



*Facultad
de
Ciencias*

**SISTEMA SOFTWARE DE FICCIÓN
INTERACTIVA COMO FÓRMULA DE
INCLUSIÓN DIGITAL**
**(Interactive fiction software system as an
approach for digital inclusion)**

Trabajo de Fin de Grado
para acceder al

GRADO EN INGENIERÍA INFORMÁTICA

Autor: José Francisco Martín Lisaso

Director: Andrés Iglesias Prieto

Co-Director: Akemi Gálvez Tomida

Junio - 2019

Agradecimientos

Hace alrededor de tres décadas que mi madre renunció a la oportunidad de cursar estudios universitarios y tomó junto a mi padre la decisión de formar una familia en su lugar. Este trabajo es para ellos, por aquella elección de hace más de treinta años.

Y para todas las demás personas que habéis perseverado hasta el extremo de conducirme aquí a pesar de mis grandes esfuerzos en contra.

Gracias.

Resumen

Los computadores se han convertido en una importante fuente de entretenimiento y en un medio de expresión más en la sociedad de hoy en día. Sin embargo, se ha observado que las desigualdades en renta, educación y las provocadas por cuestiones de ingresos, cultura, edad, género o diversidad funcional están negando el acceso efectivo de grupos e individuos a productos culturales generados en el contexto de las tecnologías de la información. Este proyecto parte del objetivo de combatir dichas dinámicas de discriminación y exclusión digital a través del propio entretenimiento electrónico.

Para conseguirlo se opta por trabajar con el género de la ficción interactiva; un tipo de videojuego basado fundamentalmente en texto prometedor de cara a facilitar la accesibilidad de personas con diversidad funcional asociada a aspectos de carácter visual, auditivo y/o motriz, privadas habitualmente de gran parte de la oferta de entretenimiento por computador. Se consideran las principales barreras de accesibilidad con las que se puede encontrar un usuario de este tipo de videojuego y se diseña e implementa un conjunto de módulos para uno de los sistemas de autoría de ficción interactiva más populares en la actualidad. A través de estos módulos se pretende extender las funcionalidades del sistema seleccionado y paliar en cierta medida las barreras apreciadas.

Por último, se crean tres ficciones interactivas que sirvan a modo de ejemplo de los módulos anteriores y se publica y publica libremente en Internet todo el software desarrollado a fin de fomentar que otros autores creen nuevas ficciones interactivas accesibles.

Palabras clave: *Diversidad funcional; Inclusión digital; Accesibilidad; Entretenimiento electrónico; Ficción interactiva.*

Abstract

Computers have become an important source of entertainment and another means of expression in today's society. However, it has been observed that inequalities in wealth, education and those caused by issues of income, culture, age, gender or functional diversity are denying groups and individuals effective access to cultural products generated in the context of information technologies. This project is aimed on the objective of countering these dynamics of digital discrimination and exclusion through electronic entertainment itself.

The genre of interactive fiction is used as a tool to achieve this goal. Interactive fiction is a type of video game based fundamentally on text that *a priori* can facilitate accessibility to people with functional diversity associated with visual, auditory and/or motor aspects, who are usually deprived of most part of computer entertainment offerings. The main accessibility barriers with which a user of this type of video game usually face are considered, and a set of modules is designed and implemented for one of the most popular interactive fiction authoring systems at present. The aim of these modules is to extend the functionalities of the selected system and alleviate to a certain extent the observed barriers.

Finally, three interactive fictions are created to serve as examples of the previous modules. All developed software is advertised and freely published on the Internet in order to encourage other authors to create new accessible interactive fictions.

Keywords: *Functional diversity; Digital Inclusion; Computer accessibility; Electronic entertainment; Interactive fiction.*

Índice general

1. Introducción	1
1.1. Desigualdad en la sociedad de la información	1
1.2. El computador como fuente de entretenimiento y medio de expresión	3
1.3. Objetivos del proyecto	5
1.4. Procedimiento y estructuración de la memoria	5
2. La ficción interactiva	7
2.1. Definición de <i>ficción interactiva</i>	7
2.2. Sistemas de autoría	9
2.2.1. Criterios para la selección de un sistema de autoría	9
2.2.2. Sistemas estudiados	10
2.3. El sistema de autoría Inform	12
2.3.1. Inform 6	12
2.3.2. Inform 7	12
2.3.3. Z-machine y Glulx	13
3. Extensiones al sistema Inform	15
3.1. Relación de barreras abordadas	15
3.2. Ciclo de vida del proyecto	16
3.3. Especificación de requisitos	17
3.4. Diseño de alto nivel	19
3.4.1. Arrays	19

3.4.2. Capacities	20
3.4.3. Choice sets	20
3.4.4. GWindows GUI	21
3.4.5. Hyperlinks	21
3.4.6. Parsing preprocessor	22
3.4.7. Spanish extended grammar	23
3.4.8. Spanish library messages	23
3.4.9. Text styles	24
3.5. Diseño detallado	25
4. Tres ficciones interactivas	29
4.1. Ciclo de vida adoptado en esta fase	29
4.2. Diseño de la estructura narrativa	30
4.2.1. Estructura de <i>La pequeña cerillera</i>	31
4.2.2. Estructura de <i>Historia de Grisóstomo y Marcela</i>	31
4.2.3. Estructura de <i>La zona</i>	33
4.3. <i>Script</i> de pre-compilación	34
4.4. Sistema de sintaxis simplificada	35
4.4.1. Hipervínculos que lanzan la sintaxis simplificada	37
4.5. Sistema de ayuda	37
5. Conclusiones y trabajo futuro	39
5.1. Grado de consecución de objetivos	39
5.1.1. <i>Rayuela de Arena I</i>	41
5.1.2. <i>Jam Cultura Abierta</i>	42
5.1.3. Exposición: Ficción interactiva y juegos narrativos	43
5.2. Líneas de trabajo futuro	43

Índice de figuras

1.1.	<i>Colossal Cave Adventure</i> ejecutándose en un computador PDP-11/34, con salida a través de una consola VT100 (Wikimedia Commons).	4
2.1.	Captura de pantalla de <i>Photopia</i> (A. Cadre, 1998). Creada con Inform 6.	13
3.1.	Diagrama del “ <i>modelo en cascada</i> ”, por P. Kemp y P. Smith (2010).	16
3.2.	Distribución de ventanas con GWindows GUI.	22
3.3.	Jerarquía de ventanas de la interfaz definida en GWindows GUI.	23
3.4.	Diagrama en detalle del módulo <i>Choice sets</i> (3.4.3).	25
3.5.	Diagrama en detalle del módulo <i>Spanish library messages</i> (3.4.8).	26
3.6.	Diagrama en detalle del resto de módulos definidos en 3.4.	27
4.1.	Portadas de las tres ficciones interactivas desarrolladas.	29
4.2.	DFA con la estructura de <i>La pequeña cerillera</i>	32
4.3.	DFA con la estructura de <i>Historia de Grisóstomo y Marcela</i>	32
4.4.	DFA con la estructura de <i>La zona</i>	33
4.5.	Implementación por defecto con la descripción de un objeto.	35
4.6.	Descripción de un objeto usando el <i>script</i> de pre-compilación.	35
4.7.	Possible implementación de la acción <code>##DefaultAction</code>	36
4.8.	Possible implementación de la rutina <code>UnknownVerb</code>	36
4.9.	Possible implementación de la rutina <code>PrintVerb</code>	37
4.10.	Ejemplo de definición del comando por defecto para los hipervínculos.	37
4.11.	Captura de pantalla de <i>La pequeña cerillera</i> (2018).	38

IV TRABAJO DE FIN DE GRADO

4.12. Captura de pantalla de <i>Historia de Grisóstomo y Marcela</i> (2018).	38
5.1. <i>Historia de Grisóstomo y Marcela</i> en la exposición de <i>Jam Cultura Abierta</i> . .	42
5.2. <i>La pequeña cerillera</i> en la exposición <i>Ficción interactiva y juegos narrativos</i>	43

Índice de cuadros

2.1. Comparación de sistemas de autoría.	11
3.1. Requisitos funcionales asociados con la accesibilidad.	17
3.2. Requisitos funcionales asociados a otras extensiones de la librería Inform.	18
3.3. Requisitos funcionales asociados a otras extensiones de la librería Inform (continuación).	19
3.4. Requisitos no funcionales.	20
3.5. Estilos de Text styles y sus correspondencias en <i>Glulx</i> y <i>Z-Machine</i>	24

1 | Introducción

*Computers are mostly used against people instead of for people, used to control people instead of to free them.
Time to change all that.*

«Los computadores se usan ante todo en contra de las personas en lugar de a favor de las personas, se usan para controlar a las personas en lugar de para liberarlas. Es hora de cambiar todo eso.»

—PEOPLE'S COMPUTER COMPANY (oct. 1972)

En este capítulo se describen algunas situaciones de desigualdad que se manifiestan en la sociedad de la información y se discute la idoneidad de utilizar software de entretenimiento como posible fórmula a través de la cual fomentar la inclusión digital. En última instancia se ofrece una relación de los objetivos planteados para el presente trabajo y la estructura general de la memoria.

1.1 Desigualdad en la sociedad de la información

Vivimos en un mundo de grandes desigualdades. El acceso a bienes, servicios y recursos asociados con la ciudadanía varía sustancialmente entre grupos e individuos distintos. La manifestación de estos patrones de diferenciación social es recurrente, además, y se puede observar en múltiples ámbitos y niveles. En la esfera económica, por ejemplo, un informe de 2018 del Banco Mundial[Lange et al., 2018] evidencia cómo la distribución de la riqueza en el mundo varía dramáticamente entre países pobres y países ricos¹. Si se reduce la escala y se analiza la realidad de un único país es fácil constatar cómo se repiten dinámicas similares: en España, desde 2014 «el 1 % más rico [de la población] concentra casi 25 de cada 100 euros de riqueza neta (24,4 % del total), mientras que el 50 % más pobre se tiene que repartir 7 euros de cada

¹Según el informe citado, la renta per cápita media en un país de la OCDE era de 708,389 US\$ en 2014, mientras que en otros países con ingresos medio-altos la renta per cápita media ese mismo año era de 112,798 US\$, en países con ingresos medio-bajos apenas de 25,948 US\$, y en países con bajos ingresos tan sólo de 13,629 US\$.

