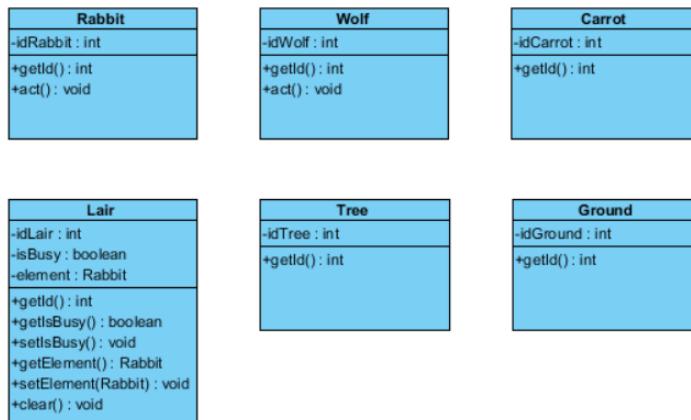


Ilustración 57 - Iteración 2 - Diagrama de clases

Se han creado 6 clases, cada una para un elemento diferente: *Rabbit*, *Wolf*, *Carrot*, *Tree*, *Lair* y *Ground*.

Además, se han creado métodos para modificar los parámetros de los elementos que se almacenan en la estructura de arrays: *setEating()*, *isEating()*, *eating()*, *getIsHide()*, *setIsHide()*, *setSleeping()*, *isSleeping()*, *sleeping()*, *hasEaten()*, *canEat()* y *defaultEat()*.

- **Modificación de la interfaz de usuario inicial para agregar los parámetros de configuración de los elementos.**

Se ha modificado la interfaz de usuario creada en un principio para agregar aquellos parámetros necesarios para la creación de los elementos, mencionados en el apartado ‘Definición de los parámetros iniciales de los elementos’ de esta etapa.

Cantidad de zanahorias	Cantidad de árboles	Cantidad de madrigueras	
2	2	2	
Cantidad de conejos	Cantidad de lobos		
2	2		
Generación de zanahorias			
Cada x turnos	50	Cantidad	4
Generación de lobos			
Cada x turnos	100	Cantidad	2

Ilustración 58 - Iteración 2 - Interfaz de usuario – Gestión de elementos

A continuación se detallan los tiempos de ejecución de esta fase:

Enero 2016						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
					1	2
3	4	5	6	7	8 Parámetros iniciales de los elementos (1h)	9
10	11 Gestión de la estructura de los elementos (4h)	12 Gestión de la estructura de los elementos (1,5h)	13 Colocación de elementos (1h)	14 Gestión de los atributos de los elementos (1,5h)	15 Modificación interfaz de usuario (0,5h)	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

■ Implementación - Iteración 2

Ilustración 59 - Tiempos de ejecución - Iteración 2 - Gestión de elementos

5.1.3. Acciones de los elementos

Para las acciones de los elementos, se han realizado las siguientes tareas:

- **Definición de los parámetros iniciales de las acciones.**

Para poder realizar las acciones de los elementos, se van a tener en cuenta los siguientes parámetros de configuración, que se incorporarán a los ya añadidos anteriormente:

- Reproducción de presa por día: indica la cantidad máxima de presas que se pueden reproducir durante un día.
- Mínimo de comidas por día: cantidad de veces por día que tienen que comer los elementos para poder sobrevivir.
- Necesidad de dormir: indica si los elementos tienen la obligación de dormir para sobrevivir y si no es necesario.
- Lugar para dormir: indica a las presas el lugar donde pueden dormir.
- Ciclos para comer: indica los ciclos que necesita un elemento para comer.
- Ciclos para dormir: indica los ciclos que necesita un elemento para dormir.
- Puntos por ciclo para la realización de acciones: indica los puntos por ciclo que tienen los elementos para poder realizar acciones.
- Consumo por acción: indica el consumo (en puntos) por cada acción realizada (ver, moverse, olfatear, oír, dormir y reproducirse).
- Rangos por acción: indica los rangos en los que actúa una acción (ver, olfatear y oír), es decir, el número de posiciones alrededor de un elemento.

- **Identificación de acciones.**

Se han identificado las acciones que pueden realizar los elementos denominados dinámicos, es decir, depredadores y presas:

- Ver: permite a un elemento ver los elementos que se encuentran a su alrededor dentro de un rango determinado, teniendo en cuenta las siguientes reglas:
 - Un elemento no puede ver otro elemento que se encuentre escondido.
 - El rango de visión de un elemento está determinado por el tiempo atmosférico.
 - Un elemento no se ve a sí mismo ni las posiciones del mundo que se encuentren vacías.
- Moverse: permite a un elemento moverse por el mundo siguiendo unas reglas de movimiento:
 - Un elemento no se puede colocar en otra posición donde ya haya un elemento, excepto si es para comer o para esconderse, en el caso de las presas.
 - Si un elemento quiere esconderse un lugar donde ya haya otro elemento escondido, no puede moverse hacia ahí.
 - Un elemento no se puede mover fuera de los límites del mundo.
- Comer: esta acción se produce cuando un elemento se mueve hacia una posición donde se encuentra un elemento al que puede comer.
- Dormir: permite a un elemento dormir durante un período de tiempo establecido. En el caso de los depredadores, pueden dormir en cualquier posición del mundo que les esté permitida; en el caso de las presas, el lugar donde pueden dormir está establecido previamente.
- Esconderse: esta acción se produce cuando un elemento presa se mueve hacia una posición donde se encuentra un elemento que permite su ocultación a los demás elementos.
- Olfatear: permite a un elemento detectar los elementos que se encuentran a su alrededor dentro de un rango establecido, pero no les revela su posición dentro del mundo.
- Oír: permite a un elemento detectar los elementos que se encuentran a su alrededor dentro de un rango establecido, pero no les revela su posición dentro del mundo.

- Reproducirse: permite a los elementos presa reproducirse. Para ello, se tiene que cumplir la siguiente condición: dos elementos presa se tienen que encontrar en posiciones contiguas (las posiciones contiguas diagonales no son válidas) y querer realizar dicha acción, además de cumplir la restricción de que sólo se pueden reproducir cada cierto período de tiempo.
 - **Gestor de acciones.**
- Se ha definido una función para controlar las acciones que pueden realizar los elementos, mencionadas en el apartado anterior. Se puede ver esta acción detallada en el siguiente diagrama:

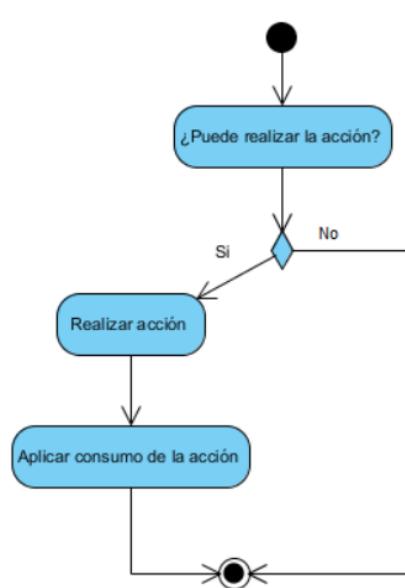


Ilustración 60 - Diagrama de actividad - Gestor de acciones

- En cuanto al código implementado, se puede ver reflejado en estas funciones: *canAct()* y *actionManager()*.
- **Control de acciones por ciclo.**

La ejecución de una acción conlleva el gasto de puntos. Cada acción tiene asociada una cantidad de puntos. Durante cada ciclo, cada elemento dispone una cantidad de puntos para realizar acciones. En caso de que un elemento no contenga puntos, no podrá realizar la acción deseada. Esta cantidad de puntos vuelve a su estado inicial al final de cada ciclo, de forma que al inicio de cada ciclo vuelva a tener todos los puntos.

Este control sobre los puntos de cada elemento se detalla a continuación:

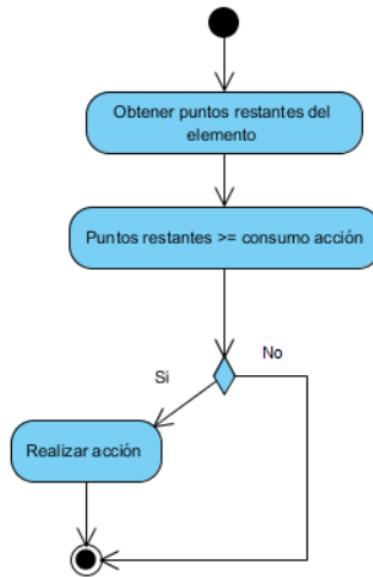


Ilustración 61 - Diagrama de actividad - Control de acciones por ciclo

- **Realización de las acciones.**

A continuación se explicará el funcionamiento de cada una de las acciones:

- Ver

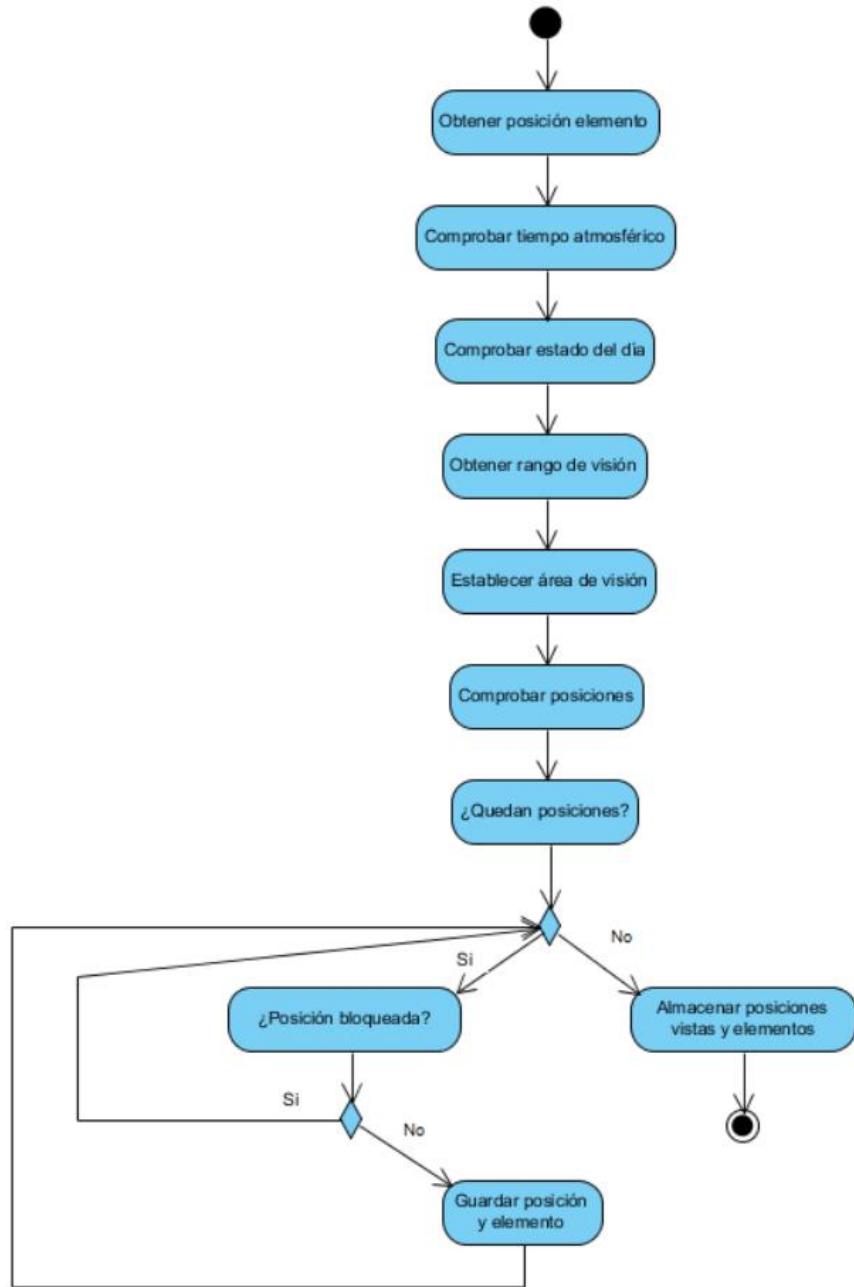


Ilustración 62 - Diagrama de actividad - Acción – Ver

Dicha acción se implementa mediante la función `see()`.

- **Moverse**

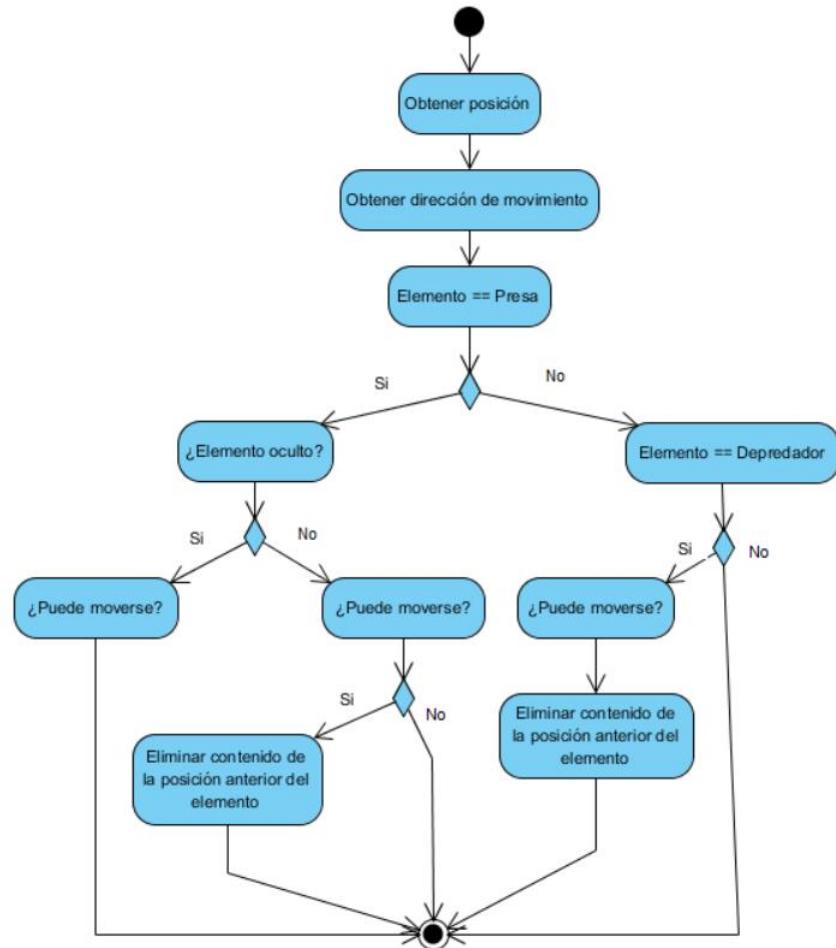


Ilustración 63 - Diagrama de actividad - Acción - Moverse

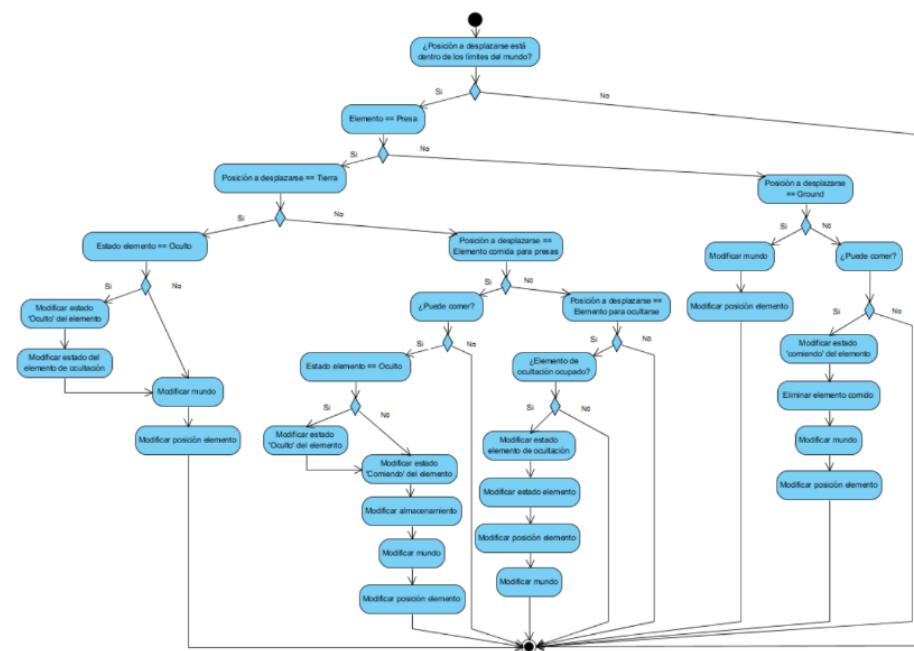


Ilustración 64 - Acción - Moverse - Método (puede moverse)

Dicha acción se implementa mediante las funciones `move()` y `canMove()`.

- **Comer**

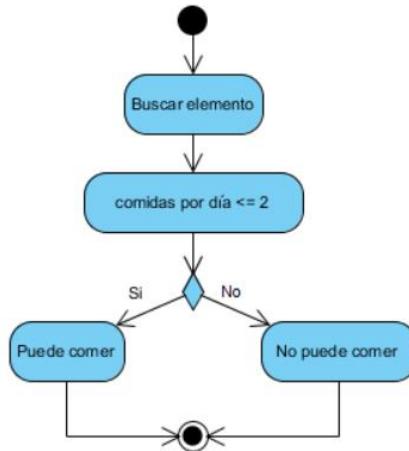


Ilustración 65 - Diagrama de actividad - Acción – Comer

Dicha acción se implementa dentro de la acción mover, ya que comer implica que haya un desplazamiento previo hacia una posición. Si en esa posición hay un alimento y el elemento puede comer, realiza la acción.

- **Dormir**

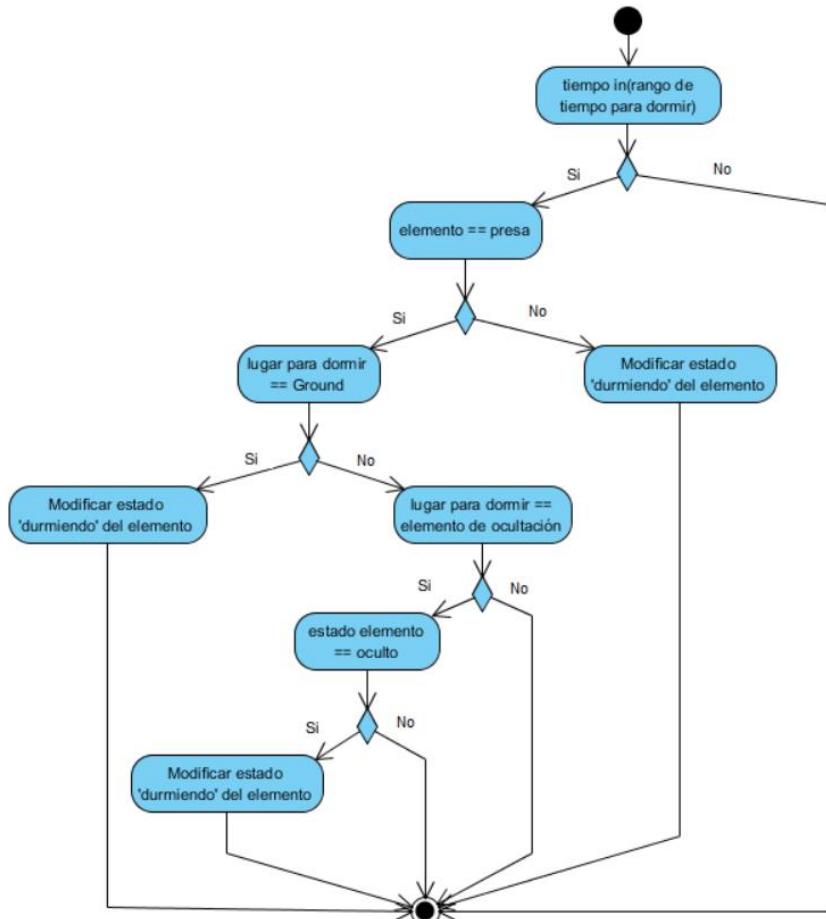


Ilustración 66 - Diagrama de actividad - Acción – Dormir

Dicha acción se implementa mediante la función `toSleep()`.

- **Olfatear**

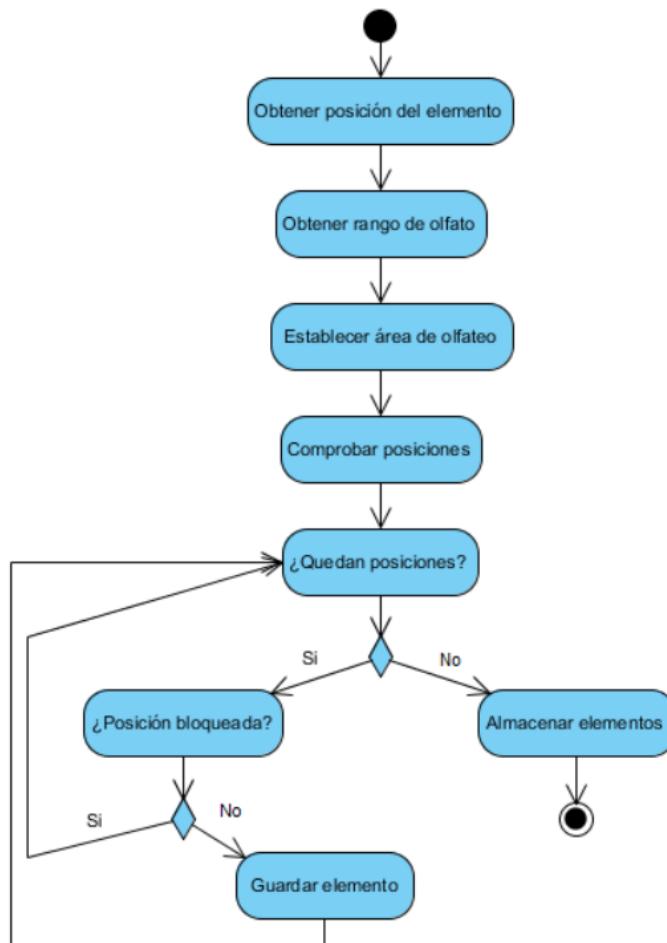


Ilustración 67 - Diagrama de actividad - Acción – Olfatear

Dicha acción se implementa mediante la función `smell()`.

- **Oír**

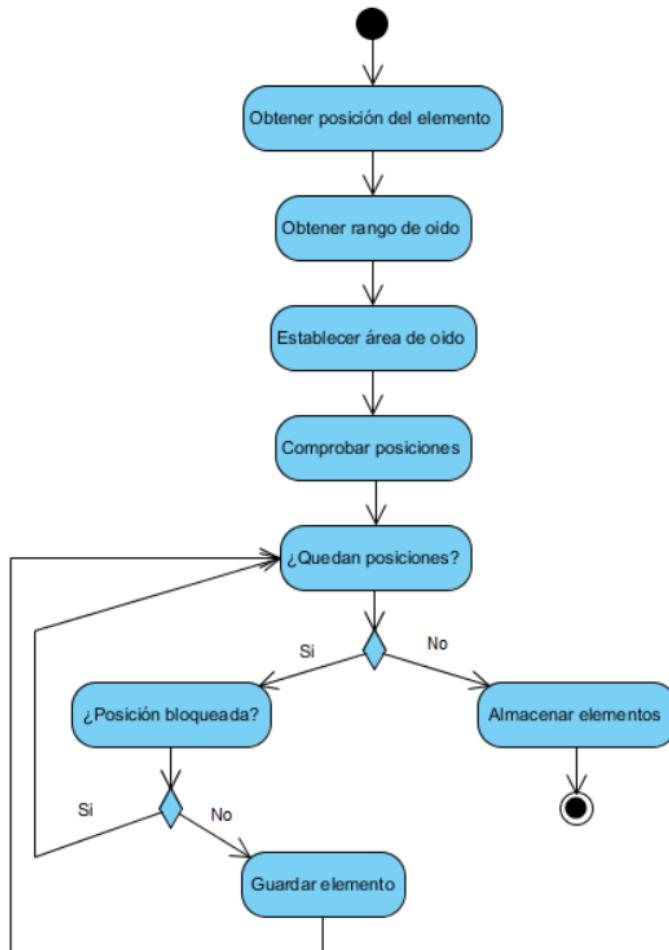


Ilustración 68 - Diagrama de actividad - Acción – Oír

Dicha acción se implementa mediante la función `hear()`.

- Reproducirse

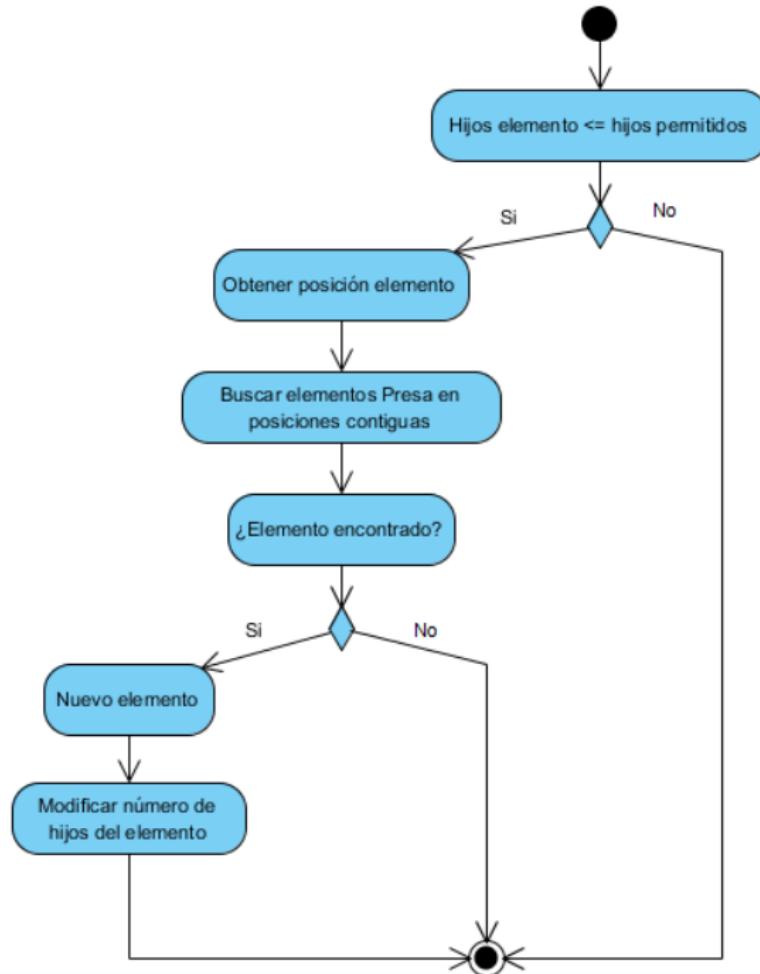


Ilustración 69 - Diagrama de actividad - Acción – Reproducirse

Dicha acción se implementa mediante la función *reproduce()*.

- **Modificación de la interfaz de usuario para agregar los parámetros de configuración de las acciones.**

Se ha modificado la interfaz de usuario para agregar aquellos parámetros necesarios para la realización de las acciones de los elementos, mencionados en el apartado ‘Identificación de acciones’ de esta etapa.

Reproducción de conejos

Límite por día	1
----------------	---

Mínimo de comidas por día

Conejos	1	Lobos	1
---------	---	-------	---

Necesidad de dormir

Conejos	si	Lobos	si
---------	----	-------	----

Lugar para dormir (conejos)

Tierra

Ilustración 70 - Iteración 2 - Interfaz de usuario - Acciones de los elementos (1)

Turnos para comer

Conejos	10	Lobos	10
---------	----	-------	----

Turnos para dormir

Conejos	20	Lobos	20
---------	----	-------	----

Ilustración 71 - Iteración 2 - Interfaz de usuario - Acciones de los elementos (2)

Puntos por turno para acciones

Conejos	3	Lobos	3
---------	---	-------	---

Consumo por acción

--Smell

Conejos	1	Lobos	1
---------	---	-------	---

--Hear

Conejos	1	Lobos	1
---------	---	-------	---

--See

Conejos	1	Lobos	1
---------	---	-------	---

--Move

Conejos	1	Lobos	1
---------	---	-------	---

--Sleep

Conejos	1	Lobos	1
---------	---	-------	---

--Reproduce

Conejos	1	Lobos	1
---------	---	-------	---

Ilustración 72 - Iteración 2 - Interfaz de usuario - Acciones de los elementos (3)

Rango de visión	
Conejos	1
Lobos	1
Rango de olfato	
Conejos	1
Lobos	1
Rango de oído	
Conejos	1
Lobos	1

Ilustración 73 - Iteración 2 - Interfaz de usuario - Acciones de los elementos (4)

Finalmente, se ha realizado un diagrama que recoge las principales actividades que transcurren durante la ejecución de la aplicación, desde que comienza la ejecución del script hasta su finalización:

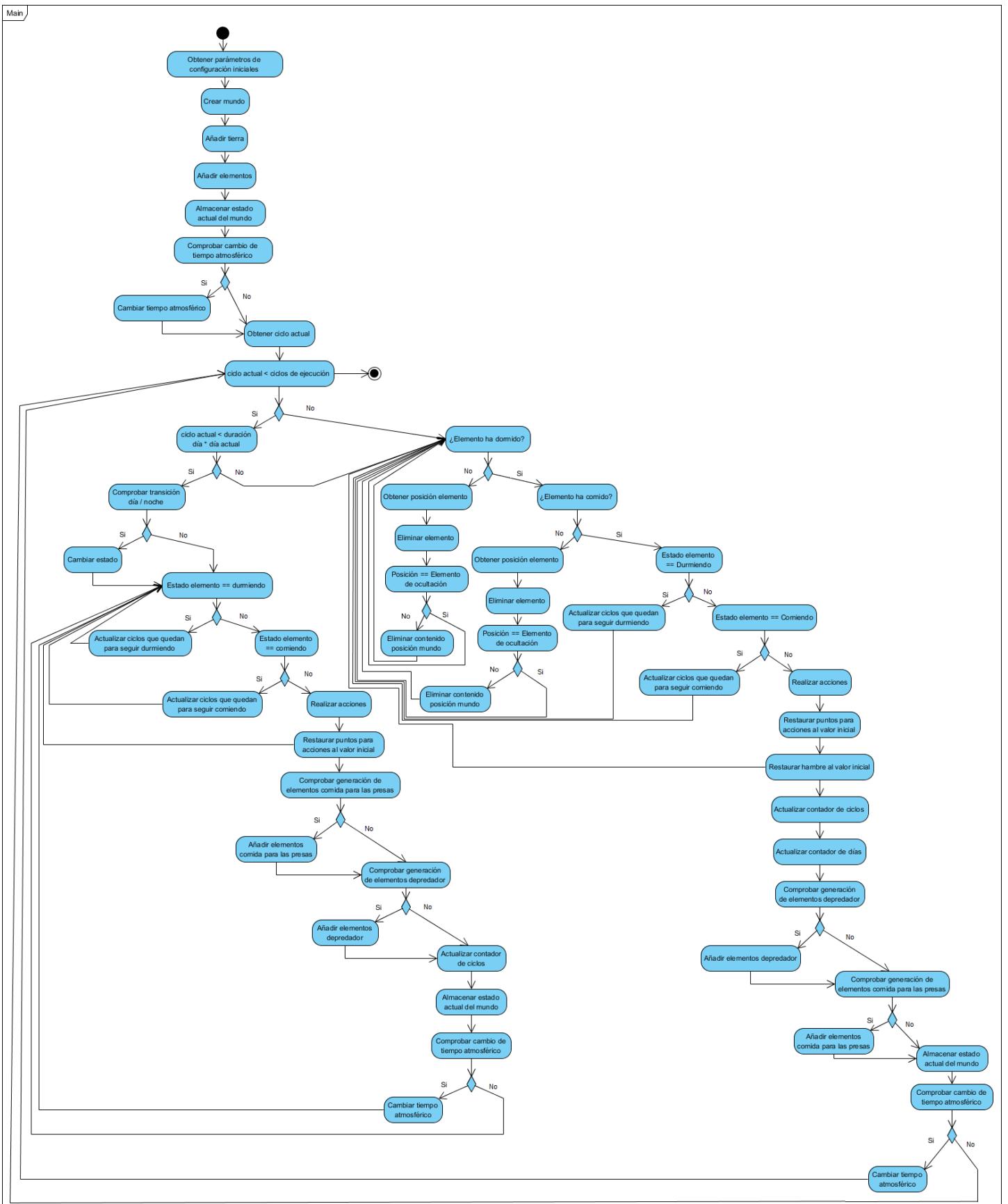


Ilustración 74 - Iteración 2 - Diagrama de actividad - Principal

A continuación se detallan los tiempos de ejecución para esta fase:

Enero 2016						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18 0h	19 Parámetros iniciales e identificación de acciones (3,5h)	20 Gestor de acciones y control de acciones por ciclo (1h)	21 Acción ver, olfatear y oír (5,5h)	22 Acción mover y comer (5h)	23
24	25 0h	26 Acciones dormir y reproducirse (5h)	27 Modificación de la interfaz de usuario y pruebas (2,5h)	28	29	30
31						

■ Implementación - Iteración 2

Ilustración 75 - Tiempos de ejecución - Iteración 2 - Acciones de los elementos

5.1.4. Realización de pruebas

- Definición de los parámetros iniciales del mundo ✓
- Definición, creación, modificación y visualización del mundo ✓
- Configuración de los estados del mundo ✓
- Creación de archivos de log ✓
- Creación de la interfaz de configuración ✓
- Definición de los parámetros iniciales de los elementos ✓
- Creación, modificación, eliminación y búsqueda de elementos en el almacenamiento interno ✓ ✗
 - En este aspecto han surgido problemas a la hora de buscar los elementos en los arrays de almacenamiento correspondientes, sobre todo al modificar atributos internos de cada elemento. El principal motivo es la alta complejidad que va adquiriendo la estructura de arrays al ir añadiendo más datos y parámetros a cada elemento, ya que se tiene que almacenar más información y la forma actual para gestionarla no es efectiva. Este aspecto afecta tanto a la modificación como a la eliminación.
 - En cuanto a la creación de elementos, las pruebas han sido correctas.
- Colocación de los elementos en el mundo ✓

- Acceso y modificación de los elementos ✓ ✗
 - Las pruebas han sido correctas en cuanto a este aspecto si se tiene en cuenta que los elementos se prueban de forma individual.
 - El fallo aparece debido al motivo planteado en el apartado de prueba (“Creación, modificación, eliminación y búsqueda de elementos en el almacenamiento interno”), al realizar estas pruebas de acceso y modificación en la estructura completa, no de forma individual.
- Modificación de la interfaz de usuario para agregar los parámetros de configuración de los elementos ✓
- Definición de los parámetros iniciales de las acciones ✓
- Identificación de acciones ✓
- Gestor de acciones ✓
- Control de acciones por ciclo ✓
- Realización de las acciones ✓ ✗
 - Ver ✓
 - Moverse ✓ ✗
 - Los elementos de ocultación desaparecen del mundo sin motivo aparente. Ocurre cuando los elementos presa se mueven entre elementos de ocultación o cuando salen de ellos.
 - El resto de movimientos posibles funciona correctamente.
 - Comer ✓
 - Dormir ✗
 - Cuando se aplica la restricción “Lugar obligatorio para dormir” (lugar para las presas), esta acción deja de funcionar.
 - Cuando se aplica el parámetro que decide si un elemento necesita dormir para sobrevivir, la aplicación deja de funcionar.
 - Olfatear ✓
 - Oír ✓
 - Reproducirse ✗
 - El sistema se vuelve inestable, dejando de funcionar en la mayoría de las ocasiones debido a la gran cantidad de elementos creados debido a la reproducción de las presas.

- Los tiempos de ejecución se ven altamente incrementados en los casos en los que la aplicación funciona de forma correcta.
- Modificación de la interfaz de usuario para agregar los parámetros de configuración de las acciones ✓

5.1.4.1. Tiempos de ejecución

Las pruebas mencionadas se han realizado con la ejecución de un mundo pequeño y con pocos elementos, es decir, únicamente lo necesario para ver si los resultados son los esperados y poder decir que se ha superado la prueba.

Pero, ¿qué pasa si aumentamos el tamaño del mundo y metemos una cantidad de elementos bastante grande? A continuación se muestra una tabla con las pruebas realizadas:

Tamaño	Presas	Depredadores	Comida para presas	Elementos de ocultación	Obstáculos	Ciclos de ejecución	Días simulados	Tiempo de ejecución (s)	Tiempo de ejecución (min)	Memoria consumida (mb)
10	10	1	1	1	1	100	1	0,04	-	2
10	10	1	1	1	1	10000	100	2,95	-	2
10	10	1	1	1	1	50000	500	Error de memoria	-	-
50	50	1	1	1	1	100	1	0,68	-	2
50	50	1	1	1	1	10000	100	11,59	-	2
50	50	1	1	1	1	50000	500	Error de tiempo	-	-
100	100	1	1	1	1	100	1	2,91	-	2
100	100	1	1	1	1	10000	100	279,65	4,66	6
100	100	1	1	1	1	50000	500	1401,86	23,36	10

Ilustración 76 - Tiempos de ejecución - Iteración 2

Cabe destacar que los resultados obtenidos no son para nada los esperados. En algunas ocasiones, el sistema se colapsa debido a que sobrepasa el tiempo de ejecución permitido en el servidor (una vez aumentado, por defecto son 30 segundos) y la memoria del servidor (una vez aumentada). Además, cuando la ejecución se completa satisfactoriamente, los tiempos de ejecución son demasiado elevados.