100»[Marcos and Macías, 2019]. Por otra parte, si bien el marco económico es útil para establecer comparaciones, la desigualdad es un fenómeno complejo que trasciende una cuestión de renta —no es casual, al fin y al cabo, que las listas de personas más ricas del planeta estén lideradas año tras año por varones de raza blanca[Forbes, 2019], o que la pobreza mundial se concentre principalmente en el África sub-sahariano—. El abanico de oportunidades de las que puede disfrutar una persona está íntimamente ligado con el constructo social bajo el que se la etiquete; aspectos como el género, la raza, la nacionalidad, la religión o la diversidad funcional determinan el grado en que un individuo puede acceder a los recursos sociales, políticos, económicos y culturales de su entorno, por delante muchas veces de su potencial o de los méritos que haya demostrado.

En este contexto de desigualdad, los computadores pueden erigirse como herramientas prometedoras a través de las que combatir la diferenciación social. Dicha consideración viene motivada fundamentalmente por las siguientes razones:

- En primer lugar su **polivalencia**; el computador como máquina programable de propósito general. Múltiples grupos e individuos diferentes presentan problemáticas y necesidades diferentes. Los computadores permiten abordar y resolver un conjunto sumamente amplio de problemas distintos.
- Su **capacidad multimedia**. Los computadores pueden trabajar sobre múltiples medios —texto, imagen, sonido,...—, de tal manera que pueden llegar a ser operados aún por personas con dificultades para acceder a un medio concreto, especialmente en el caso de dificultades vinculadas a la diversidad funcional que pueden condicionar la movilidad y/o percepción sensorial de estas personas.
- Por último su **ubicuidad**. En 2018 se estimaba que alrededor del 48,3 % de los hogares del mundo contaban con acceso directo a un computador personal, una estadística que prácticamente se había doblado en menos de dos décadas y que continúa creciendo[Statista, 2018], junto con el aumento igualmente progresivo de usuarios de dispositivos móviles[Statista, 2016].

A pesar de estas puntualizaciones se ha observado, sin embargo, que la paulatina democratización en el acceso a los computadores y la generalización del uso de las tecnologías de la información entre la población, lejos de atenuar los fenómenos de diferenciación y exclusión social, los acentúan: «existe un consenso entre académicos, grupos sin ánimo de lucro y activistas en defensa de las personas en situación de pobreza de que la exclusión digital (si bien no se enmarca habitualmente de este modo) está marginando aún más a individuos y comunidades ya oprimidos y privados de derechos. Las desigualdades en ingresos, educación y las provocadas por las diferencias de raza, cultura, edad, género y discapacidad no sólo se están transfiriendo a la sociedad de la información, sino que están reforzando la diferenciación social y la polarización de los grupos»[Parsons and Hick, 2009]. Esta creciente brecha digital, tal y como señalan igualmente Parsons y Hick (2009), no se limita a las disparidades originadas entre aquellos que pueden acceder físicamente a un computador y aquellos que no pueden hacerlo. No basta con tener acceso a las tecnologías de la información, sino que resulta necesario disponer también de las herramientas para utilizar

estas tecnologías efectivamente. A pesar de esta necesidad evidente, las dinámicas de discriminación y segregación están negando arbitrariamente el acceso a ciertos grupos de personas bien a los computadores, bien a los conocimientos y destrezas necesarios para hacer un uso efectivo de ellos —así lo reflejan, por ejemplo, estudios acerca de la discriminación digital motivada por las diferencias raciales en los Estados Unidos[Alvarez, 2003]—. Es decir, los constructos sociales bajo los que se etiqueta a una persona determinan en gran parte, una vez más, el grado en que ésta puede sacar o no provecho de las tecnologías de la información.

Con el presente trabajo se pretende verificar si, a pesar de las dinámicas discriminatorias observadas, el computador puede ser realmente un recurso útil de inclusión social más. Para conseguirlo se propone un acercamiento desde el dominio del entretenimiento digital; se plantea la creación de un conjunto de utilidades software a través de las que se facilite el desarrollo de un tipo de videojuegos conocidos como *ficciones interactivas*, fijando especial atención en mejorar la accesibilidad. La pretensión es que aquellas ficciones interactivas construidas con ayuda de estas utilidades puedan ser disfrutadas con la menor cantidad de barreras posibles tanto por el consumidor habitual de videojuegos como por un amplio espectro de personas con diversidad funcional asociada a cuestiones de carácter visual, auditivo y/o motriz, privadas habitualmente de gran parte de la oferta de entretenimiento por computador, y contribuir con ello a reducir —siquiera mínimamente— la brecha digital.

1.2 El computador como fuente de entretenimiento y medio de expresión

La elección del dominio del entretenimiento digital responde, más allá de filias personales, a dos cuestiones clave. Por una parte es patente el interés que despiertan los videojuegos y el impacto que tienen en el conjunto de la sociedad de hoy en día: se trata del «principal motor del entretenimiento global y representa una industria que ha sido capaz de generar 134.900 millones de dólares en 2018, creciendo un 10,9 %»[DEV, 2018b]. Atendiendo a la realidad concreta de nuestro país: «el sector productor de videojuegos españoles facturó en 2017 713 millones de euros, un 15,6 % más que 2016. De la misma manera, incrementó su plantilla en 2017 en un 16,5 %, alcanzando los 6.337 profesionales»[DEV, 2018b]. Tal actividad económica justifica por sí sola el interés por abrir la industria a un mayor grupo potencial de consumidores a los que, por razones de diversidad funcional y al formar parte de colectivos minoritarios y relegados a un segundo plano, habitualmente se les niega la oportunidad de disfrutar de esta clase de producto.

Múltiples movimientos recientes de los diferentes actores que conforman la industria del entretenimiento digital demuestran una tendencia en este sentido: desde la aparición de conferencias y eventos dedicados a la cuestión de la accesibilidad en los videojuegos, como la *Gaming Accessibility Conference*, promovida por IGDA-GASIG (International Game Developers Association - Game Accessibility Special Interest



Figura 1.1: *Colossal Cave Adventure* ejecutándose en un computador PDP-11/34, con salida a través de una consola VT100 (Wikimedia Commons).

Group)[GAconf, 2019]; el surgimiento de portales de accesibilidad como el publicado por la multinacional Electronic Arts Inc.[EA, 2018] o las *Game Accessibility Guidelines*[GAG, 2019] («pautas de accesibilidad para juegos»); a, sin pretender ser exhaustivos, la aparición de prensa especializada en el análisis de videojuegos desde una perspectiva de la accesibilidad como es por ejemplo el caso de DAGERS[Straub et al., 2019].

La segunda motivación tiene más que ver con una cuestión que parece formar parte de la naturaleza humana: a lo largo de su historia, el ser humano ha ido aprovechándose de diferentes invenciones y avances tecnológicos para encontrar nuevas vías a través de las que expresarse y contar historias. Los ejemplos son muchos y variados: desde la invención de la escritura y el surgimiento, con ella, de la literatura; a avances en los métodos para distribuir pintura al óleo que terminan dando pie a la aparición de nuevos movimientos pictóricos como el expresionismo; sistemas de telecomunicaciones por medio de ondas de radio que propician la emisión de dramatizaciones radiofónicas²; o invenciones como el cinematógrafo que prácticamente desde un primer momento es utilizado para crear narrativas ficticias³, por citar algunos. Con los computadores no ha sido diferente: si bien sus primeras aplicaciones durante los años 1950 comprendían cuestiones como «el cálculo de nóminas

²Entre las dramatizaciones radiofónicas la más famosa probablemente sea: *La guerra de los mundos* (1938) de Orson Welles.

³El primer film argumental de la historia: *El regador regado* (1895) de Louis Lumière, fue una de las primeras proyecciones con las que se presentó el invento del cinematógrafo en París[Cousins, 2019].

en empresas, registro de ventas, análisis del rendimiento de ventas y otras cuestiones empresariales»[THOCP, 2013], antes ya incluso de la explosión del computador personal y su adopción por parte del público general, hicieron su aparición las primeras narraciones digitales⁴. El académico Walter R. Fisher resume de un modo muy elocuente esta singularidad de nuestra especie: «se han propuesto muchas metáforas para representar la naturaleza fundamental de los seres humanos: *homo faber*, *homo economicus*, *homo politicus*, *homo sociologicus*, “hombre psicológico”, “hombre eclesiástico”, *homo sapiens* y, por supuesto, “hombre racional”. Yo propongo ahora añadir *homo narrans* a la lista»[Fisher, 1984]. El presente trabajo se adhiere a la visión que tiene Fisher sobre el ser humano y abre la oportunidad de indagar en la exploración y la experimentación en el campo aún joven de la narrativa electrónica.

1.3 Objetivos del proyecto

Este TFG persigue un doble objetivo:

1. Por un lado, diseñar e implementar un conjunto de herramientas software dedicadas a mejorar la accesibilidad de las obras de ficción interactiva y que puedan ser reutilizadas por otros programadores a fin de facilitar el desarrollo de nuevas obras con las que fomentar la inclusión digital.
2. Por otro lado, diseñar e implementar un conjunto de obras de ficción interactiva con el foco puesto en mejorar la accesibilidad a este género literario y que sirvan, en primer lugar, como ejemplo y demostración de uso de las herramientas generadas en el punto anterior, y que permitan además experimentar con las posibilidades narrativas del computador.

1.4 Procedimiento y estructuración de la memoria

A continuación se ofrece una visión general del trabajo desarrollado:

1. **Documentación y estudio de los conceptos fundamentales asociados al medio de la ficción interactiva.** Fase dedicada a la investigación de literatura existente sobre este tipo de software, su evolución histórica y familiarización con las obras más destacadas, así como las tendencias actuales y la previsible evolución del medio. A lo largo del capítulo 2 se detalla la definición y caracterización del género de la ficción interactiva tal y como se ha estudiado durante este proceso.