A continuación se detallan los tiempos de ejecución para esta fase:

Enero 2016						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27 Modificación de la interfaz de usuario y pruebas (2,5h)	28	29	30
31						

Implementación - Iteración 2

Ilustración 77 - Tiempos de ejecución - Iteración 2 - Pruebas

5.1.5. Conclusión y mejoras

En este caso, no todas las pruebas han resultado exitosas, por lo que es necesario realizar algunos cambios:

[IT2 – M1] : Cambiar la estructura principal de la aplicación para solventar los problemas con los tiempos de ejecución.

[IT2 – M2] : El archivo que guarda los estados del mundo a lo largo de la ejecución se puede volver demasiado grande y difícil de tratar debido a la gran cantidad de información que se va guardar en él.

[IT2 – M3] : Necesidad de modificar la estructura de búsqueda de elementos en el almacenamiento interno.

[IT2 – M4] : Modificar la acción “mover” para que funcione correctamente.

[IT2 – M5] : Modificar la acción “dormir” para poder aplicar los parámetros de configuración.

[IT2 – M6] : Cambiar el sistema de reproducción para evitar un colapso del sistema.

[IT2 – M7] : Optimización de código.

Los tiempos de ejecución de esta fase se ven reflejados en la fase anterior, **5.1.4**

Realización de pruebas.

5.2. Iteración 3

El objetivo principal de esta iteración es solventar los problemas ocurridos en la iteración anterior, así como realizar mejoras que favorezcan el mejor funcionamiento de la aplicación.

5.2.1. Configuración y creación del mundo

- Definición de los parámetros iniciales del mundo**

Respecto a los parámetros de configuración inicial establecidos en la iteración 1, no se ha hecho ningún cambio.

- Definición, creación, modificación y visualización del mundo**

Se mantiene la misma estructura que en la iteración 1, ya que no ha surgido ningún problema.

- Configuración de los estados del mundo (tiempo, día/noche y tiempo atmosférico) y sus correspondientes cambios dentro de la ejecución del mismo.**

Se continúa con la estructura planteada en la iteración 1, puesto que no hay ningún error que corregir.

- **Creación de archivos de log**

El archivo que guarda los estados del mundo ha sufrido un pequeño cambio, debido a la gran cantidad de información que se generaba [IT2 – M2]. En vez de almacenar el contenido de cada posición del mundo, ahora se guardan únicamente las posiciones que contienen algún elemento.

Además se ha creado una nueva función, conf(), para llenar el log al inicio de la aplicación con los datos que introduce el usuario.

- **Interfaz de configuración de los parámetros del mundo**

No ha sufrido ningún cambio respecto a la iteración anterior puesto que no se han detectado nuevos parámetros de configuración.

A continuación se detallan los tiempos de ejecución para esta fase:

Enero 2016						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28 0h	29 0h	30
31						

■ Implementación - Iteración 3

Ilustración 78 - Tiempos de ejecución - Iteración 3 - Configuración y creación del mundo (1)

Febrero 2016						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
	1 Archivos de log (3h)	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29					

■ Implementación - Iteración 3

Ilustración 79 - Tiempos de ejecución - Iteración 3 - Configuración y creación del mundo (2)

5.2.2. Gestión de elementos

- **Definición de los parámetros iniciales de los elementos**

No se ha agregado ningún parámetro de configuración inicial para los elementos, por lo que se mantienen los mencionados en la iteración 1.

- **Creación, modificación, eliminación y búsqueda de elementos dentro de la estructura de almacenamiento interno para los mismos**

Dado que la búsqueda de elementos dentro de la estructura de almacenamiento provocaba un problema en los tiempos de ejecución [IT2 – M1], ya que estos eran excesivamente grandes, se han hecho modificaciones en la misma. Se ha implementado una estructura indexada para facilitar la búsqueda de los elementos, de forma que se realiza un acceso directo en vez de realizar una búsqueda del elemento sobre la estructura. De esta forma se solventa el problema mencionado en [IT2 – M3].

Se continúa manteniendo la misma estructura en cuanto a división por categorías: estáticos, dinámicos y premio.

Las demás acciones sobre un elemento continúan con la misma estructura, únicamente se han realizado modificaciones de optimización.

Manteniendo el mundo con la misma estructura pero modificando la estructura que da forma a los elementos, se ha solventado este problema consiguiendo tiempos de ejecución aceptables.

- **Colocación de los elementos en el mundo**

Se ha realizado una pequeña modificación en la colocación de elementos en el mundo, siendo la razón de esta la optimización de código [IT2 – M7].

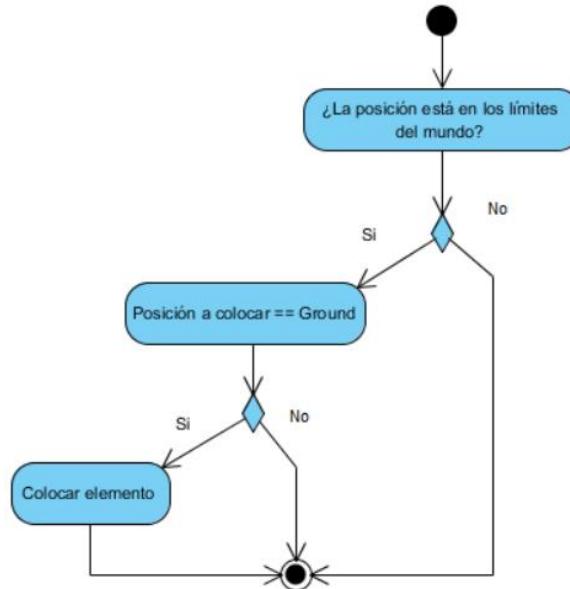


Ilustración 80 - Diagrama de activad - Colocación de elementos

- **Acceso y modificación de los atributos de los elementos**

Se han realizado importantes cambios en este aspecto, ya que se ha reestructurado el almacenamiento de los datos de los elementos. Como consiguiente, se pueden apreciar los cambios en el siguiente diagrama de clases:

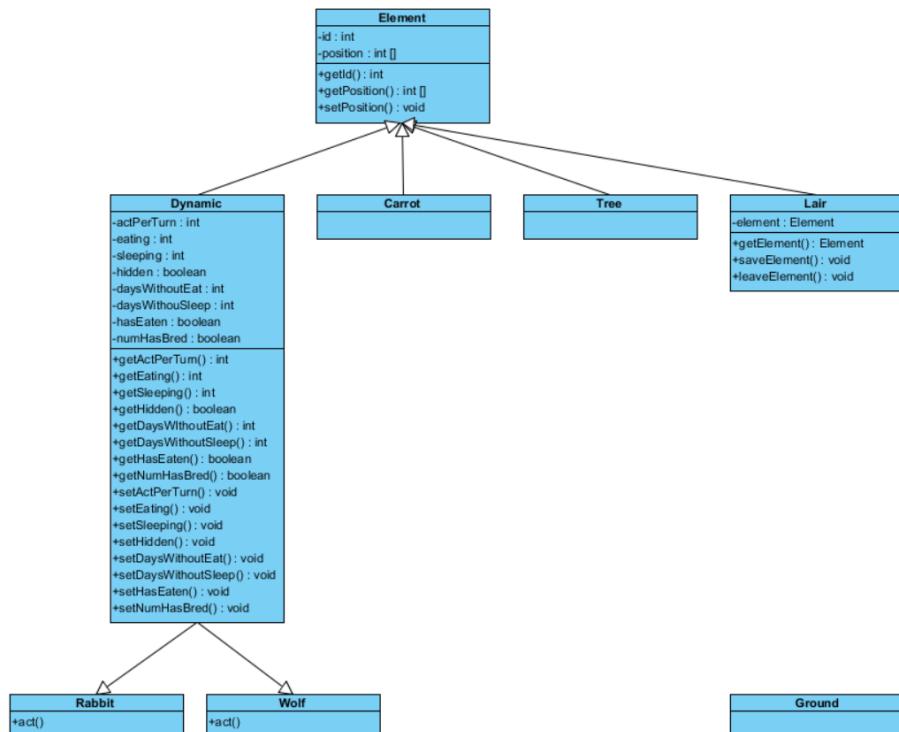


Ilustración 81 - Diagrama de clases - Iteración 3

En la iteración anterior se mantenía una estructura en la que había seis clases, pertenecientes a cada tipo de elemento. Dado que no se seguía ningún tipo de jerarquía y esto podía ser uno de los causantes de los altos tiempos de ejecución, se ha decidido crear una jerarquía dentro de la estructura de los elementos tal y como se aprecia en el diagrama anterior, haciendo uso de la herencia de objetos que proporciona el lenguaje PHP. Se esta forma se clasifican los elementos en base al tipo y se mejora el sistema de identificadores para cada elemento, ya que se evita tener un identificador para cada tipo de elemento.

Una vez realizado este cambio, se han satisfecho las sospechas respecto a la mejora en los tiempos de ejecución, ya que han sido satisfactorias [IT2 – M1].

- Modificación de la interfaz de usuario para agregar los parámetros de configuración de los elementos**

Debido a que no se han detectado nuevos parámetros de configuración, se mantiene la misma interfaz que en la iteración anterior.

A continuación se detallan los tiempos de ejecución para esta fase:

Febrero 2016						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
	1	2 Estructura almacenamiento elementos [7,5h]	3 0h			6
7	8	9	10	11 0h	12 0h	13
14	15 0h	16 Colocación de los elementos (0,25h)	17 Acceso y modificación de los atributos de los elementos (3,5h)	18	19	20
21	22	23	24	25	26	27
28	29					

■ Implementación - Iteración 3

Ilustración 82 - Tiempos de ejecución - Iteración 3 - Gestión de elementos

5.2.3. Acciones de los elementos

- Definición de los parámetros iniciales de las acciones**

No se ha agregado ningún parámetro de configuración inicial para las acciones de los elementos, por lo que se continúa con los definidos en la iteración 1.

- Identificación de acciones**

No se ha añadido ninguna acción nueva, con lo que se siguen manteniendo las descripciones realizadas en la iteración 1.

- **Gestor de acciones**

Se mantiene la misma estructura descrita en la iteración 1.
- **Control de acciones por ciclo**

Se mantiene el mismo sistema para controlar si un elemento puede realizar o no una acción, controlando la cantidad de puntos que le quedan después de realizar cada acción.
- **Realización de las acciones**
 - **Ver**

Se mantiene la misma estructura respecto de la iteración anterior. Únicamente se han realizado pequeños cambios de optimización, los cuales no afectan a la funcionalidad descrita.
 - **Moverse**

En la anterior iteración esta funcionalidad no se ejecutaba correctamente **[IT2 – M4]**, por lo que se ha reestructurado de la siguiente forma para corregir y mejorar su funcionamiento:

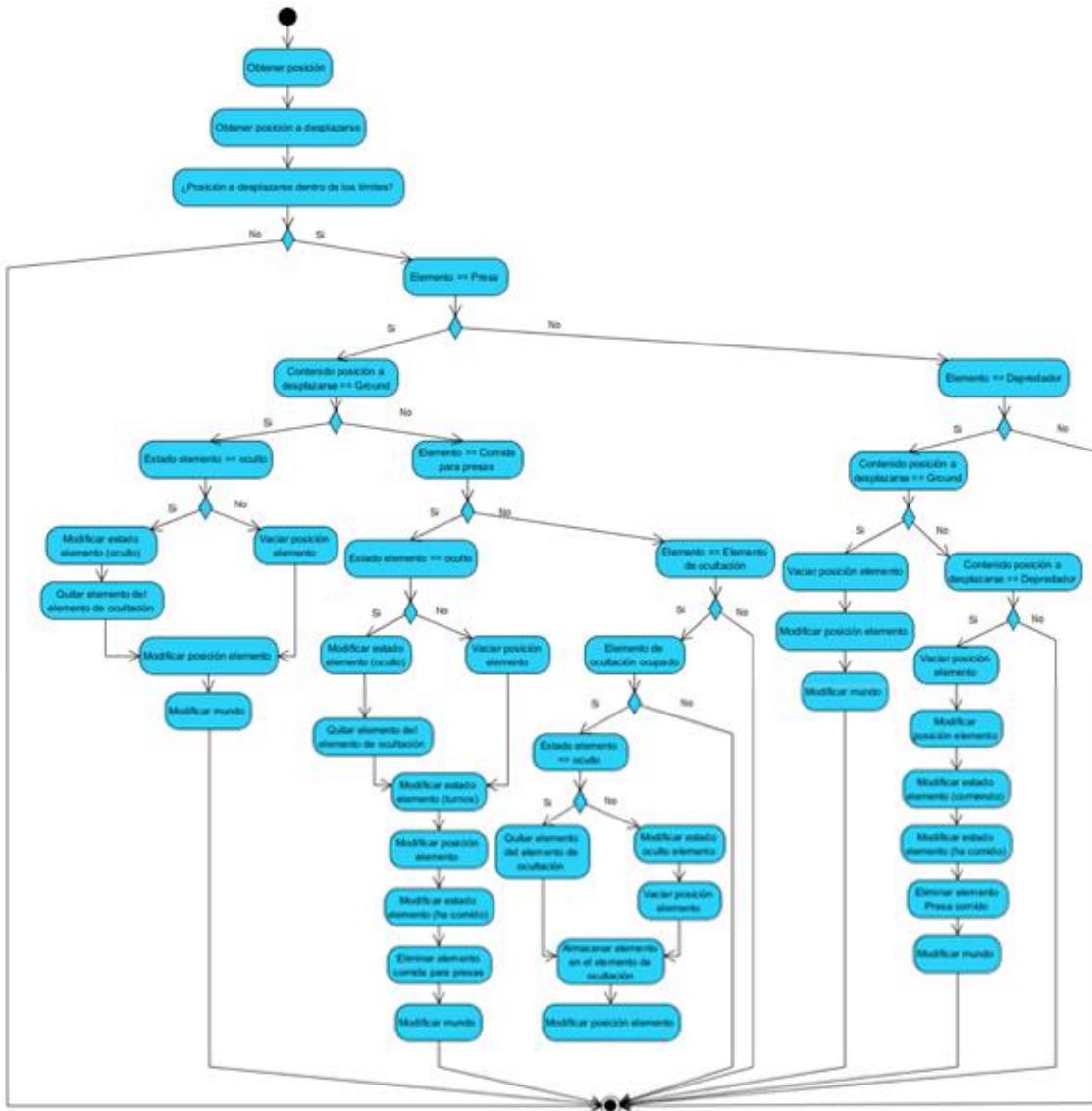


Ilustración 83 - Diagrama de actividad - Acción - Moverse

Tras realizar estos cambios, se han conseguido solventar los problemas encontrados y se ha mejorado la estructura.

○ Dormir

Dado que esta funcionalidad no se ejecutaba correctamente [IT2 – M5], se ha corregido obteniendo como resultado:

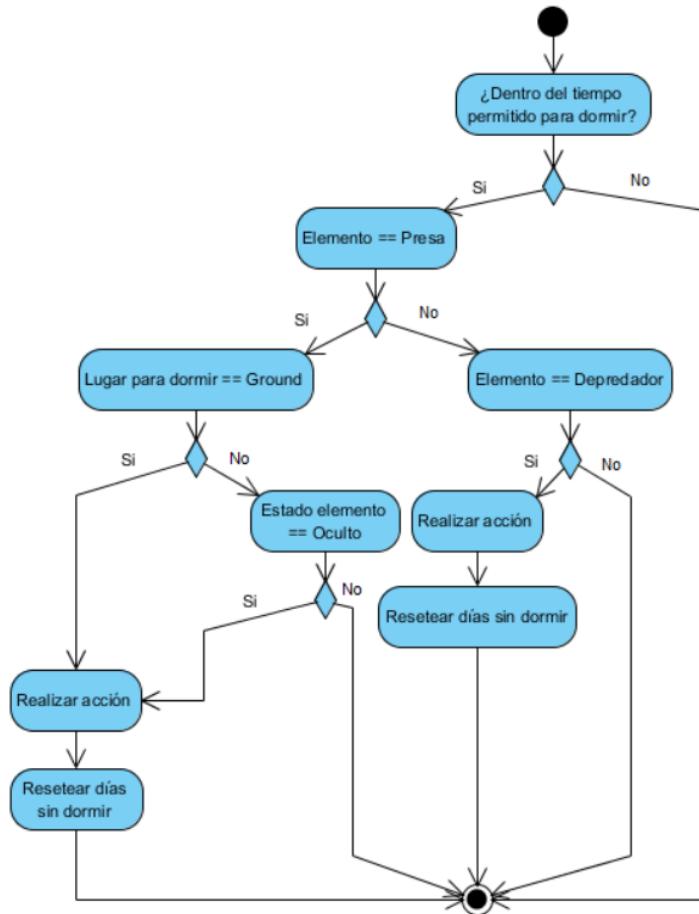


Ilustración 84 - Diagrama de actividad - Acción – Dormir

De esta forma responde con normalidad a los parámetros de configuración inicial y los elementos realizan esta acción correctamente.

- **Olfatear**

Se continúa con la misma estructura que en la iteración anterior.

- **Oír**

Se continúa con la misma estructura que en la iteración anterior.