⁴Tal es el caso, por ejemplo de *Colossal Cave Adventure*, capturado en la figura 1.1. Este videojuego de W. Crowther y D. Woods desarrollado entre 1976 y 1977 en un computador *mainframe* PDP-10, además de ser una de las primeras obras narrativas por computador es, de hecho, la primera ficción interactiva

2. **Documentación y análisis de las principales herramientas software actuales utilizadas por los autores de ficción interactiva.** Se estudian las ventajas y desventajas de los principales sistemas de autoría utilizados por las comunidades de autores de habla inglesa y española —el cuadro 2.1 del capítulo 2 muestra una comparativa de los sistemas estudiados—. Durante esta fase se aprende a utilizar la herramienta *Inform 6* seleccionada.
3. **Especificación, diseño e implementación de módulos para extender las funcionalidades del sistema de autoría Inform 6.** Se desarrolla un conjunto de extensiones a Inform 6, ideadas principalmente para mejorar la accesibilidad de las obras creadas con este sistema, y se ponen a disposición de otros autores libremente en Internet. El capítulo 3 detalla las especificaciones y el diseño del conjunto de módulos desarrollados durante esta fase.
4. **Desarrollo de obras que sirvan como demostración y ejemplo de las herramientas creadas y difusión entre las comunidades de autores de ficción interactiva.** En el capítulo 4 se describe el conjunto de tres obras creadas a modo de ejemplo de las herramientas construidas previamente. A fin de dar mayor visibilidad al trabajo realizado, además de publicar con una licencia libre en Internet tanto los ejecutables como el código fuente de estas ficciones interactivas, se participa en varios eventos y competiciones de videojuegos y se presentan dos de las obras en exposiciones públicas en Madrid y Granada.
5. En el capítulo 5, por último, se detallan una serie de **conclusiones** extraídas tras la finalización del trabajo y la experiencia obtenida durante su desarrollo y la participación en las exposiciones citadas previamente. Así mismo, se apuntan algunas **Líneas de trabajo abiertas para un futuro**.

2 | La ficción interactiva

YOU ARE STANDING AT THE END OF A ROAD BEFORE
A SMALL BRICK BUILDING.

«ESTÁS AL FINAL DE UN CAMINO, DELANTE DE UNA
PEQUEÑA CONSTRUCCIÓN DE LADRILLO.»

—COLOSSAL CAVE ADVENTURE
(William Crowther y Don Woods, 1976 – 77)

Antes de iniciar el desarrollo del software se intenta definir con precisión el género de la ficción interactiva y se analizan las principales herramientas actuales a disposición de los autores para crear obras de este tipo. En la sección 2.3 se presenta el sistema de autoría Inform con el que se desarrollarán las próximas fases del trabajo.

2.1 Definición de *ficción interactiva*

Coloquialmente se utiliza el término **ficción interactiva** (traducción directa del inglés *interactive fiction*) para referirse a un amplio y heterogéneo grupo de obras de carácter interactivo que suelen tener en común un preponderante componente narrativo. Bajo esta clasificación es habitual encontrar colocadas propuestas de lo más diversas; desde librojuegos (físicos o digitales¹) a hiperficciones, cómics interactivos, novelas visuales (*visual novels*), aventuras gráficas, u otra clase de videojuegos. Incluso nuevas propuestas de ficción televisiva o cinematográfica con componente interactivo pueden llegar a ajustarse a esta etiqueta², al menos tal y como es entendida informalmente.

El término *interactive fiction*, sin embargo, fue acuñado por la compañía de software Infocom a principios de la década de 1980[Nelson, 2001] para hacer referencia al tipo muy concreto de obra que comercializaba y que en español se ha identificado

¹Entre los librojuegos, los más populares posiblemente sean aquellos del tipo *Elige Tu Propia Aventura*, o *CYOA*, por sus siglas en inglés.

²Como por ejemplo: *Black Mirror: Bandersnatch* (2018) dirigida por D. Slade y producida/distribuida por la compañía Netflix.

tradicionalmente como juego de texto o aventura conversacional. Se incide en diferenciar aquí *ficción interactiva* de *juego/aventura conversacional* debido a que dentro de este género es perfectamente posible minimizar o renunciar por completo al aspecto lúdico y perseguir en su lugar una vertiente más literaria, por una parte, y porque no toda trama de ficción interactiva tiene por qué adoptar forma de aventura; la obra *Photopia* (A. Cadre, 1998), capturada en la figura 2.1, es un ejemplo paradigmático de ficción interactiva que no se ajusta a los parámetros clásicos de las aventuras conversacionales.

De esta manera, acotando el término a su sentido más estricto —aquel empleado por Infocom—, la definición de ficción interactiva se limitaría a un **género literario sobre soporte electrónico que ofrece al usuario la posibilidad de explorar un mundo simulado y producir cambios en este entorno y en el propio progreso de la narración, por medio de la introducción de sencillos comandos de texto**.

Cabe llamar la atención sobre tres ideas fundamentales de la definición anterior. En primer lugar, la ficción interactiva es ante todo un género literario; es decir, su desarrollo se lleva a cabo principalmente a través de la expresión verbal. En segundo lugar, el papel destacado que desempeña la interacción del usuario en la experiencia de este tipo de software. Algunos autores, como Niesz y Holland, reflejan claramente esta importancia al ofrecer su definición particular para la ficción interactiva como «obras de ficción que invitan explícitamente al lector a interactuar con ellas mediante consultas o respuestas, a participar activamente de la historia y a cambiar deliberadamente el desarrollo de la trama, del personaje, del entorno o del lenguaje, junto con el autor»[Niesz and Holland, 1984]. Y, por último; las ficciones interactivas se sustentan sobre la simulación de un cierto entorno. «[Esta simulación] ofrece una modalidad de interacción continua, análoga a la interacción real, lo que puede resultar en una experiencia más inmersiva y cautivadora para los jugadores que las simples historias con numerosas ramificaciones»[Riedl et al., 2006]. El hecho de utilizar un modelo de mundo como base puede redundar igualmente en beneficio de ficciones interactivas con pretensiones pedagógicas de algún tipo, tal y como señalan también Riedl et al. (2006), al ofrecer potencialmente una experiencia de aprendizaje más realista, así como una progresión narrativa coherente.

El profesor N. Montfort del MIT recoge las ideas expuestas más arriba y caracteriza sucintamente la ficción interactiva del siguiente modo[Montfort, 2004]:

1. Es un software de computador que acepta texto como entrada e igualmente produce texto como salida (principalmente).
2. Es una narrativa potencial; esto es, un sistema que produce una narrativa a lo largo de la interacción con él.
3. Es la simulación de un entorno o modelo de mundo.
4. Es una estructura de reglas a través de las cuáles se persigue un resultado, lo que también se puede determinar como un juego.

A lo largo del presente trabajo se utilizará esta caracterización de Montfort como

marco de referencia para hablar de obras de ficción interactiva.

2.2 Sistemas de autoría

De acuerdo con la definición ofrecida anteriormente, toda ficción interactiva debe realizar al menos un cierto conjunto de tareas comunes que incluyen: modelar el entorno simulado en que se desarrolla la obra y el comportamiento de los objetos presentes dentro de la simulación; describir este entorno al usuario a través de una salida textual; procesar la entrada de texto del usuario para poder traducirla en acciones que provoquen efectos directos sobre la simulación. Estas tareas pueden implementarse con un nivel de sofisticación muy variable, pero si se atiende a otras obras de ficción interactiva publicadas a lo largo de las últimas décadas, es común observar una elevada profundidad en las interacciones y simulaciones de gran complejidad. Para facilitar el proceso de creación de ficciones interactivas con niveles de complejidad equivalentes existe un amplio ecosistema de herramientas software a disposición de los autores.

Habitualmente, un **sistema de autoría** está compuesto por un *parser* encargado de analizar la entrada de usuario y determinar sus unidades gramaticales, una librería software encargada de definir el comportamiento por defecto de los objetos dentro del entorno simulado y un compilador para generar los ficheros ejecutables.

2.2.1 Criterios para la selección de un sistema de autoría

Con la determinación de no repetir innecesariamente un trabajo ya realizado y evitar generar de raíz un sistema completamente nuevo, en una de las primeras fases del proyecto se ha procedido a investigar la oferta de sistemas de autoría disponibles y sus características, y a seleccionar aquel que se pudiese adecuar mejor a los propósitos del trabajo. A continuación se detallan los principales criterios establecidos antes de proceder a la selección de un sistema de autoría concreto:

1. **Debe ofrecer potencia suficiente.** Se pretende evitar en la medida de lo posible la interrupción de cualquier trabajo potencial debido a limitaciones graves de la tecnología seleccionada de partida. Se buscará así un sistema con un analizador de comandos y un soporte de modelo de mundo complejos y robustos. A ser posible el sistema debe ser capaz también de soportar contenidos audiovisuales que acompañen al texto (aunque este soporte sea reducido).
2. **Debe estar exhaustivamente documentado,** a fin de garantizar un proceso de formación ágil durante las primeras etapas de desarrollo del trabajo.
3. **Debe contar con una comunidad activa y numerosa de usuarios.** De este modo se garantiza que el sistema haya sido ampliamente sometido a prueba por autores con enfoques e intereses muy diferentes, y siga contando con soporte

por parte, bien del autor original, o bien de la comunidad. Este punto puede facilitar igualmente el proceso de formación en la herramienta, en caso de haber obras escritas con ese sistema que tengan disponible libremente su código fuente, y usuarios activos capaces de responder a eventuales dudas que puedan surgir durante el desarrollo.

4. **Debe permitir la creación de obras escritas en español.** No es un requisito fundamental, pero se considera oportuno fomentar la producción interactiva en lengua española —como una forma también de reivindicación cultural en un género cuya gran mayoría de obras publicadas están escritas en inglés—. La utilización del español agilizará igualmente el proceso de escritura, y facilitará el acceso por parte de la comunidad hispanohablante (si bien, por contra, reducirá significativamente el público potencial).

2.2.2 Sistemas estudiados

Los sistemas estudiados se han seleccionado de acuerdo a las siguientes consideraciones: número de obras creadas con el sistema y publicadas en los últimos veinte años³, actividad de los hilos sobre el sistema en los foros de las comunidades anglofona e hispanohablante⁴ o número de publicaciones académicas acerca del sistema. En total se han analizado once sistemas de autoría:

- ADRIFT (*Adventure Development & Runner – Interactive Fiction Toolkit*).
- AGE (*Aetheria Game Engine*).
- Alan (*Adventure LANguage*).
- Curveship.
- Diseñador de Aventuras AD (también conocido por sus siglas, DAAD).
- Hugo (*Hugo Interactive Fiction Development System*).
- Inform (en sus versiones 6 y 7).
- ngPAWS.
- Quest.
- Superglús.
- TADS (*The Adventure Development System*).

En el cuadro 2.1 se ofrece una comparativa de estos sistemas⁵, atendiendo entre otros a los criterios señalados en 2.2.1.

³Se consideran los últimos veinte años por intentar discriminar sistemas de autoría populares durante los años 80 y principios de los 90 que permitían generar ficciones interactivas —populares entonces por su enfoque principalmente lúdico y conocidas, como ya se ha señalado, como *juegos de texto o aventuras conversacionales*— para plataformas consideradas retro hoy en día: ZX Spectrum, Amstrad CPC, Commodore 64, etc.

⁴Direcciones web de los principales foros en las comunidades de habla inglesa y española, respectivamente: <https://intfiction.org/>; <http://foro.caad.es/>.

⁵La información ofrecida en el cuadro se ha extraído de los espacios web oficiales de cada uno de los sistemas de autoría (en caso de existir), así como de las Wikis de las comunidades anglofona (<http://www.ifwiki.org>) e hispanohablante (<http://wiki.caad.es/>), y de *The Interactive Fiction Database* (<https://ifdb.tads.org/>).

SISTEMA	LENGUAJE DE PROGRAMACIÓN	POTENCIA DE LA LIBRERÍA	DOCUMENTACIÓN	VOLUMEN DE OBRAS CREADAS	PERMITE CREAR OBRAS EN ESPAÑOL	PLATAFORMAS
ADRIFT	Gráfico	Medio-bajo	Esencial	≈ 450	Si	Windows, Mac, Linux, Web
AGE	Gráfico / BeanShell	Medio-alto	Esencial	≈ 16	Si	Java
Alan	Autodocumentado (estilo COBOL)	Medio	Esencial	≈ 55	Si	Windows, Mac, Linux
Curveship	Python	Medio-alto	Esencial	≈ 3	No	Python
DAAD	Bajo nivel (estilo PAWS)	Bajo	Esencial	6	Si	Spectrum, CPC, Commodore 64...
Hugo	Alto nivel	Medio	Esencial	≈ 55	No	Windows, Mac, Linux
Inform ngPAWS	Alto nivel	Alto	Numerosa	≈ 2160	Si	Z-Machine, Gluix Web
Quest	Bajo nivel (estilo PAWS)	Bajo	Esencial	≈ 5	Si	
Superglús	Gráfico/Scripts de alto nivel	Medio-alto	Numerosa	≈ 620	Si	Windows, Web
TADS	Bajo nivel (estilo PAWS)	Bajo	Esencial	≈ 15	Si	Gluix
	Alto nivel	Alto	Numerosa	≈ 470	No	TADS, Web

Cuadro 2.1: Comparación de sistemas de autoría.

2.3 El sistema de autoría Inform

Tras proceder al análisis de los sistemas especificados en 2.2.2 se ha optado por utilizar el sistema de autoría **Inform** en su versión 6.

De entre todas las herramientas estudiadas, Inform es la que ofrece infraestructura para crear unos de los modelos de mundo más profundos, superado tal vez sólo por TADS —si bien en la práctica ambos sistemas pueden considerarse similares[Eve, 2009] y, a diferencia de éste último, con Inform es posible crear obras en español—; hay gran cantidad de libros, tutoriales y código fuente de ejemplo para facilitar el proceso de aprendizaje del sistema —mucho mayor que otros sistemas menos extendidos como el sistema de origen español AGE, o el sistema Curveship creado por N. Montfort—; y cuenta con la comunidad de usuarios más activa y numerosa⁶, desde la que todavía surge un gran número de obras, extensiones y herramientas nuevas, y que da soporte a través de los foros dedicados al sistema a las dos versiones que siguen siendo utilizadas simultáneamente hoy en día: Inform 6 e Inform 7.

En cualquier de sus dos versiones, Inform permite por defecto el desarrollo de ficciones interactivas en inglés. Existen, no obstante, diversas traducciones disponibles libremente de sus módulos de idioma para crear obras en otras lenguas, entre ellas el castellano. El módulo de idioma español se conoce como **INFSP** (con versiones para I6 e I7).

2.3.1 Inform 6

Disponible desde 1996, la versión 6 de Inform está integrada por dos componentes principales: un compilador que genera ficheros ejecutables a partir de código fuente escrito en su propio lenguaje de programación de propósito específico —orientado a objetos y funcional e inspirado en el lenguaje C[Nelson, 2018]—; y una extensa librería de software encargada de automatizar las tareas de procesar la entrada de texto del usuario y de registrar el estado del modelo del mundo en que se desarrolla la obra.

2.3.2 Inform 7

Con la última iteración del sistema publicada en 2006, su versión 7, se lleva a cabo una reinterpretación completa del lenguaje de programación. Este nuevo lenguaje desecha el paradigma estructurado y orientado a objetos de la versión anterior y pasa a adoptar uno basado en reglas que trata de imitar la escritura en lenguaje natural (en

⁶El índice TIOBE, que trata de dar una medida del uso e influencia de distintos lenguajes de programación, da testimonio del éxito y aceptación de Inform al colocar el suyo entre los 100 lenguajes más influyentes del mundo —ha llegado a ocupar la posición 49 de este índice en 2015; por encima de Mathematica y justo detrás de FORTRAN[Nelson, 2018]—; un hito especialmente destacable teniendo en cuenta lo específico de su dominio de trabajo.

you: a day on the slopes, the brisk February wind against your face; polishing off a keg back at the lodge; those two girls you and Rob had hit it off with, the ones who'd given you their address in town. "We all should get together sometime!" they'd said. Of course, Rob insisted that by "sometime" they'd meant "later tonight." You hadn't been so sure, but then you'd blacked out before you could argue the point.

How Rob came to be driving your car you're not exactly sure. Apparently he couldn't wait till you were sober enough to drive it yourself. From the way he's weaving all over the road, he also apparently couldn't wait till HE was sober enough to drive it, either.

Rob changes the radio station a few times, eventually settling on the station you were listening to in the first place.

> **examine rob**

Rob's face is flushed, as you might expect from someone fresh off the slopes. Except you're NOT fresh off the slopes. You came in at maybe three or three-thirty, and now it's past midnight.

Rob looks at the scrap of paper with the address on it as the two of you go screaming through an empty intersection. "Aw, man, it's a fake address! They gave us a f- no, wait, it's upside down." He turns the paper right-side up. "Oh, hey, they're right on Bartlett Hill Road. Sweet!"

> |

Figura 2.1: Captura de pantalla de *Photopia* (A. Cadre, 1998). Creada con Inform 6.

inglés). Inform 7 introduce también un nuevo entorno integrado de desarrollo propio, con herramientas especializadas en las tareas de prueba y depuración de las obras, así como una traducción directa de la librería Inform 6 al nuevo lenguaje. Al tratarse de una traducción directa, la potencia del analizador de comandos y la riqueza del modelo del mundo es en la práctica idéntica entre Inform 6 e Inform 7 —más aún; el nuevo compilador de I7 sigue siendo compatible con código escrito en I6, de modo que autores que han trabajado con la versión 6 del sistema pueden reutilizar gran parte de su código ante una eventual decisión de pasar a crear obras con la versión posterior (no así al contrario)—.

Estas observaciones, junto con una cantidad inferior de obras escritas en español en Inform 7, y una preferencia personal por el paradigma orientado a objetos sobre el paradigma orientado a reglas, han motivado la decisión de utilizar Inform 6.

2.3.3 Z-machine y Glulx

La primera versión del compilador Inform publicada por Graham Nelson en 1993 (como parte de la versión 1 del sistema) generaba ejecutables para **Z-machine**, una máquina virtual de 16 bits creada en 1979 por Infocom para sus obras, comercializados entre los años 1979 y 1989⁷. De este modo, el sistema Inform se aprovechaba de la popularidad de esta compañía de software y permitía que las obras creadas con el nuevo sistema pudiesen ejecutarse directamente sobre una gran cantidad de equipos para los que *Z-machine* había sido ya portada. Todas las versiones subsiguientes del

⁷ *Z-machine* recibe su nombre de la primera de las aventuras de texto publicadas por Infocom: *Zork* (1979).

sistema han seguido permitiendo el compilado para esta máquina virtual.

Desde la versión 6.30, el compilador Inform pasa a ser multiplataforma y permite generar ejecutables tanto para *Z-machine* como para una nueva máquina virtual llamada **Glulx**.

Glulx, diseñada por Andrew Plotkin en 1999, se ideó con el propósito de solventar algunas de las limitaciones impuestas por *Z-machine* a las obras creadas con el sistema Inform, como superar la aritmética basada 16 bits y utilizar en su lugar una aritmética de 32 bits, o utilizar datos y direcciones también de 32 bits para posibilitar el manejo de archivos de juego de hasta 4 gigabytes[Plotkin, 2014]. Glulx incluye asimismo soporte nativo para la interfaz de entrada/salida Glk, con lo que las ficciones interactivas compiladas para esta máquina virtual pueden implementar un amplio conjunto de funcionalidades especificadas por Glk que no son soportadas por *Z-machine*, entre ellas: un nuevo mecanismo para la utilización de recursos multimedia, un mayor conjunto de estilos de texto disponibles o la posibilidad de utilizar hipervínculos[Cadre, 2002]. Además, al igual que sucede con *Z-machine*, *Glulx* está implementada en un amplio conjunto de plataformas; desde Windows/Mac/Linux, a plataformas móviles y Web.

A la hora de desarrollar el presente trabajo se dará preferencia al uso de Glulx sobre *Z-machine* a fin de poder explotar el conjunto de funcionalidades adicionales ofrecidas por esta máquina virtual.

3 | Extensiones al sistema Inform

«Se puede escribir ficción convencional —estática— usando simplemente lápiz y papel, o una máquina de escribir, o un procesador de textos; las exigencias para crear FI, sin embargo, son algo mayores, y el proceso creativo un poco más complejo.»

—THE INFORM BEGINNER'S GUIDE
(R. Firth y S. Kesserich, 2004)

Se documenta a continuación el desarrollo del conjunto de extensiones con el que se pretende, principalmente, mejorar la accesibilidad de obras creadas con Inform —si bien se desarrollan además otros módulos para extender las funcionalidades base de este sistema de autoría, a fin de facilitar el proceso de creación de nuevas ficciones interactivas—. El capítulo recoge las barreras de accesibilidad tenidas en cuenta, captura de requisitos, diagramas de clase (allí donde se ha estimado oportuno) y el diseño detallado de los diferentes módulos ideados.