- **Reproducirse**

Debido a los problemas surgidos en esta funcionalidad en la iteración anterior

[IT2 – M6], se ha reestructurado esta acción:

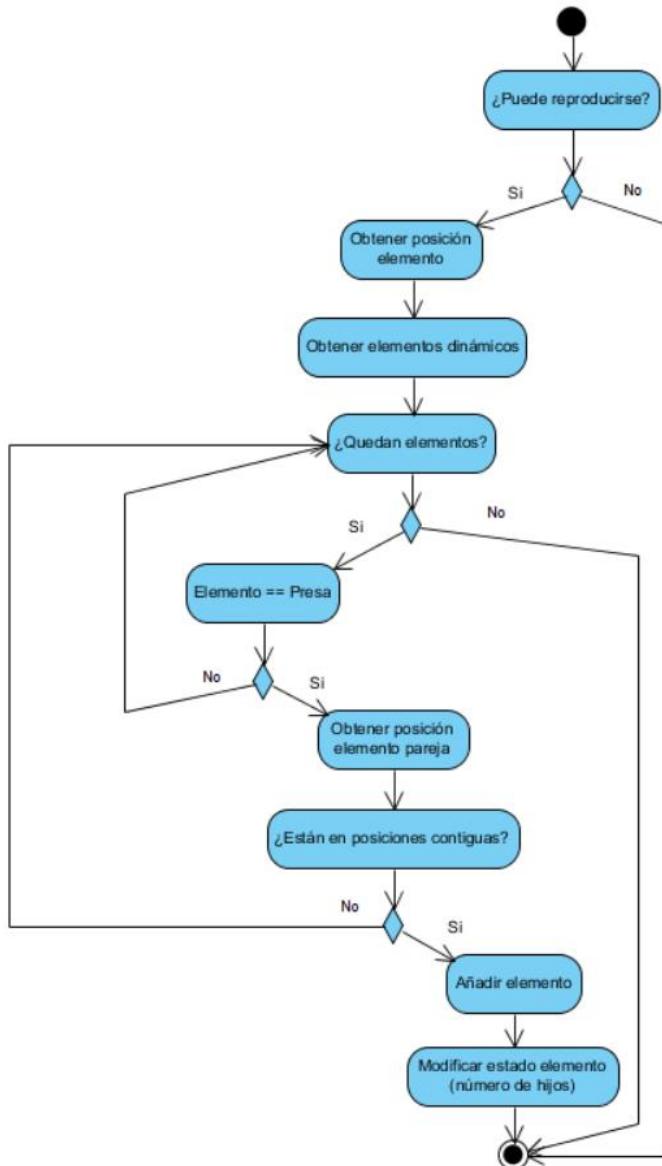


Ilustración 85 - Diagrama de actividad - Acción - Reproducirse

De esta forma, se evita que el sistema se colapse por la excesiva cantidad de elementos generada.

- **Modificación de la interfaz de usuario para agregar los parámetros de configuración de las acciones**

Continúa la misma interfaz que en la iteración anterior debido a que no se ha agregado ningún parámetro de configuración para las acciones.

Finalmente, se ha realizado un diagrama que recoge las principales actividades que transcurren durante la ejecución de la aplicación, desde que comienza la ejecución del script hasta su finalización:

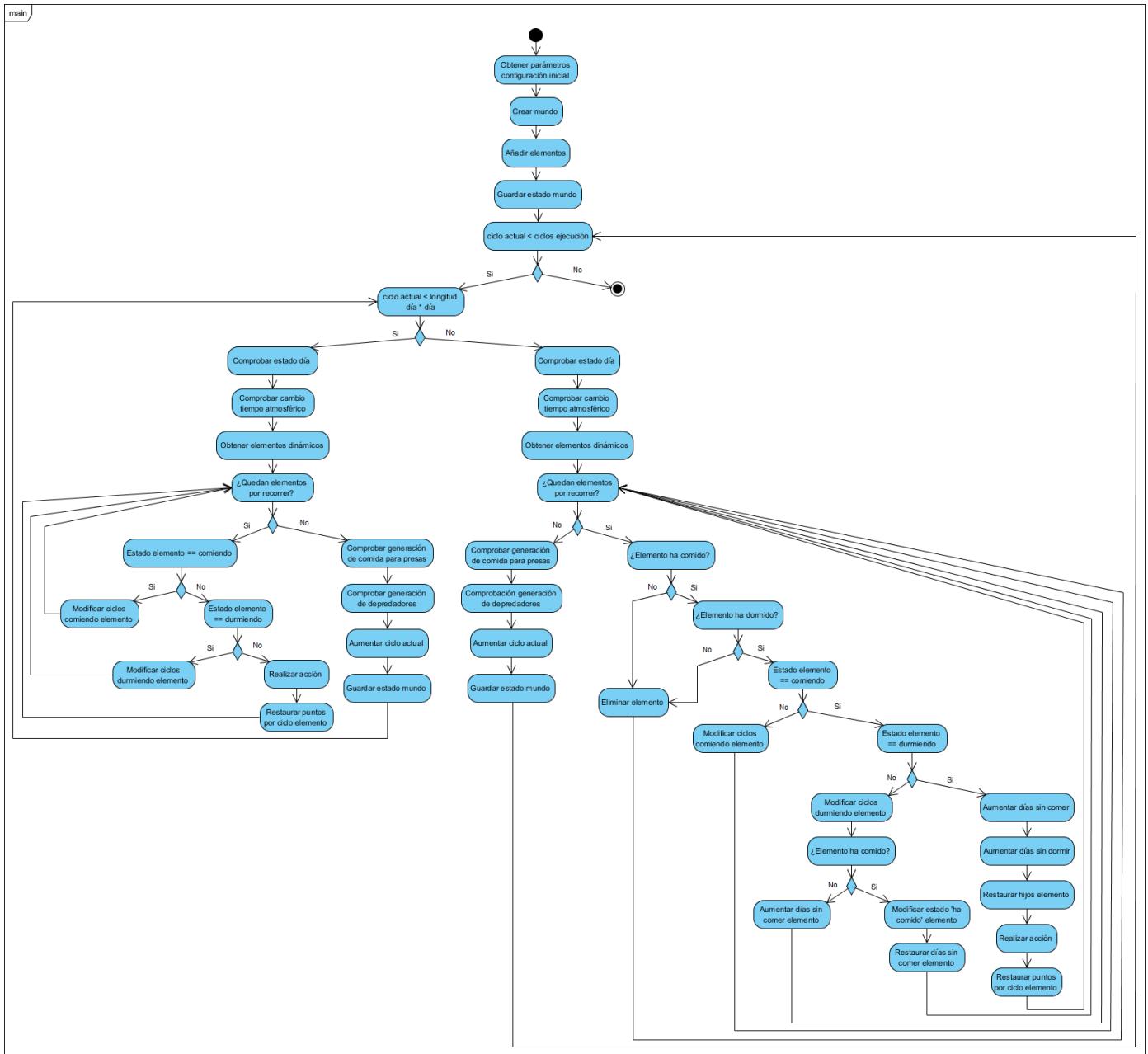


Ilustración 86 - Iteración 3 - Diagrama de actividad - Principal

A continuación se detallan los tiempos de ejecución para esta fase:

Febrero 2016						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18 Acción mover y comer (4,5h)	19 Acción mover y comer (3h)	20
21	22 0h	23 Acción dormir (2.5h)	24 Acción reproducirse (3.5h)	25 Pruebas (2,25h)	26	27
28	29					

■ Implementación - Iteración 3

Ilustración 87 - Tiempos de ejecución - Iteración 3 - Acciones de los elementos

5.2.4. Realización de pruebas

Se han realizado pruebas para comprobar el funcionamiento de esta etapa:

- Definición de los parámetros iniciales del mundo ✓
- Definición, creación, modificación y visualización del mundo ✓
- Configuración de los estados del mundo ✓
- Interfaz de configuración de los parámetros del mundo ✓
- Creación de archivos de log ✓
- Definición de los parámetros iniciales de los elementos ✓
- Creación, modificación, eliminación y búsqueda de elementos ✓
- Colocación de los elementos en el mundo ✓
- Acceso y modificación de los atributos de los elementos ✓
- Interfaz de configuración de los elementos ✓
- Definición de los parámetros iniciales de los elementos ✓
- Identificación de acciones ✓
- Gestor de acciones ✓
- Control de acciones por ciclo ✓
- Realización de acciones ✓
- Interfaz de configuración de las acciones ✓

5.2.4.1. Tiempos de ejecución

Una vez realizados los cambios en la arquitectura de los elementos, se han conseguido unos tiempos de ejecución bastante aceptables, que se reflejan en la siguiente tabla:

Tamaño	Presas	Depredadores	Comida para presas	Elementos de ocultación	Obstáculos	Ciclos de ejecución	Días simulados	Tiempo de ejecución (s)	Tiempo de ejecución (min)	Memoria consumida (mb)
100	100	1	1	1	1	10000	100	44,41	-	2
100	100	1	1	1	1	50000	500	220,09	3,67	8
100	100	1	1	1	1	10000	100	32,92	-	4
100	100	1	1	1	1	50000	500	172,94	2,88	10
50	50	20	20	20	20	10000	100	9,06	-	2
50	50	20	20	20	20	10000	100	34,25	-	4
50	50	20	20	20	20	50000	500	164,56	2,74	10
100	100	1	1	1	1	10000	100	172,73	2,88	8
100	100	1	1	1	1	50000	500	346,57	5,78	18

Ilustración 88 - Tiempos de ejecución – Iteración 3

A continuación se detallan los tiempos de ejecución para esta fase:

Febrero 2016						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29					

Implementación - Iteración 3

Ilustración 89 - Tiempos de ejecución - Iteración 3 - Pruebas

5.2.5. Conclusión y mejoras

A lo largo de esta iteración se han conseguido grandes avances. En primer lugar, se han corregido los de la iteración anterior en cuanto a funcionalidades se refiere; en segundo lugar y, como aspecto más relevante, se ha conseguido mejorar el sistema para conseguir un sistema estable manteniendo unos tiempos de ejecución razonables. Estos tiempos de ejecución varían según la configuración que se establezca inicialmente.

Después de las pruebas realizadas se ha llegado a la conclusión de que es necesario establecer unos límites en cuanto a los parámetros iniciales (tamaño del mundo y cantidad de elementos) para poder mantener un sistema sostenible. Si la cantidad de elementos y las dimensiones del mundo son muy grandes, el sistema no será capaz de gestionar tanta cantidad de información y se colapsará, provocando un error en la aplicación.

Hasta el momento se ha conseguido una primera versión funcional de la aplicación.

5.3. Iteración 4

Una vez finalizadas las dos iteraciones anteriores, se han conseguido grandes avances de cara la versión final de la aplicación. Además de solucionar errores, se ha mejorado el rendimiento de la aplicación con creces con los cambios aplicados.

En esta iteración se van a realizar los siguientes cambios:

- Definición de nuevos parámetros de configuración.
- Almacenamiento de la configuración inicial.
- Modificación en los archivos de log.
- Modificación de algunas acciones.
- Modelado del comportamiento de los elementos.

5.3.1. Configuración y creación del mundo

- **Definición de los parámetros iniciales del mundo**

Se ha hecho un pequeño cambio respecto a la iteración anterior. Se ha quitado el parámetro “Duración de un día”, que establecía cuántos ciclos duraba un día, por “Duración del día”, que establece el número de ciclos que dura el día, es decir, desde que amanece hasta que anocchece. Como complemento a este parámetro, se sigue manteniendo la duración de la noche; por lo que la suma de los dos hace un día completo. Este cambio se ha realizado para evitar una confusión con los términos día / noche y para evitar restas innecesarias a la hora de programar esta funcionalidad.

- **Almacenamiento de los parámetros iniciales**

Se ha creado una función que recoja los parámetros introducidos por el usuario y los guarde en el sistema:

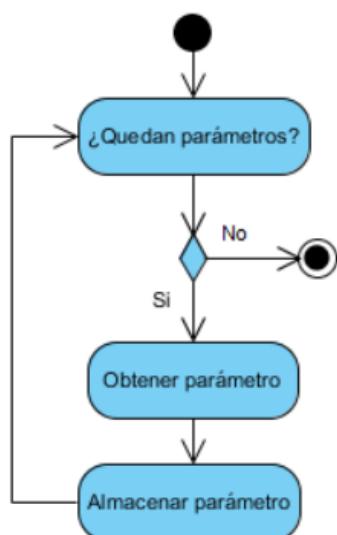


Ilustración 90 - Diagrama de actividad - Almacenamiento de los parámetros iniciales

Para facilitar al usuario la repetición de experimentos con los mismos parámetros iniciales o similares, se guarda en un archivo la configuración inicial para que la próxima vez que vuelva a ejecutar la aplicación pueda cargar dicho archivo y, por tanto, los parámetros de configuración inicial serán los que se almacenaban en el archivo guardado.

Esta funcionalidad se ve reflejada en la función *conf()*.

- **Definición, creación, modificación y visualización del mundo**

Se ha modificado la forma en la que se guardan los distintos estados del mundo. En vez de utilizar un .txt, se ha pasado a utilizar un .csv, dado que se puede estructurar el documento y es más fácil a la hora de visualizar el resultado y trabajar con esos datos posteriormente.

Además, se han establecido unos límites en cuanto al tamaño del mundo para evitar una sobrecarga del sistema: como máximo se podrá crear un mundo de 2000 posiciones.

También se ha impuesto una restricción en cuanto a los ciclos de ejecución máximos: 20.000. Esta restricción se realiza por el mismo motivo y para evitar largas esperas mientras se ejecuta la aplicación.

- **Configuración de los estados del mundo (tiempo, día/noche y tiempo atmosférico) y sus correspondientes cambios dentro de la ejecución del mismo**

Se ha modificado la configuración debido al cambio realizado en la definición de los parámetros iniciales. Las funciones *getLengthDay()* y *getLengthNight()* retornan la longitud del día y la noche respectivamente.

- **Creación de archivos de log**

Se ha realizado una importante modificación en cuanto a la estructura de los archivos:

- Configuración: se almacena en un .json la configuración inicial para poder cargarla en próximas ejecuciones y mantener así la misma configuración sin necesidad de volver a introducir los parámetros. Esta opción se ha implementado mediante la función *conf()*.
- Mundo: se almacenan en un .csv los distintos estados del mundo a lo largo de la ejecución, viéndose reflejado en la función *writeWorld()*.
- Acciones: se almacenan en un .csv las distintas interacciones que ocurren en el mundo así como las acciones que realizan los elementos. Cada acción que ocurre en el mundo se almacena en este archivo, con lo cual se realiza este registro en todas las funciones que implique la realización de una acción.

- Debug: se mantiene el mismo uso que en iteraciones anteriores.

- **Modificación de la interfaz de configuración inicial**

Se modificó el parámetro descrito en “Definición de los parámetros iniciales del mundo”.

Duración total		
Duración del día	Duración de la noche	

Ilustración 91 - Iteración 4 - Configuración y creación del mundo - Interfaz de usuario

A continuación se detallan los tiempos de ejecución para esta fase:

Febrero 2016						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	0h			0h	

■ Implementación - Iteración 4

Ilustración 92 - Tiempos de ejecución - Iteración 4 - Configuración y creación del mundo

(1)

Marzo 2016						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
		1 Parámetros iniciales y almacenamiento (1,75h)	2 Definición, creación modificación y visualización (1,5h)	3 Configuración de los estados del mundo (0,75h)	4 Archivos de log e interfaz de usuario (2h)	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

■ Implementación - Iteración 4

Ilustración 93 - Tiempos de ejecución - Iteración 4 - Configuración y creación del mundo

(2)

5.3.2. Gestión de elementos

- **Definición de los parámetros iniciales de los elementos**

Se ha eliminado la funcionalidad “regeneración de depredadores”, por lo que también se han eliminado los parámetros de configuración inicial correspondientes.

A su vez, se ha añadido un nuevo parámetro: tiempo de vida de la comida de las presas, que establece durante cuántos ciclos se conserva la comida para las presas.

Pasado ese número de ciclos, la comida desaparece, siempre y cuando no se haya comido antes.

- **Creación, modificación, eliminación y búsqueda de elementos dentro de la estructura de almacenamiento interno para los mismos**

Dado el éxito de la estructura creada en la iteración anterior para este propósito, se continúa manteniendo la misma.

- **Colocación de los elementos en el mundo**

Se continúa con la estructura descrita en la iteración anterior.

- **Acceso y modificación de los atributos de los elementos**

Con el objetivo de mejorar el funcionamiento de la aplicación y de almacenar los nuevos parámetros de configuración inicial, se han modificado algunos atributos de los elementos y añadido o eliminado otros, quedando el esquema de la siguiente forma:

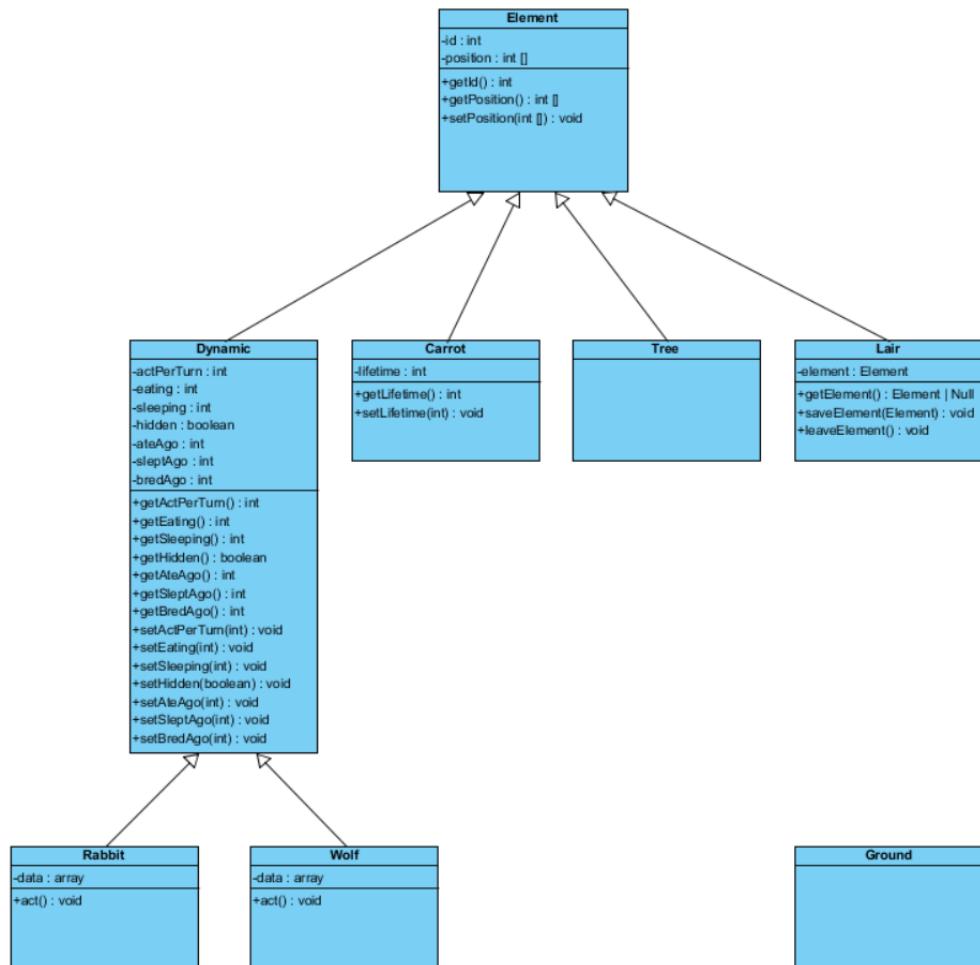


Ilustración 94 - Diagrama de clases - Iteración 4

En el caso de la comida para las presas, se ha añadido el tiempo de vida con sus correspondientes métodos de obtención y modificación.