3.1 Relación de barreras abordadas

En este primer apartado se consideran algunos de los obstáculos de accesibilidad más limitantes con los que se puede encontrar un usuario al intentar operar un software de ficción interactiva, y las medidas que se planea adoptar para intentar mitigarlos:

- **Barreras visuales.** Una gran ventaja de la ficción interactiva es que, al sostenerse principalmente sobre texto, es posible hacer este tipo de obra accesible incluso para personas con carencia total o parcial de visión simplemente utilizando softwares ya existentes de sintetización de voz, sin necesidad de ningún desarrollo adicional. A lo largo del capítulo se plantean, no obstante, diferentes funcionalidades enfocadas en hacer más cómoda y legible la lectura así como diferenciar más claramente los distintos niveles de información presentes en una obra (a través, por ejemplo, de la facilitación de estilos de texto diferentes para los mensajes diegéticos y los extradiegéticos).

- **Barreras auditivas.** De nuevo, al fundamentarse sobre el texto impreso, la ficción interactiva resulta accesible de forma natural para personas con diversos problemas de audición¹.
- **Barreras motrices.** Se proveerán mecanismos que agilicen la introducción de comandos por parte del usuario; desde la utilización de sintaxis simplificada para los comandos más habituales a la creación de un mecanismo de hipervínculos que permita interactuar con las obras directamente a través de clics de ratón en lugar de la introducción de órdenes de texto. Se procurará además no hacer uso de eventos temporizados que requieran de entradas coordinadas por parte del usuario.
- **Barreras cognitivas.** Se utilizarán principios de la metodología de redacción en lectura fácil. Aunque los textos finales en una ficción interactiva están sujetos a las intenciones artísticas de su autor, se procederá a la reescritura de la librería de mensajes por defecto de Inform, utilizando como referencia los materiales recogidos en *Lectura fácil: métodos de redacción y evaluación*[García Muñoz, 2012].

3.2 Ciclo de vida del proyecto

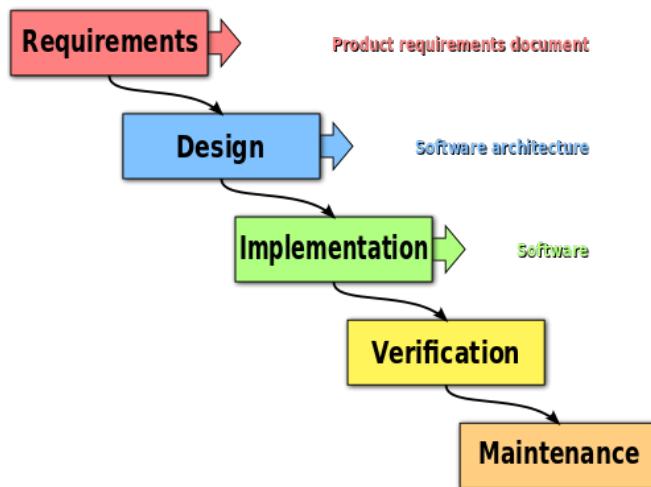


Figura 3.1: Diagrama del “*modelo en cascada*”, por P. Kemp y P. Smith (2010).

Para la tarea de desarrollo del conjunto de extensiones se opta por un **ciclo de vida en cascada** (ver figura 3.1). El alcance y la complejidad de cada extensión individual se estima lo suficientemente reducido como para aceptar los riesgos que puede

¹Aunque se pretenden mitigar las barreras visuales y auditivas con las que pueden encontrarse los usuarios, de ningún modo se hace necesario prescindir de las posibilidades multimedia del computador (si acaso, antes al contrario; estas posibilidades facilitarán en gran medida la accesibilidad). Se procurará, sin embargo, que cualquier información audiovisual –presentada a través de ilustraciones o pistas de audio– se explice igualmente en el texto, aunque resulte redundante. Así se garantiza que todos los usuarios puedan acceder a esa información.

ID.	DESCRIPCIÓN
RF-A01	El autor podrá iniciar la aplicación de un estilo de texto.
RF-A02	El autor podrá establecer un estilo de texto para los mensajes extradiegéticos de la obra.
RF-A03	El autor podrá establecer un prefijo para los mensajes extradiegéticos.
RF-A04	El autor podrá establecer un sufijo para los mensajes extradiegéticos.
RF-A05	El autor podrá inicializar recomendaciones de aspecto por defecto para los diferentes estilos de texto disponibles.
RF-A06	El autor podrá configurar las dimensiones de los márgenes y las ventanas de ilustraciones en la interfaz gráfica de usuario.
RF-A07	El autor podrá generar hipervínculos asociados a un objeto que, al ser seleccionados, lanzarán una acción por defecto sobre ese objeto.
RF-A08	El autor podrá modificar la acción por defecto de los hipervínculos.
RF-A09	El autor podrá generar hipervínculos asociados a una cadena de caracteres que, al ser seleccionados, lanzarán el comando especificado por dicha cadena.
RF-A10	El usuario podrá lanzar acciones por defecto sobre un objeto tecleando simplemente el nombre del objeto.
RF-A11	El autor podrá modificar la acción por defecto de la sintaxis simplificada contemplada en REF-A10.

Cuadro 3.1: Requisitos funcionales asociados con la accesibilidad.

suponer este modelo sobre otros ciclos de vida populares como por ejemplo el iterativo incremental —con el modelo en cascada, si se detectan problemas con el diseño o los requisitos funcionales en etapas finales del desarrollo del software implicaría rehacer grandes cantidades de trabajo—. Además, el amplio conocimiento adquirido sobre el sistema de autoría Inform y sus posibilidades, la claridad de los objetivos propuestos o el hecho de no estar trabajando en las necesidades de un cliente concreto, motivan la preferencia de esta elección.

3.3 Especificación de requisitos

En los cuadros 3.1, 3.2 y 3.3 se extraen una serie de requisitos funcionales relacionados con cuestiones de accesibilidad y extensiones a la librería Inform por defecto. Por último en esta fase, se han extraído una serie de requisitos no funcionales con apreciaciones adicionales a tener en cuenta durante las próximas etapas del desarrollo (ver cuadro 3.4).

ID.	DESCRIPCIÓN
RF-E01	El autor podrá establecer una flexión gramatical para los mensajes de la librería, seleccionando tanto entre 1 ^a , 2 ^a y 3 ^a persona como entre pasado, presente y futuro.
RF-E02	El autor podrá consultar el género de un objeto presente en el modelo de mundo de la obra.
RF-E03	El autor podrá consultar el número de un objeto presente en el modelo de mundo de la obra.
RF-E04	El autor podrá consultar si un contenedor del modelo de mundo tiene capacidad suficiente en número de objetos soportados para albergar a otro.
RF-E05	El autor podrá consultar si un contenedor tiene capacidad suficiente en tamaño (en cualquier eje) para albergar a otro objeto dado.
RF-E06	El autor podrá comprobar si un contenedor tiene capacidad suficiente en volumen para albergar a otro objeto dado.
RF-E07	El autor podrá comprobar si un contenedor tiene capacidad suficiente en peso para albergar a otro objeto dado.
RF-E08	El autor podrá capitalizar la primera letra de cada palabra en un array de caracteres.
RF-E09	El autor podrá concatenar dos arrays de caracteres en uno solo.
RF-E10	El autor podrá consultar la longitud de un array de caracteres.
RF-E11	El autor podrá imprimir en pantalla un array de caracteres, carácter a carácter, en mayúsculas o minúsculas.
RF-E12	El usuario podrá utilizar los nuevos comandos: <i>Dance</i> , <i>GoDown</i> , <i>GoUp</i> , <i>KnockOn</i> , <i>Masturbate</i> , <i>Read</i> , <i>Shout</i> , <i>Sit</i> , <i>SleepWith</i> , <i>StandUp</i> , <i>Untie</i> , <i>Use</i> , <i>Write</i> y <i>Xyzzy</i> .
RF-E13	El autor podrá generar conjuntos de elecciones seleccionables por el usuario y activarlos/desactivarlos en cualquier momento en tiempo de ejecución.
RF-E14	El usuario podrá seleccionar una elección de un conjunto activo utilizando entrada de texto libre.
RF-E15	El autor podrá añadir subelementos a cada una de las elecciones de un conjunto, de tal manera que cuando un usuario seleccione dicha elección, todos sus subelementos sean añadidos al conjunto activo.
RF-E16	El autor podrá vincular <i>n</i> elecciones a otra elección dada, de tal manera que cuando un usuario seleccione dicha elección, todas las otras elecciones vinculadas sean consideradas también como tratadas.

Cuadro 3.2: Requisitos funcionales asociados a otras extensiones de la librería Inform.

ID.	DESCRIPCIÓN
RF-E17	El autor podrá marcar una elección como permanente, de tal forma que ésta no desaparezca del conjunto activo aún cuando el usuario la haya seleccionado (permitiendo que vuelva a ser seleccionada de nuevo).
RF-E18	El autor podrá marcar una elección como invisible, de tal forma que ésta pueda ser seleccionada por el usuario aunque no sea mostrada al imprimir el inventario de elecciones del conjunto activo.

Cuadro 3.3: Requisitos funcionales asociados a otras extensiones de la librería Inform (continuación).

3.4 Diseño de alto nivel

Tal y como se señaló en el capítulo 2.3, el lenguaje Inform sigue un paradigma múltiple: imperativo y orientado a objetos, inspirado en gran medida en el lenguaje C[Nelson, 2018]. Mientras que las propiedades OOP se utilizan ampliamente durante el modelado de mundo, la lógica encargada de gestionar el bucle de juego principal se basa principalmente en rutinas de ámbito global ejecutadas de manera secuencial. Gran parte de las extensiones que se desarrollan a continuación afectan directamente a ese bucle, de manera que el enfoque adoptado está basado principalmente en el paradigma imperativo puro, salvo excepciones en las que se recurre a la orientación a objetos.

A pesar de esta puntualización, durante el diseño del software se ha decidido tener presentes dos conocidos principios de la programación orientada a objetos: el principio de responsabilidad única (SRP, por sus siglas en inglés), según el cuál «cada módulo, clase o función debería tener responsabilidad sobre una sola parte de la funcionalidad proporcionada por el software»[Martin, 2003], y el principio abierto/cerrado (OCP), que establece que «las entidades software (clases, módulos, funciones, etc.) deberían estar abiertas a la extensión, pero cerradas a modificaciones».[Meyer, 1997]². Con ello se busca asegurar la independencia de los módulos (de tal manera que los autores puedan incluir en sus obras únicamente las funcionalidades que les interesen, sin forzarles a utilizar el paquete completo), y facilitar la personalización de estas funcionalidades de acuerdo a las necesidades puntuales de cada obra.

El siguiente es el conjunto de módulos ideados para satisfacer los requisitos planteados en la sección 3.3:

²Estos dos principios se identifican con las letras *S* y *O* del acrónimo *SOLID*, popular en el paradigma OOP.

ID.	DESCRIPCIÓN
RF-NF01	A la hora de desarrollar el software se adoptarán criterios de estilo similares a los presentados por la propia librería Inform.
RF-NF02	Todas las extensiones deben ser bi-plataforma (<i>Z-Machine/Glulx</i>). Aquellas funcionalidades que no estén disponibles en la máquina virtual <i>Z-Machine</i> simplemente quedarán desactivadas.
RF-NF03	Se prestará especial atención a la legibilidad del código fuente de las extensiones, considerando el objetivo de distribuirlas entre otros autores para facilitar su uso.
RF-NF04	Se adoptará un sistema de documentación para las funciones, clases y objetos análogo al sistema javadoc, con objeto de facilitar el futuro uso entre otros autores.
RF-NF05	Los módulos o extensiones deben ser completamente independientes entre ellos. Los autores deberán poder utilizar cada una de forma individual, sin forzarles a que adopten el paquete completo de extensiones en sus obras.
RF-NF06	Los módulos estarán escritos para la versión 6.34 del compilador Inform, utilizando la codificación de caracteres UTF-8.

Cuadro 3.4: Requisitos no funcionales.

3.4.1 Arrays

Inform trata a las cadenas de caracteres (*strings*) como constantes[Nelson, 2001], de manera que resulta imposible manipularlas en tiempo de ejecución. Para simular operaciones sobre cadenas de caracteres es necesario hacer uso arrays, tal y como se ha dejado especificado en la relación de requisitos del cuadro ???. Este pequeño conjunto de operaciones se implementará en su propio módulo.

3.4.2 Capacities

Módulo encargado de implementar lógica adicional al comportamiento por defecto de la librería Inform para tratar las reglas de capacidad de contenedores y soportes en el modelo de mundo. Existe un módulo: `recept`, de Peer Schaefer, que implementa un comportamiento similar al aquí planteado y que se puede utilizar como referencia³.

³La forma en que está implementado este módulo `recept` no respeta el principio SRP e incluye, además, mensajes de respuesta en inglés no modificables (se haría necesario manipular el código del módulo para preparar una traducción al español, vulnerando también el principio OCP). Por este motivo, en lugar de reutilizar la extensión se opta por rediseñar un módulo completamente nuevo.

3.4.3 Choice sets

Este módulo implementará un sistema de elección de opciones de entrada de entre un conjunto delimitado y arbitrario definido por el autor, lo que abre la puerta a la utilización de comandos en lenguaje natural (útil, por ejemplo, en el diseño de sistemas de conversación). El diseño está basado precisamente en las notas de Eric Eve sobre sistemas de conversación [Eve et al., 2008].

Existe un pequeño módulo: *NPC Conversacion*, de Francisco J. López, sobre el que puede basarse el diseño del nuevo módulo (con el permiso del autor).

3.4.4 GWindows GUI

Módulo con una aproximación de interfaz gráfica adaptable al tamaño de la ventana principal de la aplicación. Ideada para mejorar la experiencia de lectura del texto y con soporte gráfico para ilustraciones a través de dos ventanas laterales configurables —la figura 3.2 muestra la distribución de ventanas en una interfaz utilizando GWindows GUI. Como comparación, se puede consultar de nuevo la figura 2.1 con el aspecto de la interfaz Inform 6 por defecto—.

Este módulo se desarrollará sobre la librería GWindows de L. Ross Raszewski: un potente *framework* para gestionar el modelo de pantalla en la máquina virtual Glulx. Gracias a GWindows «es posible crear una estructura de objetos a través de los que describir la disposición de pantalla [...]. Se encarga de gestionar aquellas cuestiones “estrechamente vinculadas” a las ventanas, como el dibujado de imágenes, la entrada y salida de texto, la actualización de pantalla, la configuración de sugerencias de estilo y los cambios de tamaño de la ventana principal» [Raszewski, 2004].

En la figura 3.3 se incluye el árbol de objetos GWindows definido por el módulo.

gstatusX

Módulo adicional para GWindows que permite crear barras de estado simplificadas (en las que simplemente se imprime el nombre de la localidad actual del usuario, centrado en pantalla)⁴.

3.4.5 Hyperlinks

La máquina virtual *Glulx*, al incluir soporte nativo para la librería de entrada/salida Glk, ofrece un conjunto de funcionalidades adicionales a aquellas disponibles en Z-

⁴Las barras de estado son ventanas de texto que se utilizan habitualmente en las ficciones interactivas creadas con Inform y en las que, en general, se muestra información tal como el nombre de la localidad actual del usuario, su puntuación o la cantidad de turnos empleados, por ejemplo.

	<p>> ambrosio Como los otros <u>pastores</u>, vestía con una zamarra de negra lana y estaba coronado con una guirnalda que a lo que parecía era de ciprés. Aliviaba parte de su peso sobre un pico que en ese momento hacía servir a modo de báculo.</p> <p>> muerto Lo habían traído antes de la luz primera en unas andas cubiertas de mucha diversidad de flores y de ramos. El <u>cuerpo</u>, cubierto igualmente de flores, estaba vestido como pastor, de edad, al parecer, de treinta años; y, aunque muerto, mostraba que vivo había sido de rostro hermoso y de disposición gallarda. Alrededor de él tenía en las mismas andas algunos libros y muchos <u>papeles</u>, abiertos y cerrados.</p> <p>> pastores Salvo por los gentilhombres, vestidos con largos ropajes oscuros, y sus sirvientes, que llevaban camisolas sencillas, los demás tenían toda la apariencia de ser <u>pastores</u> de la zona; vestidos con pellicos negros y coronadas las cabezas con guirnaldas de ciprés y de amarga adelfa. Traía cada uno un grueso bastón de acebo en la mano.</p> <p>El pastor siguió con su predica:</p> <p>—Ese cuerpo, señores, que con piadosos ojos estáis mirando, fue depositario de un alma en quien el cielo puso infinita parte de sus riquezas. Ese es el cuerpo de <u>Grisóstomo</u>, que fue único en el ingenio, solo en la cortesía, extremo en la gentileza, fénix en la amistad, magnífico sin tasa, grave sin presunción, alegre sin bajeza, y, finalmente, primero en todo lo que es ser bueno, y sin segundo en todo lo que fue ser desdichado.</p> <p>> </p>	

Figura 3.2: Distribución de ventanas con GWindows GUI.

Machine entre las que se encuentra la posibilidad de responder a eventos de selección de hipervínculos[Plotkin, 2017a]. Este módulo incluirá la infraestructura necesaria para facilitar la utilización de hipervínculos como método de interacción (además de la introducción habitual de comandos de texto a través del teclado).

Los hipervínculos utilizan habitualmente estilos de texto destacados para diferenciarse del resto de la narración. Respetando el principio OCP, el módulo ha de implementar mecanismos que permitan a los autores personalizar la lógica encargada de establecer los estilos de texto destacado sin necesidad de manipular el código del propio módulo.

3.4.6 Parsing preprocessor

Resulta imposible diseñar un módulo independiente a través del que ofrecer funcionalidad para el requisito RF-A10 (ver cuadro 3.1), puesto que es necesario introducir modificaciones en numerosas rutinas de la librería Inform que son utilizadas, por su parte, por otros módulos y rutinas del sistema diferentes. Se opta, en su lugar, por dividir la funcionalidad en problemas más pequeños, trasladar a este módulo aquellos que no generan dependencias con otras partes del sistema, y dejar que cada obra se ocupe del resto de la implementación —en el capítulo 4 se desarrollan varias obras de ejemplo que otros autores pueden consultar para ver la forma en que puede implementarse la sintaxis simplificada de RF-A10—.

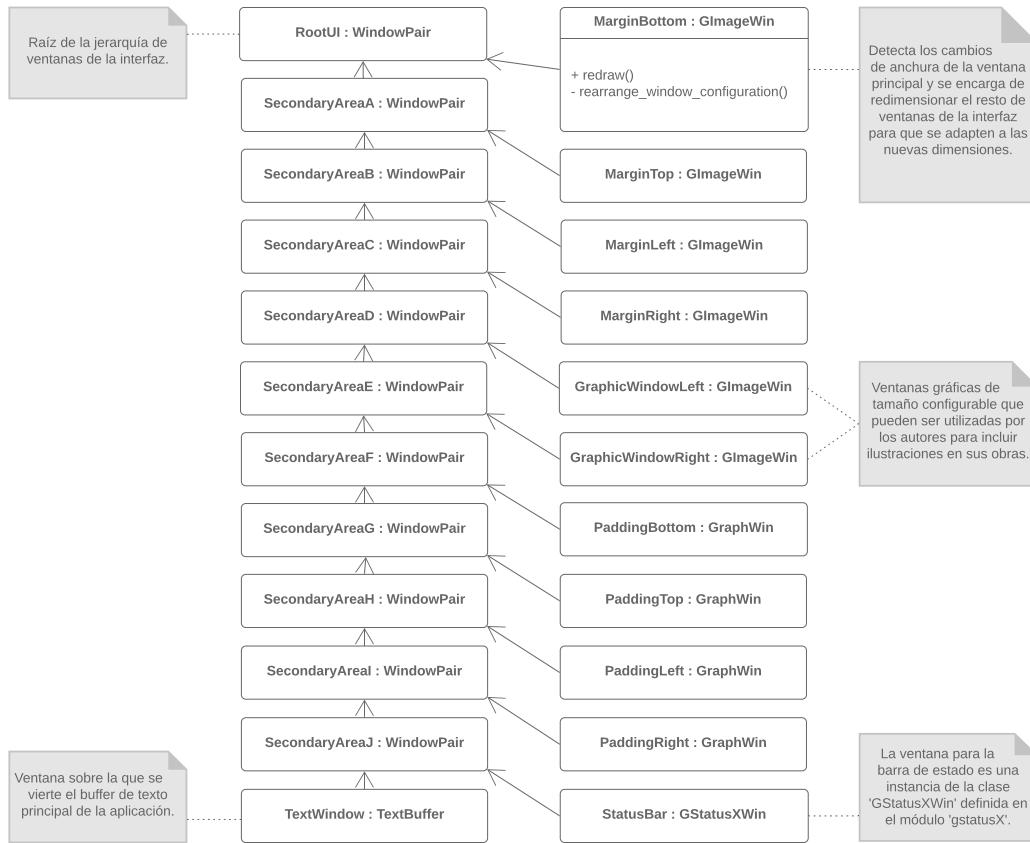


Figura 3.3: Jerarquía de ventanas de la interfaz definida en GWindows GUI.