Para los elementos dinámicos, se han añadido los métodos `getAteAgo()`, `getSleptAgo()` y `getBredAgo()`, con sus respectivos modificadores y se han eliminado los métodos `getDaysWithoutEat()`, `getDaysWithoutSleep()`, `getHasEaten()` y `getNumHasBred()`, con sus respectivos modificadores.

Además, tanto en la clase depredador como en la clase presa, se añade un nuevo atributo, `data`, que almacenará información del propio elemento.

Estos cambios se han realizado debido a la modificación de algunas acciones que se explicarán más adelante, además de para optimizar código.

- **Modificación de la interfaz de usuario para agregar los nuevos parámetros de configuración**

Se ha modificado la interfaz de configuración, de forma que se han añadido los nuevos parámetros de configuración.

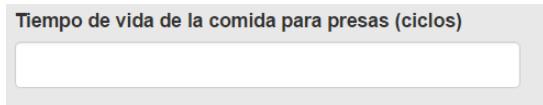


Ilustración 95 - Iteración 4 - Gestión de elementos - Interfaz de usuario

A continuación se detallan los tiempos de ejecución para esta fase:

Marzo 2016						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
		1	2	3	4	5
6	7	8	9	10	11	12
	0h	Parámetros iniciales de los elementos (1h)	Estructura de almacenamiento interno (0,75h)	Acceso y modificación de los atributos de los elementos (5h)	Acceso y modificación de los atributos de los elementos (2h)	
13	14	15	16	17	18	19
	0h	Modificación de la interfaz de usuario (1,5h)				
20	21	22	23	24	25	26
27	28	29	30	31		

■ Implementación - Iteración 4

Ilustración 96 - Tiempos de ejecución - Iteración 4 - Gestión de elementos

5.3.3. Acciones de los elementos

- **Definición de los parámetros iniciales de las acciones**

Se han añadido o modificado los siguientes parámetros de configuración:

- Ciclos sin comer para depredadores y presas: establece el número de ciclos que un elemento puede permanecer sin comer. Una vez superado ese número, el elemento muere.
- Ciclos sin dormir para depredadores y presas: establece el número de ciclos que un elemento puede permanecer sin dormir. Una vez superado ese número, el elemento muere.
- Reproducción de elementos cada x ciclos para depredadores y presas y cantidad de hijos: establece cada cuántos ciclos se puede reproducir un elemento, así como la cantidad máxima de hijos que puede tener. Esta cada cantidad se genera de forma aleatoria entre 0 y la cantidad introducida, por lo que puede no tener hijos, tener algunos o tener la cantidad introducida.
- Estado “saciado” de los depredadores y las presas: establece durante cuántos ciclos un elemento se encuentra saciado, es decir, no necesita comer.
- Estado “sin sueño” de los depredadores y las presas: establece durante cuántos ciclos un elemento se encuentra sin sueño, es decir, no necesita dormir.

- Zona de confort para depredadores y presas: estado que indica el número de ciclos después de comer y dormir en los que un elemento se encuentra en el estado de confort. Este parámetro se utiliza a la hora de la reproducción de los elementos, ya que sólo se permite la reproducción cuando un elemento se encuentra dentro del estado de confort.
- Comportamiento de los elementos: se permite configurar cómo se van a comportar los elementos dentro del mundo, es decir, qué acciones van a realizar y cuándo.

- **Identificación de acciones**

No se ha añadido ninguna acción para los elementos.

- **Gestor de acciones**

Se continúa con el mismo gestor de acciones que en la iteración anterior.

- **Control de acciones por ciclo**

Se mantiene el control realizado en la iteración anterior.

- **Realización de acciones**

- **Ver**

Se mantiene la misma estructura respecto de la iteración anterior.

- **Moverse**

Debido a los cambios realizados en la parametrización inicial, se ha modificado esta acción para adaptarse a ellos y cumplir con la nueva funcionalidad. Se pueden reflejar dichos cambios a continuación, siendo modificada en el código la función *move()*:

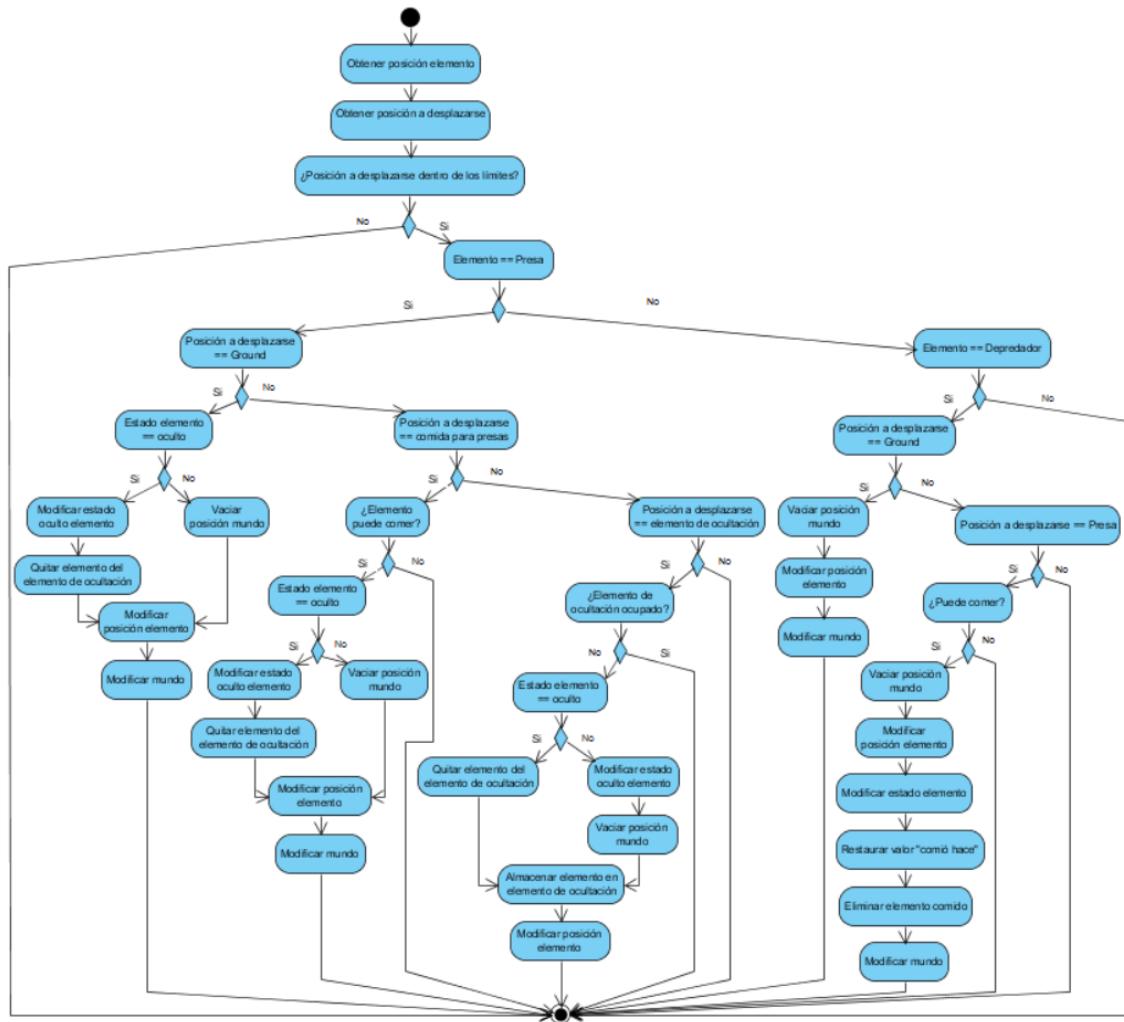


Ilustración 97 - Diagrama de actividad - Acción - Mover

- **Dormir**

Se ha modificado esta acción para adaptarla a los nuevos parámetros de configuración, siendo modificada en el código la función `toSleep()`:

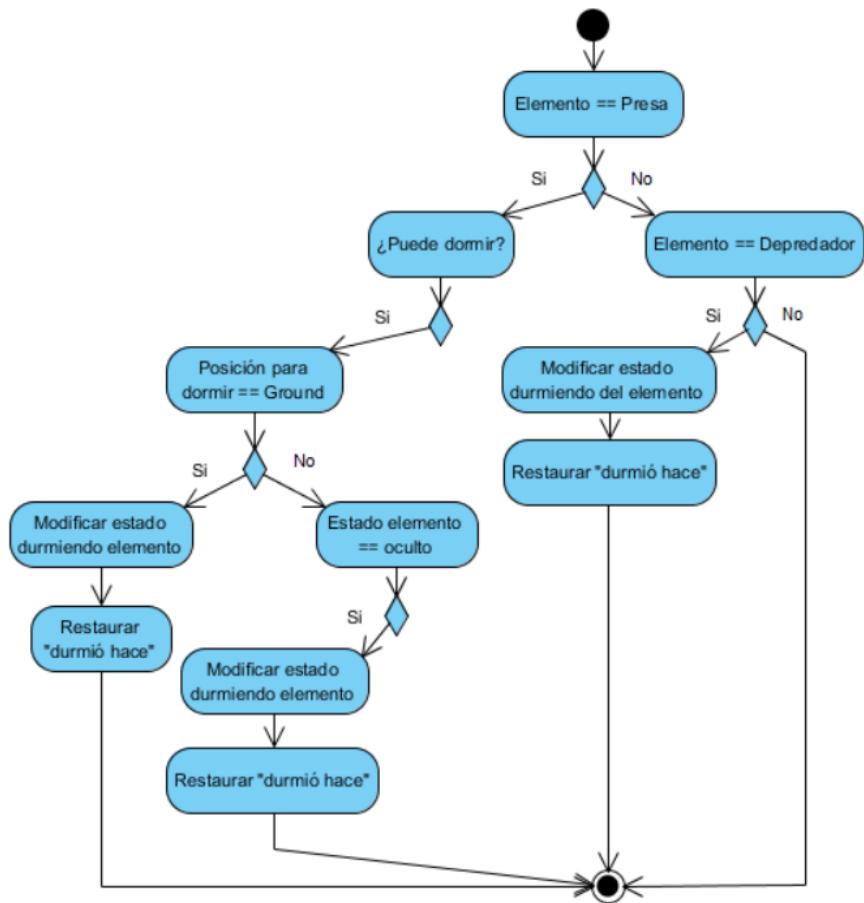


Ilustración 98 - Diagrama de actividad - Acción - Dormir

- **Olfatear**

Se mantiene igual que en la iteración anterior.

- **Oír**

Se mantiene igual que en la iteración anterior.

- **Reproducirse**

Se ha modificado para adaptarse a los nuevos parámetros de configuración, siendo modificada en el código la función *breed()*;

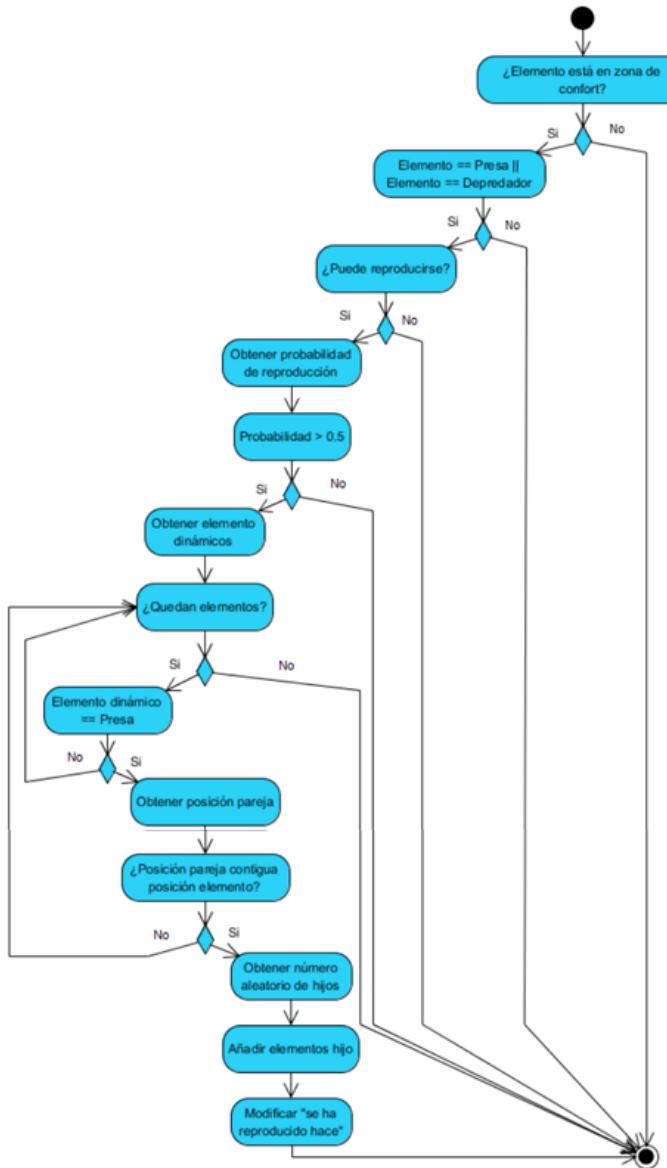


Ilustración 99 - Diagrama de actividad - Acción - Reproducirse

- **Personalización del comportamiento de los elementos**

Se ha introducido un aspecto bastante interesante para el usuario: la opción de poder personalizar el comportamiento de los elementos dentro del mundo, de poder decidir qué acción van a realizar y en qué momento lo van a hacer. Únicamente se podrá modelar el comportamiento de los depredadores y las presas, puesto que son los únicos elementos que realizan alguna acción en el sistema.

Para ello, se ha creado una API que el usuario podrá consultar para realizar esta personalización. En ella se detallan las funciones del mundo a las que tendrá acceso, únicamente con el propósito de obtener información, y las funciones de los elementos, de forma que se pueda decidir qué acción realizar, cuándo y cómo. Dicha API se podrá ver en el archivo *api.php*.

Ya que se trata de un cambio bastante relevante, se ha optado por dar la opción de ejecutar dos sistemas diferentes. En el primer sistema, se mantienen todas las funcionalidades descritas en apartados anteriores excepto la personalización del comportamiento de los elementos, ya que los elementos realizan las acciones en base a la aleatoriedad, sin seguir ningún tipo de patrón. En el segundo sistema, cuenta también con todo lo anterior y se agrega la opción de poder configurar el comportamiento de los elementos.

Este modelado del comportamiento se realizará en la interfaz de configuración inicial junto con los demás parámetros de configuración.

¿Cómo se aplica esta configuración? En el método *act()* de cada elemento se cargará el código introducido por el usuario para el comportamiento de dicho elemento, con la correspondiente verificación del mismo. De esta forma, si el usuario comete algún error al escribir su código, el sistema dará un aviso del error para que el usuario pueda corregirlo y no se colapsará el sistema. Dicha verificación se controlará mediante dos archivos que servirán de test para el código introducido: *testPredator.php* y *testPrey.php*.

- **Modificación de la interfaz de usuario para agregar los parámetros de configuración de las acciones**

Se ha modificado la interfaz de configuración, de forma que se han añadido los nuevos parámetros establecidos.

Presas - Pueden reproducirse cada (ciclos)	Presas - Cantidad máxima
<input type="text"/>	<input type="text"/>
Depredadores - Pueden reproducirse cada (ciclos)	Lobos - Cantidad máxima
<input type="text"/>	<input type="text"/>

Ilustración 100 - Iteración 4 - Acciones de los elementos - Interfaz de usuario (1)

Presas - Están saciados durante (ciclos)	Depredadores - Están saciados durante (ciclos)
<input type="text"/>	<input type="text"/>

Ilustración 101 - Iteración 4 - Acciones de los elementos - Interfaz de usuario (2)

Presas - Sin sueño durante (ciclos)	Depredadores - Sin sueño durante (ciclos)
<input type="text"/>	<input type="text"/>

Ilustración 102 - Iteración 4 - Acciones de los elementos - Interfaz de usuario (3)

Presas - Hasta (ciclos) después de comer	Depredadores - Hasta (ciclos) después de comer
<input type="text"/>	<input type="text"/>
Presas - Hasta (ciclos) después de dormir	Depredadores - Hasta (ciclos) después de dormir
<input type="text"/>	<input type="text"/>

Ilustración 103 - Iteración 4 - Acciones de los elementos - Interfaz de usuario (4)



Ilustración 104 - Iteración 4 - Acciones de los elementos - Interfaz de usuario (5)

Finalmente, se ha realizado un diagrama que recoge las principales actividades que transcurren durante la ejecución de la aplicación, desde que comienza la ejecución del script hasta su finalización:

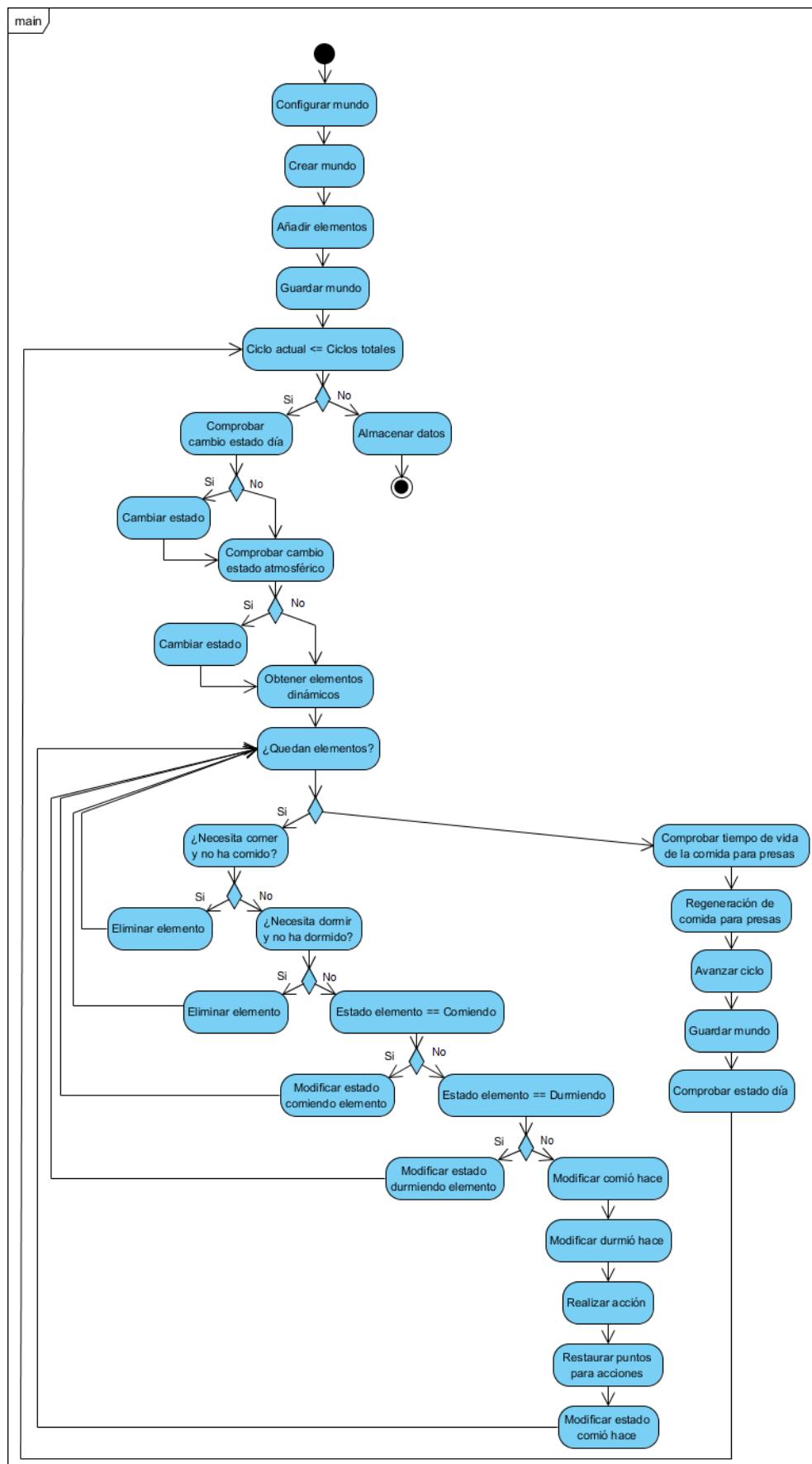


Ilustración 105 - Iteración 4 - Diagrama de actividad - Principal

A continuación se detallan los tiempos de ejecución para esta fase:

Marzo 2016						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	Definición de los parámetros iniciales de las acciones (2h)	Acción moverse (5h)	Acción dormir (4h)
20	21	22	23	24	25	26
27	28	0h	0h	Acción reproducirse (5,5h)	Personalización del comportamiento de los elementos (4,5h)	

■ Implementación - Iteración 4

Ilustración 106 - Tiempos de ejecución - Iteración 4 - Acciones de los elementos (1)

Abril 2016						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
					1	2
					Interfaz de usuario y pruebas (6h)	
3	4	5	6	7	8	9
10						16
17						23
24						30

■ Implementación - Iteración 4

Ilustración 107 - Tiempos de ejecución - Iteración 4 - Acciones de los elementos (2)

5.3.4. Realización de pruebas

- Definición de los parámetros iniciales del mundo ✓
- Almacenamiento de los parámetros iniciales ✓
- Definición, creación, modificación y visualización del mundo ✓
- Configuración de los estados del mundo ✓
- Creación de archivos de log ✓
- Modificación de la interfaz de configuración inicial para agregar los parámetros de configuración del mundo ✓

- Definición de los parámetros iniciales de los elementos ✓
- Creación, modificación, eliminación y búsqueda de elementos ✓
- Colocación de los elementos en el mundo ✓
- Acceso y modificación de los atributos de los elementos ✓
- Modificación de la interfaz de configuración inicial para agregar los parámetros de configuración de los elementos ✓
- Definición de los parámetros iniciales de las acciones ✓
- Gestor de acciones ✓
- Control de acciones por ciclo ✓
- Realización de acciones ✓
- Personalización del comportamiento de los elementos ✓
- Modificación de la interfaz de configuración inicial para agregar los parámetros de configuración de las acciones ✓

5.3.4.1. Tiempos de ejecución

Una vez realizados los cambios mencionados anteriormente, se han obtenido los siguientes tiempos de ejecución:

Tamaño	Presas	Depredadores	Comida para presas	Elementos de ocultación	Obstáculos	Ciclos de ejecución	Días simulados	Tiempo de ejecución (s)	Tiempo de ejecución (min)	Memoria consumida (mb)
30	30	100	100	50	50	10000	100	58,11	-	2
30	30	100	10	50	50	10000	100	47,03	-	2
50	50	200	10	500	500	200	10000	100	44	-
50	50	200	10	500	500	200	20000	200	71,11	1,19
50	50	200	0	500	500	200	10000	100	130,79	2,18
50	50	400	10	500	500	200	1000	10	16,4	-
50	50	400	10	500	500	200	36500	365	125,55	2,09
50	50	500	5	500	500	100	36500	365	257,36	4,29

Ilustración 108 - Tiempos de ejecución – Iteración 4

Respecto a la iteración anterior se han conseguido bajar los tiempos de ejecución y, como cambio más significativo, el descenso notable en el consumo de memoria.

A continuación se detallan los tiempos de ejecución para esta fase:

Abril 2016						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
					1 Interfaz de usuario y pruebas (6h)	2
3	4	5	6	7	8	9
10						16
17						23
24						30

■ Implementación - Iteración 4

Ilustración 109 - Tiempos de ejecución - Iteración 4 - Pruebas

5.3.5. Conclusión y mejoras

Después de finalizar esta iteración, la parte del back-end correspondiente al funcionamiento de la aplicación está casi terminada. En este instante, un usuario puede ejecutar la aplicación sin ningún tipo de problema, aunque no sería para nada amigable. La funcionalidad con la que más problemas ha habido ha sido la personalización del comportamiento de los elementos, debido a que se le permite al usuario introducir código en la aplicación. Hay que controlar a la perfección cada línea de código recibida para evitar cualquier tipo de ataque o simplemente, tener una buena gestión de errores para mostrarle al usuario los posibles errores que cometa y que los pueda corregir para poder ejecutar la aplicación.

Como mejora, continúa la optimización de código, que se hace relevante debido a que cuanto mejor sea el código y más optimizado esté, los tiempos de ejecución serán cada vez menores y mejorarán la experiencia del usuario.

5.4. Iteración 5

5.4.1. Configuración de gráficas y archivos de log

¿Qué hacer con todos los datos obtenidos? Uno de los pilares de esta aplicación es la obtención de datos para que el usuario puede realizar un análisis sobre lo ocurrido durante la ejecución y contrastar datos con diferentes ejecuciones.

Para ello, se dispone de una serie de gráficas interactivas junto con archivos que muestran información acerca de lo ocurrido durante la ejecución, implementadas en el archivo *output.php*.

- Visualización del mundo: en un tablero se muestra donde se encuentra cada elemento en cada momento de la ejecución. Además, se permite hacer un seguimiento de la misma permitiendo al usuario desplazarse temporalmente y cambiar la velocidad de reproducción, así como las funciones de un reproductor de vídeo (play, pause y stop). Para ello se ha implementado una estructura en forma de tablero con los botones para las interacciones al lado derecho. Para comprobar su funcionamiento se puede chequear el archivo output.js, en el que se encuentran las funciones play(), pause(), stop(), printWorld() y update(), que permite moverse a por los distintos estados del mundo mediante los botones “inicio”, “fin”, “anterior” y “siguiente”.

Es necesario tener previamente los datos para poder renderizarlos, por lo que se leen de un archivo y se mandan desde el servidor al cliente, ya que toda la visualización se realiza en cliente.

- Gráficas interactivas: muestran la evolución de los elementos a lo largo de la ejecución. Se han escogido una serie de gráficas que se consideran relevantes como aportación de información al usuario:

- Población total: se muestra en una misma gráfica el conjunto de depredadores, presas y comida para las presas a lo largo del tiempo.
- Población de presas: se muestra el conjunto de presas a lo largo del tiempo.
- Población de depredadores: se muestra el conjunto de depredadores a lo largo del tiempo.
- Población de comida para las presas: se muestra el conjunto de comida para las presas a lo largo del tiempo.
- Presas cazadas: se muestra la cantidad de presas cazadas por los depredadores a lo largo del tiempo.
- Alimentos para las presas comidos: se muestra la cantidad de alimento para las presas a lo largo del tiempo.
- Depredadores y presas muertos por no comer: se muestra la cantidad de animales que han muerto por no comer.
- Depredadores y presas muertos por no dormir: se muestra la cantidad de animales que han muerto por no dormir.
- Reproducción de presas: se muestra la cantidad de nuevas presas a lo largo del tiempo.
- Reproducción de depredadores: se muestra la cantidad de nuevos depredadores a lo largo del tiempo.

- Tiempo atmosférico: muestra la evolución del tiempo atmosférico a lo largo del tiempo.

Se han escogido las gráficas de google, Charts. Una vez que se han generado todos los datos referentes a lo ocurrido durante la ejecución, se envían al cliente, en donde se crean las gráficas y se rellenan con los datos recibidos para su posterior muestreo. Se puede comprobar su funcionamiento en el archivo charts.js. Cada función que se encuentra en ese archivo pertenece a una gráfica diferente.

- Archivos de log

- Configuración inicial: se permite descargar un archivo que permite al usuario volver a cargar los mismos parámetros de configuración la próxima vez que quiera ejecutar la aplicación.
- Registro de acciones: muestra en un archivo todas las acciones que han transcurriendo en el mundo a lo largo de la ejecución.
- Mundo: muestra el estado del mundo a lo largo de la ejecución.

- Información relativa al tiempo de ejecución y la memoria consumida.

A continuación se detallan los tiempos de ejecución para esta fase:

Abril 2016						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
					1	2
3	4 0h	5 0h	6 Configuración de gráficas y archivos de log (2h)	7 Configuración de gráficas y archivos de log (3h)	8 Configuración de gráficas y archivos de log (2,5h)	9
10	11 0h	12 Configuración de gráficas y archivos de log (4h)	13 Configuración de gráficas y archivos de log (2h)	14 0h	15 Configuración de gráficas y archivos de log (2,5h)	16
17	18 0h	19 0h	20 Configuración de gráficas y archivos de log (3,5h)	21 Configuración de gráficas y archivos de log (4h)	22 0h	23
24	25 0h	26 Configuración de gráficas y archivos de log (2,5h)	27 Configuración de gráficas y archivos de log (2h)	28	29	30

■ Implementación - Iteración 5

Ilustración 110 - Tiempos de ejecución - Iteración 5 - Archivos de log

5.4.2. Multilenguaje

Se ha querido realizar una aplicación que se pueda ejecutar a nivel internacional. Para ello, es necesaria la implantación de un sistema multilenguaje.

Dado que el país de origen del desarrollo de esta aplicación es España y, más concretamente Galicia, se han implantado tanto el castellano como el gallego. Para llevarla al ámbito internacional, dado que el idioma internacional es el inglés, también se ha implantado en esta aplicación.

Por tanto, se puede disfrutar esta aplicación en tres idiomas diferentes.

La gestión del multilenguaje se realiza de dos formas: mediante PHP y mediante Javascript.

En el archivo *language.php*, se gestiona el multilenguaje de la parte del servidor. Este archivo hace de puente entre la información que se pide para mostrar y el diccionario de datos. Desde la página se pide un texto y un idioma, y este proceso devuelve el texto que se pide en el idioma correspondiente. De la misma forma ocurre con el archivo *language.js*. Esta gestión se divide en dos partes debido a que los mensajes que se pueden generar en base a lo que introduzca el usuario y las gráficas se realizan del lado del cliente, por tanto es necesario utilizar el lenguaje Javascript para su gestión. En lo que se refiere a los demás textos, ya que son fijos, se utiliza PHP.

A continuación se detallan los tiempos de ejecución para esta fase:

Abril 2016						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

 Implementación - Iteración 5

Ilustración 111 - Tiempos de ejecución - Iteración 5 - Multilenguaje (1)

Mayo 2016						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
1	2 0h	3 Multienguaje (4h)	4 Multilenguaje (1,5h)	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

■ Implementación - Iteración 5

Ilustración 112 - Tiempos de ejecución - Iteración 5 - Multilenguaje (2)

5.4.3. Interfaz de usuario

En un principio se implementó una pequeña interfaz de usuario, que ha servido para ir realizando pruebas a medida que se iban implementando funcionalidades.

Ahora ha llegado el momento de convertir esa interfaz de usuario primitiva en una interfaz intuitiva y amigable para el usuario. En el apartado “4.1 Interfaz de usuario”, se realizó un boceto de cómo quedaría la interfaz finalmente.

A continuación se muestra el aspecto final la interfaz de usuario, mostrando algunas de las pantallas que permiten la navegabilidad desde que se inicia la aplicación hasta que se obtiene un resultado:

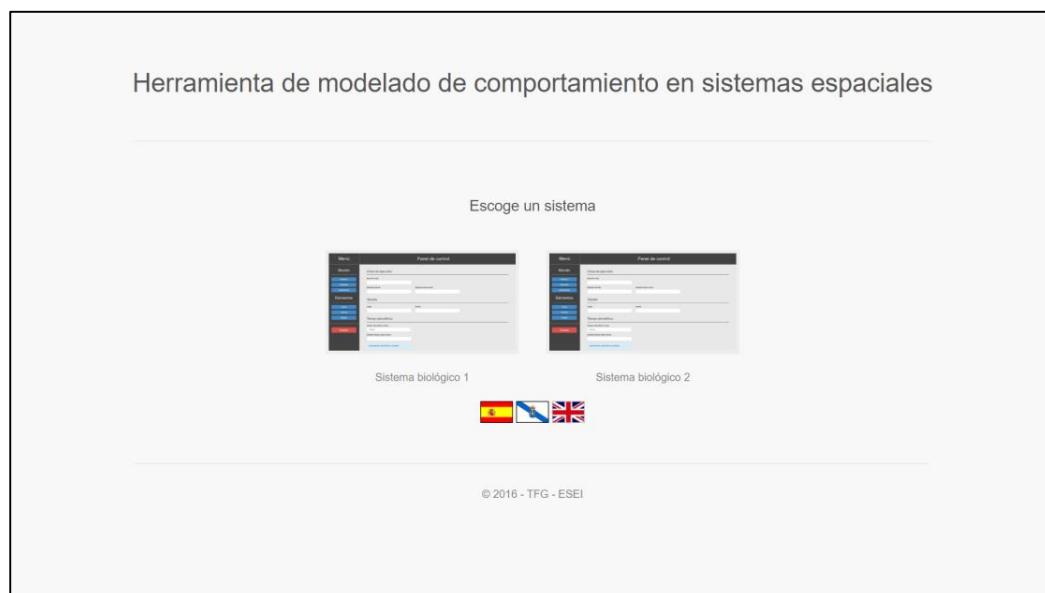


Ilustración 113 - Interfaz de usuario (1)



Ilustración 114 - Interfaz de usuario (2)



Ilustración 115 - Interfaz de usuario (3)

Menú	Panel de control
Mundo <ul style="list-style-type: none"> > Mundo < Elementos Restricciones Elementos <ul style="list-style-type: none"> Ciclos Acciones Rangos <input style="background-color: #e64a19; color: white; width: 100%; height: 30px; border: none; font-weight: bold; margin-top: 10px;" type="button" value="Empezar"/>	<p>Ciclos de ejecución</p> <p>Duración total: 1000</p> <p>Duración del día: 80 Duración de la noche: 20</p> <p>Tamaño</p> <p>Largo: 30 Ancho: 60</p> <p>Tiempo atmosférico</p> <p>Tiempo atmosférico inicial: soleado</p> <p>Cambiar tiempo cada (ciclos): 35</p> <p>0 para tiempo atmosférico constante</p>

Ilustración 116 - Interfaz de usuario (4)

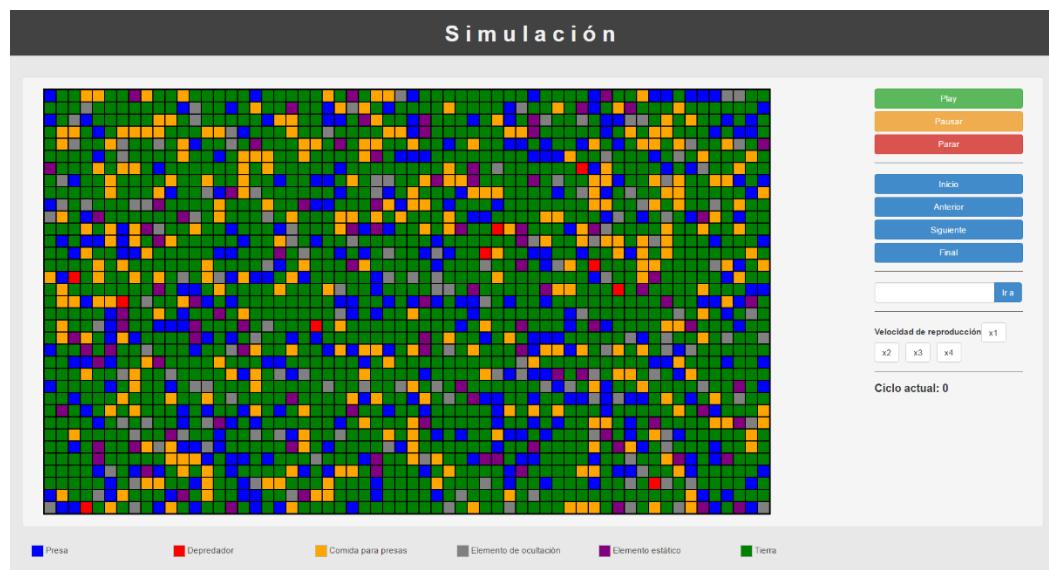


Ilustración 117 - Interfaz de usuario (5)

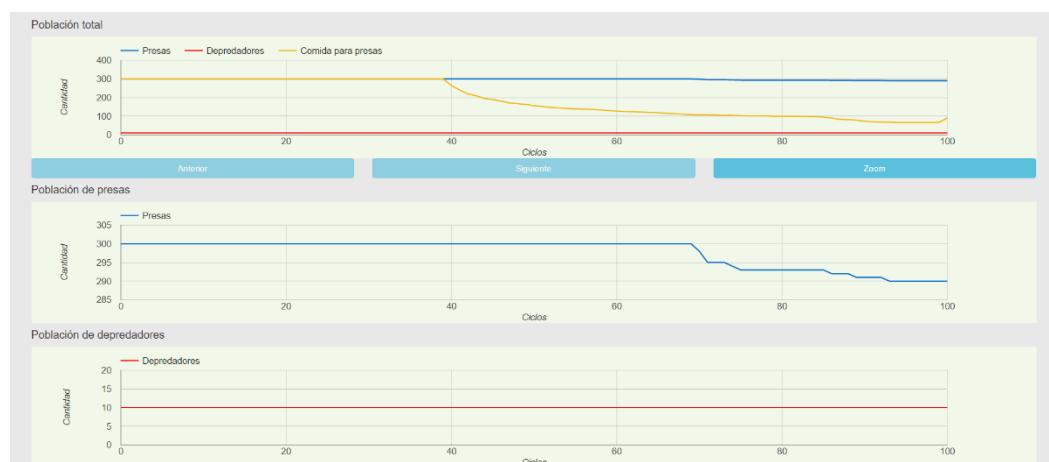


Ilustración 118 - Interfaz de usuario (6)



Ilustración 119 - Interfaz de usuario (7)

A continuación se detallan los tiempos de ejecución para esta fase:

Mayo 2016						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
1	2	3	4	5 Interfaz de usuario (6h)	6 Interfaz de usuario (6,5h)	7
8	9 0h	10 Interfaz de usuario (6,75h)	11 Interfaz de usuario (7h)	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

■ Implementación - Iteración 5

Ilustración 120 - Tiempos de ejecución - Iteración 5 - Interfaz de usuario

6. Pruebas

Una vez finalizada la implementación de la aplicación, para comprobar el correcto funcionamiento de la misma, se han realizado una serie de tests como se muestra a continuación:

- Elección del idioma ✓
- Elección de sistema ✓
- Carga de archivos ✓
- Entrada de datos en la pantalla de configuración ✓
- Manejo de errores en la pantalla de configuración ✓
- Descarga de archivos ✓
- Pruebas de ejecución ✓

Tamaño	Presas	Depredadores	Comida para presas	Elementos de ocultación	Obstáculos	Ciclos de ejecución	Días simulados	Tiempo de ejecución (s)	Tiempo de ejecución (min)	Memoria consumida (mb)	Test nº
30	60	300	10	300	150	100	10000	200	605,53	10,09	6 1
30	60	300	10	300	150	100	10000	200	207,53	3,46	6 2
30	60	300	10	300	150	100	10000	200	203,28	3,39	8 3
30	60	300	10	300	150	100	15000	300	332,98	5,55	6 4
30	60	300	10	300	150	100	15000	300	337,27	5,62	8 5
30	60	300	10	300	150	100	15000	300	352,45	5,87	8 6
30	60	300	10	300	150	100	15000	300	340,18	5,67	8 7
30	60	300	10	300	150	100	20000	400	493,03	8,22	16 8
30	60	300	15	300	150	100	20000	400	666,27	11,10	18 9
30	60	300	20	300	150	100	20000	400	665,34	11,09	16 10
30	60	300	25	300	150	100	20000	400	888,46	14,81	18 11

Ilustración 121 - Tiempos de ejecución – Pruebas

Para cada prueba realizada se ha extraído el gráfico de población de elementos, ya que es el principal aspecto que influye en los tiempos de ejecución:



Ilustración 122 - Tiempos de ejecución - Pruebas - Test 1



Ilustración 123 - Tiempos de ejecución - Pruebas - Test 2

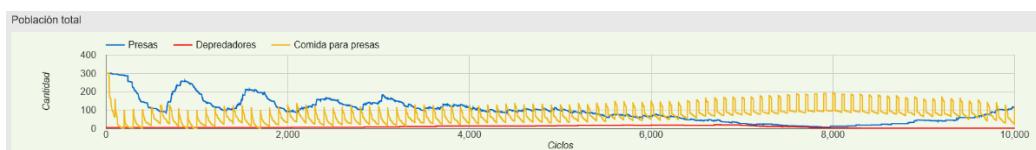


Ilustración 124 - Tiempos de ejecución - Pruebas - Test 3



Ilustración 125 - Tiempos de ejecución - Pruebas - Test 4

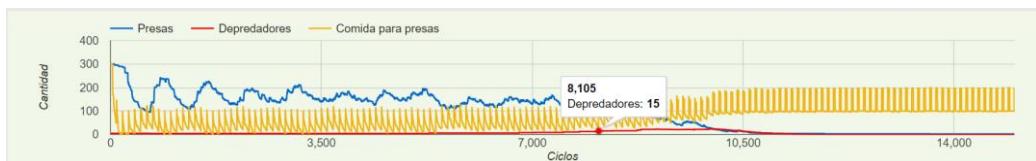


Ilustración 126 - Tiempos de ejecución - Pruebas - Test 5



Ilustración 127 - Tiempos de ejecución - Pruebas - Test 6



Ilustración 128 - Tiempos de ejecución - Pruebas - Test 7

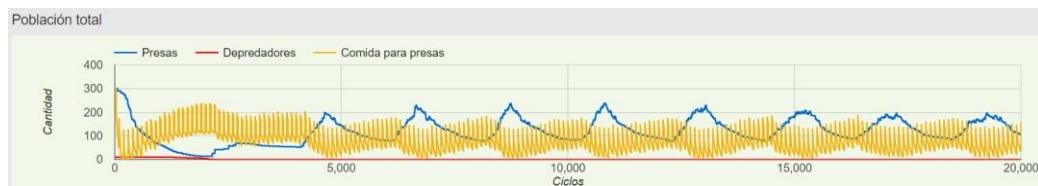


Ilustración 129 - Tiempos de ejecución - Pruebas - Test 8



Ilustración 130 - Tiempos de ejecución - Pruebas - Test 9



Ilustración 131 - Tiempos de ejecución - Pruebas - Test 10



Ilustración 132 - Tiempos de ejecución - Pruebas - Test 11

Mayo 2016						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Pruebas

Ilustración 133 - Tiempos de ejecución - Pruebas

7. Documentación final

A lo largo de todas las fases anteriores se ha ido generando la documentación correspondiente a cada una de ellas. En esta última fase, se completan los aspectos de la documentación que faltan por cubrir, se da un formato adecuado al documento y se redacta el manual de usuario.

A continuación se detallan los tiempos de ejecución para esta fase:

Mayo 2016						
domingo	lunes	martes	miércoles	jueves	viernes	sábado
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17 Documentación final (5,5h)	18 Documentación final (2,5h)	19 Documentación final (2,5h)	20 Documentación final (2h)	21
22	23	24	25	26	27	28
29	30	31				

■ Documentación final

Ilustración 134 - Tiempos de ejecución - Documentación final

III. Manual de usuario

1. Introducción

Simulation es una aplicación web que provee un entorno de simulación en el cuál se podrá comprobar el comportamiento de los individuos en un sistema configurado previamente. En este documento se aportan las nociones básicas sobre el funcionamiento de la aplicación así como su instalación.

2. Requisitos del sistema

Para ejecutar esta aplicación es necesario un ordenador con sistema Windows, Linux o Mac y tener instaladas las siguientes aplicaciones:

- Apache 2
- PHP 7
- Navegador web
- Conexión a Internet

Se recomienda un mínimo de 2GB de memoria RAM en el PC a utilizar.

3. Instalación

La instalación de la aplicación variará en base al sistema operativo escogido.

En primer lugar es necesario disponer del servidor web Apache 2 y de PHP 7. Una vez instalados, es necesario copiar el contenido del directorio /src proporcionado en el CD en el directorio web del servidor. En este punto, hay que hacer una diferenciación entre sistemas operativos:

- Windows:

Acceder al archivo *world.php* dentro de la carpeta /core y modificar las siguientes líneas:

- Comentar las líneas 363 y 396 y descomentar las líneas 364 y 397. En las líneas descomentadas es necesario cambiar la ruta del ejecutable de php (php.exe) por la ruta donde se encuentre dicho ejecutable en el equipo donde se esté realizando la instalación.

- Linux / Mac:

Acceder al archivo *world.php* dentro de la carpeta /core y modificar las siguientes líneas:

- Comentar las líneas 364 y 397 y descomentar las líneas 363 y 396.

- Para evitar cualquier tipo de error debido a los permisos de los ficheros, se aconseja otorgar permisos de lectura, escritura y ejecución a los tres grupos de usuarios. Puede que esta acción no sea necesaria, depende del sistema Linux/Mac sobre el que se vaya a ejecutar.

Posteriormente, es necesario realizar una pequeña modificación en la configuración del servidor. Para ello, se tiene que acceder al archivo *php.ini* del servidor y modificar los siguientes parámetros:

- Establecer el valor de *max_execution_time* a 0
- Establecer el valor de *max_input_time* a -1
- Establecer el valor de *memory_limit* a -1

Cuando esté todo configurado, ya se puede iniciar el servidor para ejecutar la aplicación.

4. Guía de uso

4.1. Interfaz de configuración

Una vez que el usuario inicia la aplicación se va a encontrar con la siguiente interfaz:

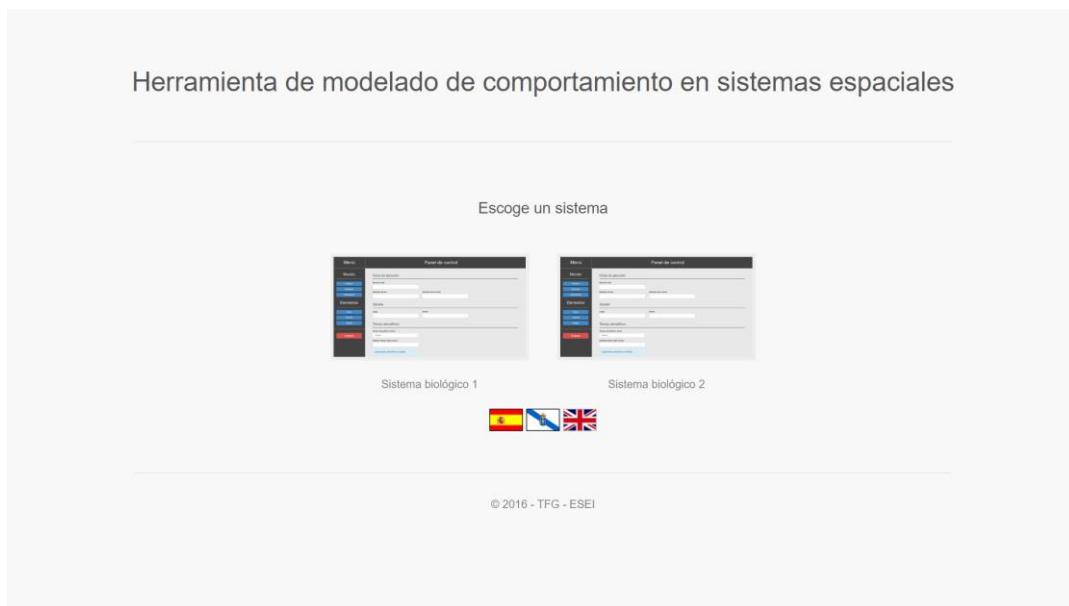


Ilustración 135 - Manual de usuario - Interfaz de configuración (1)

En primer lugar, se debe escoger el idioma en el que se quiere que se muestre la aplicación, dando opción entre tres idiomas diferentes: español, gallego e inglés. Para realizar esta elección es necesario seleccionar la bandera que se corresponda con el lenguaje elegido. Posteriormente, se da opción para escoger entre dos tipos de sistemas biológicos. Una vez escogido, se mostrará otra interfaz diferente. Cabe destacar que si no se ha escogido ningún idioma, se escogerá automáticamente el idioma por defecto.



Ilustración 136 - Manual de usuario - Interfaz de configuración (2)

En esta interfaz se dan también dos opciones: si se elige la primera, se podrá cargar un archivo de configuración; en el segundo caso, mostrará la interfaz de configuración directamente sin ningún dato relleno.

Tomando el primer camino, se llega a:



Ilustración 137 - Manual de usuario - Interfaz de configuración (3)

En este punto es necesario seleccionar el archivo de configuración que se obtendrá una vez que se ejecute la aplicación por primera vez. En caso de que el archivo seleccionado no sea el correcto, el icono se volverá de color rojo indicando que se ha producido un error y que es necesario seleccionar otro archivo.

Se han facilitado dos archivos de configuración en el directorio /src/resources/conf del CD, uno para cada opción.

Cuando se complete satisfactoriamente este paso, se accederá a la pantalla de configuración de parámetros. Puesto que se ha seleccionado un archivo, los campos de configuración estarán llenos. En el otro caso, accediendo directamente a esta interfaz sin seleccionar ningún archivo, los campos estarán vacíos. En ambos casos se puede introducir un valor en casa de que esté vacío o modificar el ya existente.

Menú	Panel de control
Mundo <input type="button" value="> Mundo <"/> <input type="button" value="Elementos"/> <input type="button" value="Restricciones"/> Elementos <input type="button" value="Ciclos"/> <input type="button" value="Acciones"/> <input type="button" value="Rangos"/> <input style="background-color: red; color: white;" type="button" value="Empezar"/>	<h3>Ciclos de ejecución</h3> <p>Duración total: <input type="text" value="1000"/></p> <p>Duración del día: <input type="text" value="80"/> Duración de la noche: <input type="text" value="20"/></p> <h3>Tamaño</h3> <p>Largo: <input type="text" value="30"/> Ancho: <input type="text" value="60"/></p> <h3>Tiempo atmosférico</h3> <p>Tiempo atmosférico inicial: <input type="text" value="soleado"/></p> <p>Cambiar tiempo cada (ciclos): <input type="text" value="35"/></p> <p><small>0 para tiempo atmosférico constante</small></p>

Ilustración 138 - Manual de usuario - Interfaz de configuración (4)

En la parte izquierda se muestra un menú con las diferentes categorías de parámetros. Seleccionando cada uno de ellos se accede a los diferentes parámetros, que se muestran en el resto de la pantalla. A continuación se explicará con detalle cada parámetro de configuración disponible:

- Seleccionando en el menú lateral la opción “Mundo”, se muestran los siguientes parámetros de configuración:

Menú	Panel de control
Mundo <input type="button" value="> Mundo <"/> <input type="button" value="Elementos"/> <input type="button" value="Restricciones"/> Elementos <input type="button" value="Ciclos"/> <input type="button" value="Acciones"/> <input type="button" value="Rangos"/> <input style="background-color: red; color: white;" type="button" value="Empezar"/>	<h3>Ciclos de ejecución</h3> <p>Duración total: <input type="text"/></p> <p>Duración del día: <input type="text"/> Duración de la noche: <input type="text"/></p> <h3>Tamaño</h3> <p>Largo: <input type="text"/> Ancho: <input type="text"/></p> <h3>Tiempo atmosférico</h3> <p>Tiempo atmosférico inicial: <input type="text" value="soleado"/></p> <p>Cambiar tiempo cada (ciclos): <input type="text"/></p> <p><small>0 para tiempo atmosférico constante</small></p>

Ilustración 139 - Menú - Opción "Mundo"

- **Ciclos de ejecución**
 - **Duración total:** indica el número de ciclos de ejecución totales.
 - **Duración del día:** indica el número de ciclos que dura el día, período comprendido desde que amanece hasta que anochece.
 - **Duración de la noche:** indica el número de ciclos que dura la noche, período comprendido desde que anochece hasta que amanece.
- **Tamaño**
 - **Largo y ancho:** indican el tamaño del mundo a crear.
- **Tiempo atmosférico**
 - **Tiempo atmosférico inicial:** permite seleccionar con qué tipo de tiempo atmosférico se empezará la ejecución (soleado, lluvioso, ventoso o neblinoso).
 - **Cambiar tiempo cada (ciclos):** indica cada cuántos ciclos se va modificar el estado del tiempo atmosférico.
- Seleccionando en el menú lateral la opción “Elementos”, se muestran los siguientes parámetros de configuración:

Menú	Panel de control		
Mundo <hr/> Mundo <hr/> > Elementos < <hr/> Restricciones Elementos <hr/> Cíclios <hr/> Acciones <hr/> Rangos Empezar	Cantidad inicial de elementos		
	Comida para presas	Elementos estáticos	Elementos de ocultación
	<input type="text"/>	<input type="text"/>	<input type="text"/>
	Presas	Depredadores	
	<input type="text"/>	<input type="text"/>	
Regeneración de comida para presas			
Cada (cíclos)	Cantidad	<input type="text"/>	
<small>0 para evitar la regeneración de comida para presas</small>			
Eliminación de comida para presas			
<small>Tiempo de vida de la comida para presas (cíclos)</small> <input type="text"/>			

Ilustración 140 - Menú – Opción “Elementos”

- **Cantidad inicial de elementos**
 - **Comida para presas:** indica la cantidad de elementos “comida para presas” inicial.
 - **Elementos estáticos:** indica la cantidad de elementos “elemento estático” inicial.
 - **Elementos de ocultación:** indica la cantidad de elementos “elemento de ocultación” inicial.
 - **Presas:** indica la cantidad de elementos “presa” inicial.
 - **Depredadores:** indica la cantidad de elementos “depredador” inicial.

- **Regeneración de comida para presas**
 - **Cada (ciclos):** indica cada cuántos ciclos se va a regenerar la comida para presas.
 - **Cantidad:** indica la cantidad de comida para presas que se regenera.
- **Eliminación de comida para presas**
 - **Tiempo de vida de la comida para presas (ciclos):** indica cuántos ciclos dura la comida para presas siempre y cuando no sea comida por una presa. Después de ese período, desaparece.
- Seleccionando en el menú lateral la opción “Restricciones”, se muestran los siguientes parámetros de configuración:

Menú	Panel de control
Mundo <ul style="list-style-type: none"> Mundo Elementos > Restricciones < Elementos <ul style="list-style-type: none"> Cíclios Acciones Rangos Empezar	<p>Comer</p> <p>Presas - Ciclos sin comer Depredadores - Ciclos sin comer</p> <p>Dormir</p> <p>Presas - Ciclos sin dormir Depredadores - Ciclos sin dormir</p> <p>Presas - Lugar obligatorio para dormir Tierra / Madriguera</p> <p>Reproducción</p> <p>Presas - Pueden reproducirse cada (ciclos) Presas - Cantidad máxima</p> <p>Depredadores - Pueden reproducirse cada (ciclos) Lobos - Cantidad máxima</p> <p>0 para evitar la reproducción de presas y depredadores</p> <p>No necesariamente se van a reproducir con estos parámetros. Necesitan encontrarse dos presas o</p>

Ilustración 141 - Menú - Opción "Restricciones"

- **Comer**
 - **Presas - Ciclos sin comer:** indica cuántos ciclos puede estar una presa sin comer. Una vez pasado ese período de tiempo, muere.
 - **Depredadores - Ciclos sin comer:** indica cuántos ciclos puede estar un depredador sin comer. Una vez pasado ese período de tiempo, muere.
- **Dormir**
 - **Presas - Ciclos sin dormir:** indica cuántos ciclos puede estar una presa sin dormir. Una vez pasado ese período de tiempo, muere.
 - **Depredadores - Ciclos sin dormir:** indica cuántos ciclos puede estar un depredador sin dormir. Una vez pasado ese período, muere.