Parsing preprocessor se ocupará de identificar correspondencias entre la entrada de texto del usuario y los objetos a su alcance en el modelo de mundo.

3.4.7 Spanish extended grammar

Extensión al conjunto de acciones comprendidas por defecto por la librería Inform. Incluirá, además de algunos otros, la traducción al español de los comandos definidos por el módulo inglés ExpertGrammar de Emily Short.

Si bien será independiente del módulo 3.4.8, podrá interactuar con éste y permitir igualmente cambios tanto en el tiempo verbal de los mensajes y la persona del narrador, como el género y el número del personaje protagonista.

3.4.8 Spanish library messages

Reescritura completa de los mensajes por defecto de la librería Inform, contemplando principios de escritura fácil para facilitar la comprensión de la información. Se ocupará de eliminar ambigüedades del narrador presentes en los mensajes de la

ID.	TEXT STYLES	ESTILO GLULX	ESTILO Z-MACHINE
0	Upright	Normal	Normal
1	Stressed	Emphasized	Italic
2	Important	Subheader	Bold
3	Monospaced	Preformatted	Fixed-width
4	Header	Header	Bold
5	Note	Note	Italic (y Bold)
6	Reversed	Alert	Reverse
7	Quote	BlockQuote	Fixed-width
8	Input	Input	Bold
9	User1	User1	Normal
10	User2	User2	Normal

Cuadro 3.5: Estilos de Text styles y sus correspondencias en *Glulx* y *Z-Machine*.

librería original (diferenciando claramente, por ejemplo, la voz de los mensajes diegéticos de los extradiegéticos), corregirá inconsistencias de estilo y tono heredadas de las aventuras clásicas de Infocom, y detallará con mayor claridad las causas de numerosos mensajes de error para que los usuarios puedan comprender mejor qué ha fallado.

Este módulo añadirá, además, flexiones gramaticales sobre cada mensaje para posibilitar la modificación en tiempo de ejecución tanto del tiempo verbal (presente, pasado y futuro) y la persona (primera, segunda y tercera) del narrador, como el género (masculino y femenino) y el número (singular, plural) del personaje protagonista controlado por el usuario.

3.4.9 Text styles

Se ocupará de definir un conjunto de rutinas bi-plataforma para seleccionar estilo de texto. Las máquinas virtuales *Z-Machine* y *Glulx* para las que compila Inform cuentan con un número diferente de estilos de texto disponibles: *Z-Machine* contempla un total de 5 estilos de texto diferentes (algunos intérpretes pueden reconocer combinaciones entre ellos, pero éstas no forman parte del estándar de la máquina)[Nelson and Fillmore, 2014], mientras que *Glulx*, por su parte, extiende el conjunto de estilos disponibles hasta los 11[Plotkin, 2017b]. No sólo eso, el modo de establecer uno u otro estilo en cada máquina virtual es radicalmente distinto. *Text styles* definirá 11 estilos de texto (basándose en los estilos disponibles en *Glulx*) y establecerá una función sobreyectiva de correspondencia con los estilos de *Z-Machine*, tal y como se refleja en el cuadro 3.5.

Implementará también un estilo virtual adicional *parser* que podrá ser utilizado

por los mensajes de tipo extradiegético emitidos por el sistema a fin de facilitar el proceso de lectura.

3.5 Diseño detallado

Para los diferentes módulos definidos en la sección anterior se ha decidido respetar el modo de programación basado fundamentalmente en funciones que utiliza la librería Inform por defecto —los módulos se van a implementar de forma similar a como se programaría con un lenguaje tipo C—, relegando a un segundo término la utilización de prácticas OOP. Si bien esta forma de programación no encaja exactamente con los diagramas de clases UML, se ha considerado que estos diagramas son una herramienta muy útil a la hora de representar gráficamente las especificaciones y la estructura del software durante la fase de diseño en detalle.

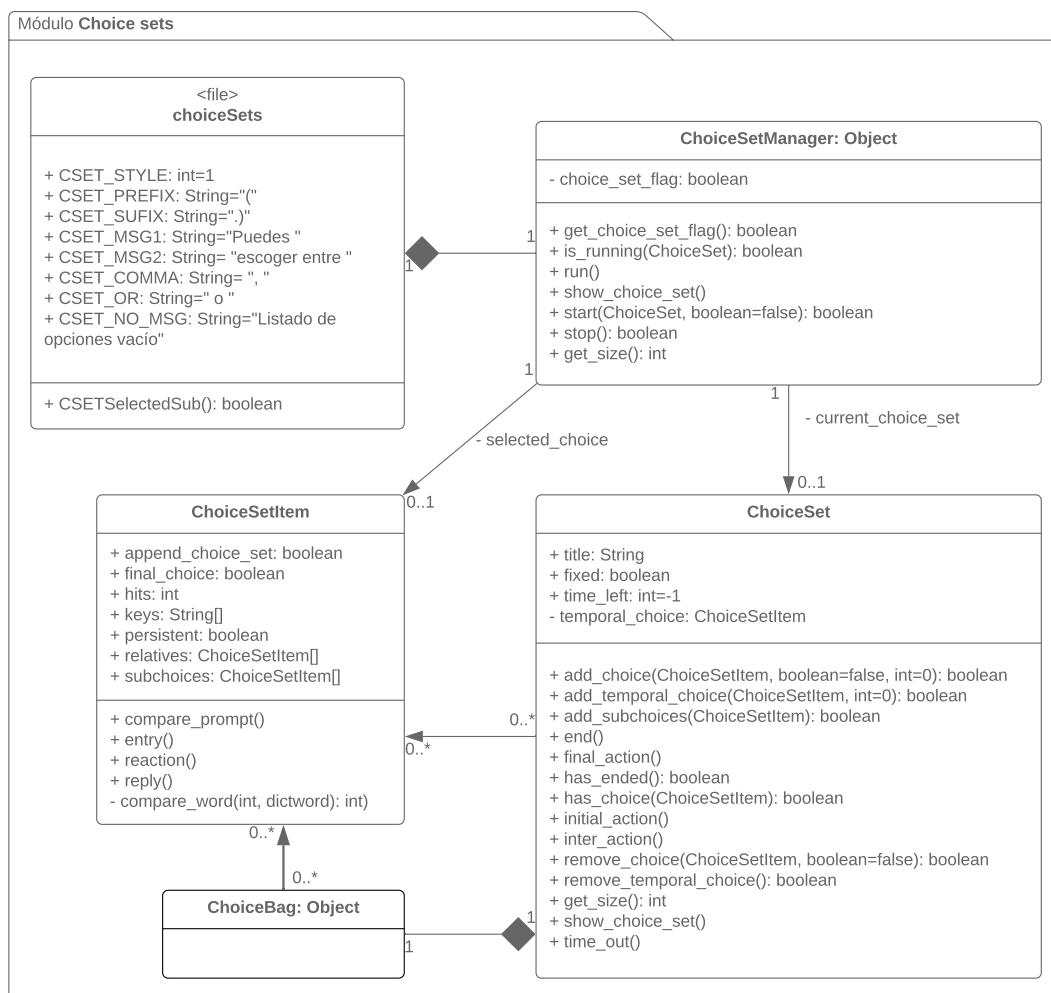


Figura 3.4: Diagrama en detalle del módulo *Choice sets* (3.4.3).

De este modo se ha optado por seguir las recomendaciones de Martin Bakal para modelar a través de UML software desarrollado de manera funcional mediante la utili-

zación de «un elemento UML denominado “file”, que es simplemente una representación gráfica de un fichero fuente. Estos “file” pueden contener todos los elementos con los que los desarrolladores en C están familiarizados, incluyendo variables, funciones, tipos, etc. Pero puede ser introducido en un diagrama UML del mismo modo que se haría con una clase en la programación orientada a objetos clásica»[Bakal, 2005].

En las figuras 3.4, 3.5 y 3.6 se muestra el diseño en detalle de los diferentes módulos desarrollados.

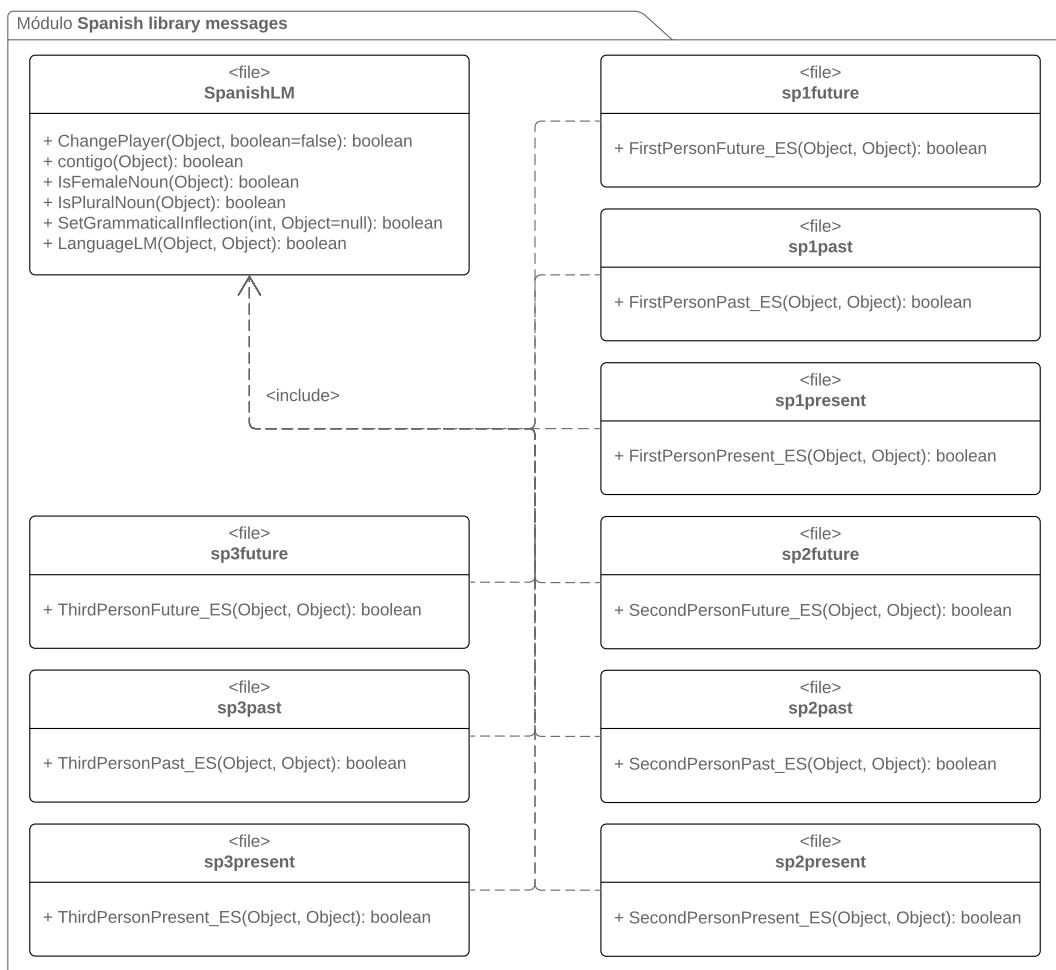


Figura 3.5: Diagrama en detalle del módulo *Spanish library messages* (3.4.8).