- **Presas - Lugar obligatorio para dormir:** indica el lugar donde están obligadas a dormir las presas, ofreciendo la opción de dormir sólo en un elemento de ocultación o de dormir tanto en un elemento de ocultación como en un elemento “Tierra”, es decir, cualquier posición del mundo que no esté ocupada por nada.
- **Reproducción**
 - **Presas – Pueden reproducirse cada (ciclos):** indica cada cuántos ciclos se pueden reproducir las presas.
 - **Presas – Cantidad máxima:** indica la cantidad de hijos que puede tener una presa como máximo, variando desde 0 a dicha cantidad.
 - **Depredadores – Pueden reproducirse cada (ciclos):** indica cada cuántos ciclos se pueden reproducir los depredadores.
 - **Depredadores – Cantidad máxima:** indica la cantidad de hijos que puede tener un depredador como máximo, variando desde 0 a dicha cantidad.
- Seleccionando en el menú lateral la opción “Ciclos”, se muestran los siguientes parámetros de configuración:

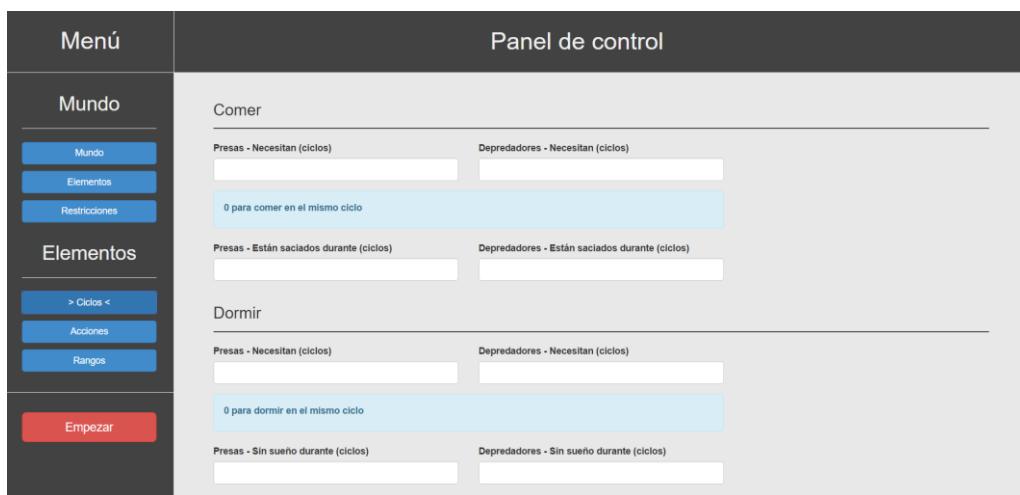


Ilustración 142 - Menú - Opción "Ciclos"

- **Comer**
 - **Presas – Necesitan (ciclos):** número de ciclos que necesitan las presas para comer un alimento.
 - **Depredadores – Necesitan (ciclos):** número de ciclos que necesitan los depredadores para comer un alimento.
 - **Presas – Están saciadas durante (ciclos):** número de ciclos donde las presas se encuentran saciadas, es decir, que no necesitan comer. Empiezan a contar una vez que acaban de comer.

- **Depredadores – Están saciados durante (ciclos):** número de ciclos donde los depredadores se encuentran saciados, es decir, que no necesitan comer. Empiezan a contar una vez que acaban de comer.
- **Dormir**
 - **Presas – Necesitan (ciclos):** número de ciclos que necesitan las presas para dormir.
 - **Depredadores – Necesitan (ciclos):** número de ciclos que necesitan los depredadores para dormir.
 - **Presas – Sin sueño durante (ciclos):** número de ciclos donde las presas no tienen sueño, es decir, que no necesitan dormir.
 - **Depredadores – Sin sueño durante (ciclos):** número de ciclos donde los depredadores no tienen sueño, es decir, no necesitan dormir.
- Seleccionando en el menú lateral la opción “Acciones”, se muestran los siguientes parámetros de configuración:

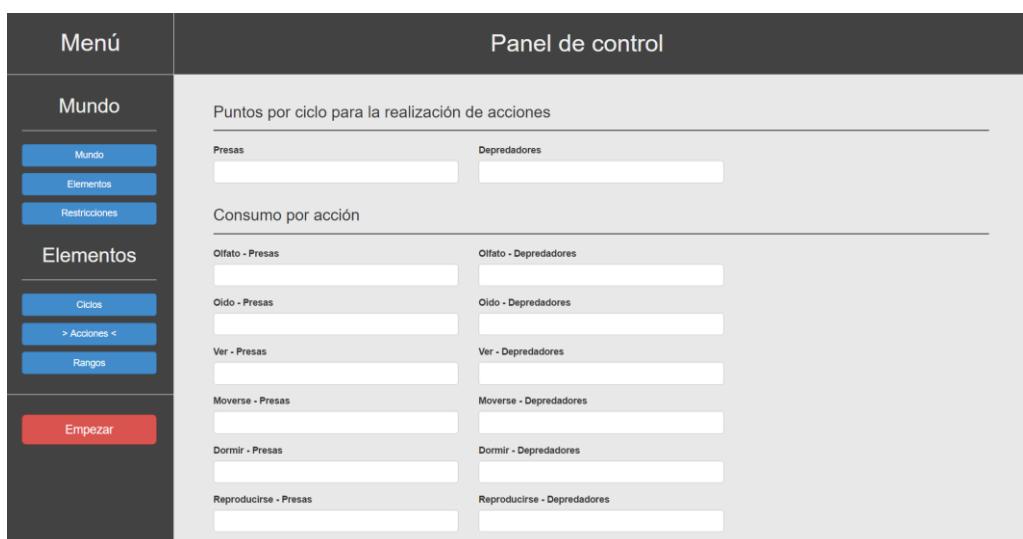


Ilustración 143 - Menú - Opción "Acciones"

- **Puntos por ciclo para la realización de acciones**
Tanto las presas como los depredadores tienen un sistema de puntos para realizar acciones. Cada acción tiene un coste de puntos asociado. Por tanto, durante un ciclo la realización de acciones viene determinada por la cantidad de puntos que tenga. Al comienzo de cada ciclo siempre se tiene la cantidad máxima de puntos establecida.
 - **Presas:** indica la cantidad de puntos totales por ciclo que tiene una presa para realizar acciones.

- **Depredadores:** indica la cantidad de puntos totales por ciclo que tiene un depredador para realizar acciones.
- **Consumo por acción**
 - **Olfato – Presas:** cantidad de puntos que consume la acción olfatear para cada presa.
 - **Olfato – Depredadores:** cantidad de puntos que consume la acción olfatear para cada depredador.
 - **Oído – Presas:** cantidad de puntos que consume la acción oír para cada presa.
 - **Oído – Depredadores:** cantidad de puntos que consume la acción oír para cada depredador.
 - **Ver – Presas:** cantidad de puntos que consume la acción ver para una presa.
 - **Ver – Depredadores:** cantidad de puntos que consume la acción ver para cada depredador.
 - **Moverse – Presas:** cantidad de puntos que consume la acción moverse para cada presa.
 - **Moverse – Depredadores:** cantidad de puntos que consume la acción moverse para cada depredador.
 - **Dormir – Presas:** cantidad de puntos que consume la acción dormir para cada presa.
 - **Dormir – Depredadores:** cantidad de puntos que consume la acción dormir para cada depredador.
 - **Reproducirse – Presas:** cantidad de puntos que consume la acción reproducirse para cada presa.
 - **Reproducirse – Depredadores:** cantidad de puntos que consume la acción reproducirse para cada depredador.
- Seleccionando en el menú lateral la opción “Rangos”, se muestran los siguientes parámetros de configuración:

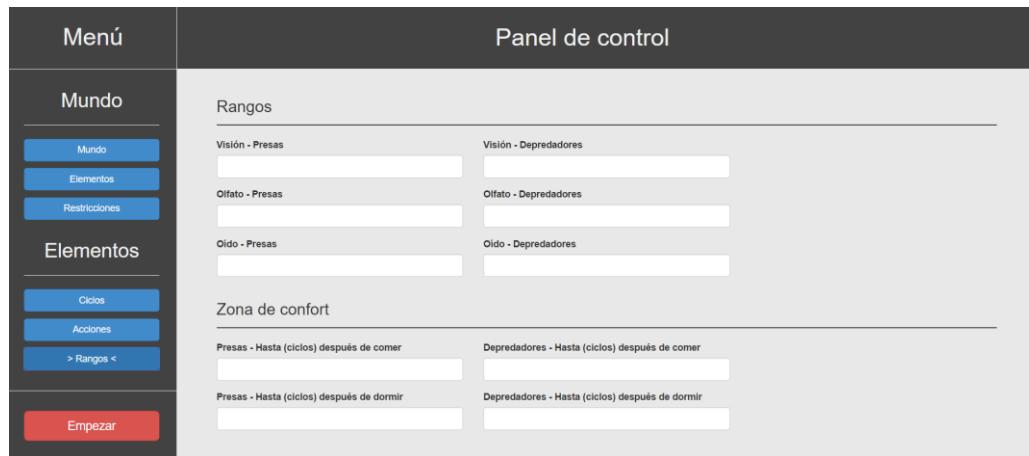


Ilustración 144 - Menú - Opción "Rangos"

■ **Rangos**

- **Visión – Presas:** indica el rango de visión de cada presa.
- **Visión – Depredadores:** indica el rango de visión de cada depredador.
- **Olfato – Presas:** indica el rango de olfato de cada presa.
- **Olfato – Depredadores:** indica el rango de olfato de cada depredador.
- **Oído – Presas:** indica el rango de oído de cada presa.
- **Oído – Depredadores:** indica el rango de oído de cada depredador.

■ **Zona de confort**

- **Presas – Hasta (ciclos) después de comer:** indica durante cuántos ciclos permanece cada presa en una zona de confort después de comer.
- **Depredadores – Hasta (ciclos) después de comer:** indica durante cuántos ciclos permanece cada depredador en una zona de confort después de comer.
- **Presas – Hasta (ciclos) después de dormir:** indica durante cuántos ciclos permanece cada presa en una zona de confort después de dormir.
- **Depredadores – Hasta (ciclos) después de dormir:** indica durante cuántos ciclos permanece cada depredador en una zona de confort después de dormir.

Nota: para que un elemento se encuentre en la zona de confort tienen que cumplirse las opciones anteriores.

Cabe destacar que todos estos parámetros son comunes a ambos tipos de sistemas. En el segundo sistema, hay una opción de configuración a mayores, que se refleja a continuación:

- Seleccionando en el menú lateral la opción “Elementos”, se muestran los siguientes parámetros de configuración:

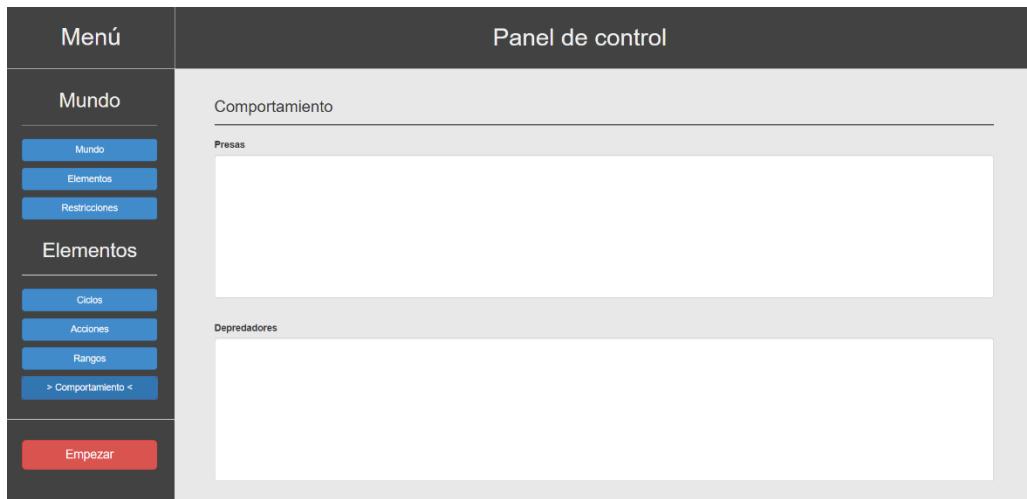


Ilustración 145 - Menú – Opción “Comportamiento”

▪ Comportamiento

- **Presas:** permite definir el comportamiento de las presas.
- **Depredadores:** permite definir el comportamiento de los depredadores.

Para definir dicho comportamiento es necesario tener conocimientos del lenguaje php, puesto que esta sección será necesario modelar el comportamiento mediante código. Para ello, se proporciona una API que contiene las funciones a las que puede acceder el usuario para la obtención de datos y las funciones que permiten la ejecución de las acciones de los elementos. Dicha API se encuentra disponible en el fichero api.php, proporcionado en el CD en el directorio /src/resources.

Una vez cubiertos todos los campos, solo queda ejecutar la aplicación. Para ello, se selecciona el botón “Empezar” en el menú izquierdo.

Durante la ejecución, se muestra una pantalla de carga que finalizará en el instante que la ejecución se complete para dar paso a la interfaz de visualización. Este proceso puede llegar a durar varios minutos, dependiendo de los parámetros de configuración escogidos.

4.2. Interfaz de visualización

Una vez que la ejecución de la aplicación se ha completado, se muestra una pantalla de visualización, donde se pueden observar los resultados de la ejecución.

En la parte superior se encuentra el entorno visual del mundo. A la derecha del mismo, están los botones de control de la interfaz, con los que se puede realizar un desplazamiento temporal por los distintos estados del mundo. Se pueden realizar las acciones:

- Play: inicia la reproducción por los distintos estados del mundo empezando en el primer estado.
- Pausar: pausa la reproducción.

- Parar: para la reproducción volviendo al primer estado.
- Inicio: se sitúa en el primer estado.
- Siguiente: avanza un estado.
- Anterior: retrocede un estado.
- Fin: se desplaza hasta el último estado.
- Ir a: permite ir a un estado en concreto.
- x_1, x_2, x_3, x_4 : permite modificar la velocidad de reproducción.

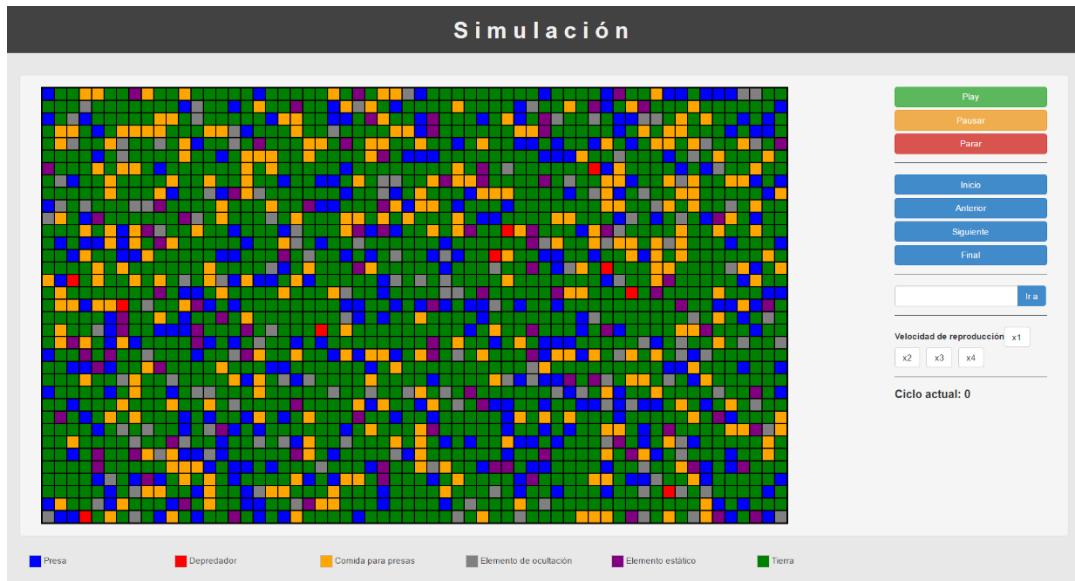


Ilustración 146 - Manual de usuario - Interfaz de visualización (1)

A continuación se encuentran las gráficas que permiten entender qué ha ocurrido durante la ejecución del mundo. Mediante los botones anterior y siguiente se permite desplazarse temporalmente a lo largo de la gráfica. El botón zoom permite acercarse en un instante concreto.



Ilustración 147 - Manual de usuario - Interfaz de visualización (2)

Finalmente, se permite la opción de descargar varios ficheros y la opción de volver a realizar otra configuración y posterior ejecución.



Ilustración 148 - Manual de usuario - Interfaz de visualización (3)

IV. Bibliografía

1. Artículos

- <http://www.genbetadev.com/genbeta-dev/te-acuerdas-de-el-juego-de-la-vida>
 - Artículo utilizado en la investigación acerca de El juego de la vida.
- http://www.cienciamexicana.mx/divulgacion/conoces-juego-vida-conway_2014103100163.html
 - Artículo utilizado en la investigación acerca de El juego de la vida.
- <http://ncase.me/simulating/>
 - Artículo utilizado en la investigación acerca de la aplicación web creada por Ncase, un simulador de incendios forestales.

2. Libros

- El gran libro de HTML5, CSS3 y Javascript, Juan Diego Gauchat (2012).
 - Libro utilizado para realizar consultas durante la fase de implementación sobre conceptos de html, css y javascript.

3. Documentaciones técnicas

- <https://developer.mozilla.org/es/docs/Web/JavaScript>
 - Documentación consultada durante la implementación sobre javascript.
- <https://secure.php.net/manual/es/index.php>
 - Documentación consultada durante la implementación sobre php.
- <http://getbootstrap.com/>
 - Documentación consultada durante la implementación sobre el framework bootstrap.
- <https://developers.google.com/chart/>
 - Documentación consultada durante la implementación sobre las gráficas de google.
- <https://httpd.apache.org/>
 - Documentación consultada durante la implementación para la configuración de apache.

4. Otros

- <http://stackoverflow.com/>
 - Foro consultado durante la implementación para resolver cualquier tipo de problema de programación.

- <http://w3schools.com/>
 - Página web que contiene tutoriales de html, css, javascript, jquery, php, sql y bootstrap, utilizada durante la fase de implementación.
- <https://es.wikipedia.org>
 - Enciclopedia online en la que se puede consultar cualquier tipo de tema.