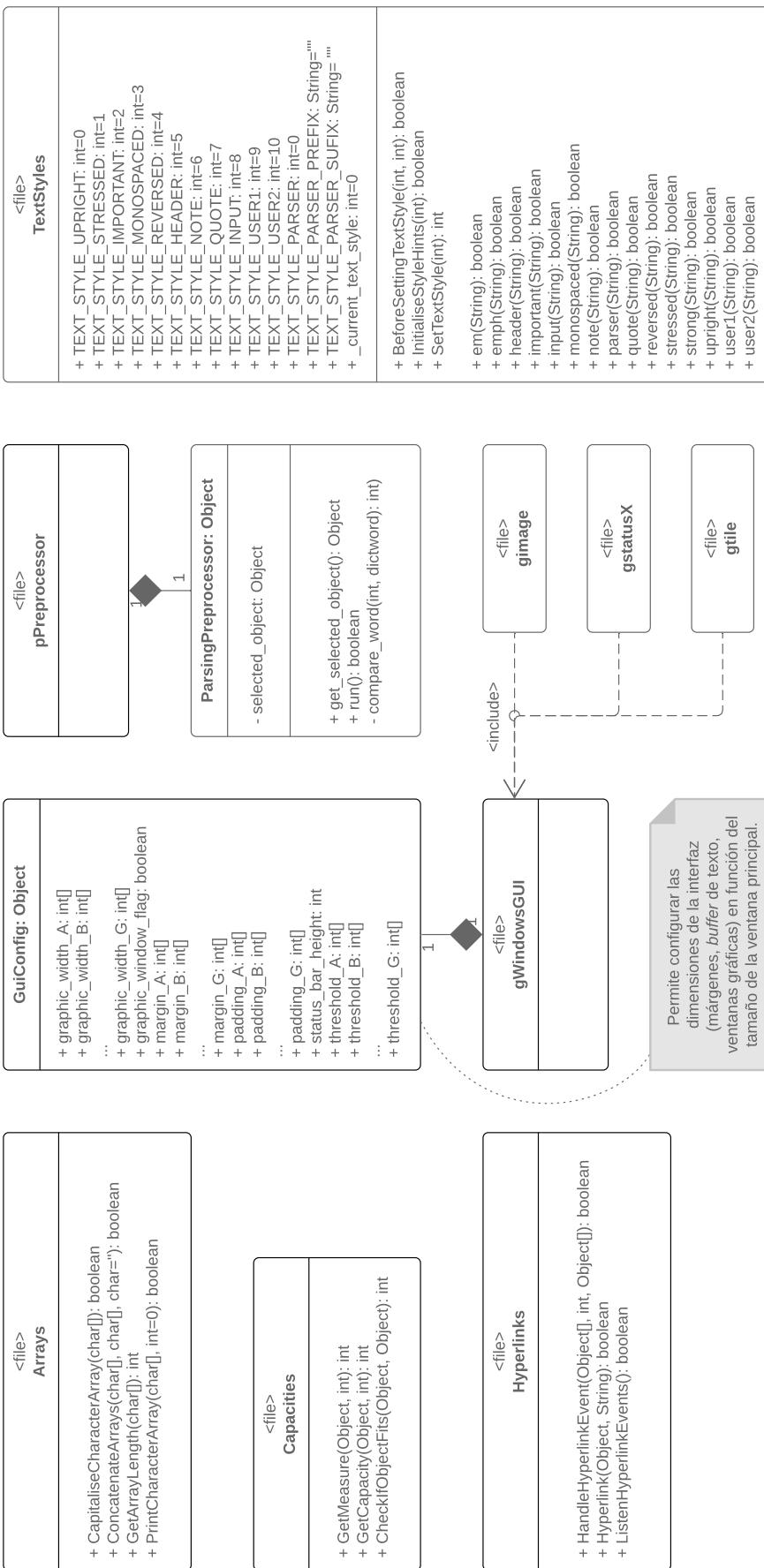


Figura 3.6: Diagrama en detalle del resto de módulos definidos en 3.4.

4 | Tres ficciones interactivas

> observa a Marcela

Los que hasta entonces no la habían visto la miraban con admiración y silencio, y los que ya estaban acostumbrados a verla no quedaron menos suspensos que los que nunca la habían visto.

—HISTORIA DE GRISÓSTOMO Y MARCELA
(J. Francisco Martín, 2018)

A lo largo este capítulo se diseña un conjunto de tres ficciones interactivas: *La pequeña cerillera*, *Historia de Grisóstomo y Marcela* y *La zona*, con objeto por una parte de exponer y servir a otros autores a modo de ejemplo de uso de los módulos desarrollados anteriormente, pero también con el de crear obras accesibles que trasciendan la función de demostración y resulten en artefactos culturales con valor sí mismos. En este capítulo se implementan, además, las últimas funcionalidades planteadas en el capítulo 3 para mejorar de la accesibilidad y que no pudieron ser integradas en un módulo independiente.



Figura 4.1: Portadas de las tres ficciones interactivas desarrolladas.

4.1 Ciclo de vida adoptado en esta fase

En esta ocasión se opta por utilizar un ciclo de vida **iterativo incremental** para desarrollar cada una de las obras, al considerarlo más ágil y más flexible ante even-

tuales cambios de diseño a lo largo del desarrollo en comparación con el ciclo de vida en cascada utilizado en el capítulo 3. Se plantean las siguientes iteraciones:

1. Camino mínimo de principio a fin de la obra, con la secuencia de comandos requeridos para alcanzar los finales principales de la historia.
2. Implementación del conjunto completo de objetos con elevada interactividad.
3. Implementación de otros objetos de decorado.
4. Si aplica, se añaden finales alternativos al camino principal, junto con los objetos adicionales requeridos por estos caminos.
5. Implementación de un comando «AYUDA» para ofrecer indicaciones sobre cómo operar el software y pistas sobre cómo avanzar en la historia.

4.2 Diseño de la estructura narrativa

Como resultado del trabajo personal final realizado en la asignatura de Lenguajes Formales: *Autómatas finitos para modelar la estructura narrativa en ficción interactiva* [Martín, 2018], se ha encontrado que esta herramienta formal puede resultar muy útil durante las fases de diseño y prueba de obras de ficción interactiva. Siguiendo lo aprendido en este trabajo, se procede a continuación a modelar la estructura de las tres obras planteadas, procurando en cada caso explorar posibilidades narrativas diferentes.

En primer lugar conviene recordar qué se conoce como **autómata finito determinista** (DFA, por sus siglas en inglés). [Gómez et al., 2013] recogen como definición de DFA a todo quíntuplo $A := (Q, \Sigma, \delta, q_0, F)$ donde:

- Q es un conjunto finito cuyos elementos se llaman estados. Suele denominarse *espacio de estados*.
- Σ es un conjunto finito de símbolos de entrada (*alfabeto*).
- $q_0 \in Q$ es el *estado inicial*.
- $F \subseteq Q$ es un conjunto de *estados finales aceptadores*.
- $\delta : Q \times \Sigma \rightarrow Q$ es una *función de transición*.

Aplicado sobre el dominio de las ficción interactiva y las herramientas a utilizar, el alfabeto Σ se corresponde con el conjunto completo de comandos contemplados por la librería Inform, más aquellos definidos en el módulo Spanish extended grammar 3.4.7 y, eventualmente, cualquier otro comando que pueda definir cada obra de manera independiente, como por ejemplo los comandos «AYUDA», en el caso de *La pequeña cerillera e Historia de Grisóstomo y Marcela*. De esta forma, las funciones de transición δ entre estados se lanzan ante comandos $x \in \Sigma$ introducidos por el usuario, de acuerdo a las reglas implementadas en el modelo de mundo de la ficción interactiva.

El principal reto está en identificar correctamente el espacio de estados en cada obra, puesto que el nivel de complejidad que puede alcanzar potencialmente una ficción interactiva es muy elevado. Tal y como se señala en [Martín, 2018]: «el espacio de estados de la obra es en realidad el conjunto de todas las permutaciones posibles de estados de todos los objetos presentes en la simulación; un conjunto muy grande como para trabajar con él cómodamente». Se puede, no obstante, contemplar únicamente el estado de aquellos objetos del modelo de mundo que afecten efectivamente al desarrollo de la narración, simplificando sustancialmente así el espacio de estados hasta unas dimensiones practicables.

4.2.1 Estructura de *La pequeña cerillera*

La pequeña cerillera es una adaptación de un cuento de Hans Christian Andersen. Procurando ser fiel al relato original, se opta por seguir una estructura lineal en la que la interacción se utilice ante todo para manipular el ritmo de la narración y para provocar una reacción emocional en el usuario; en lugar de limitarse a leer el destino —trágico— de la protagonista será el usuario quien, por medio de sus acciones, provoque que la narración avance hasta su desenlace.

El espacio de estados viene determinado por la cantidad de alucinaciones experimentadas por la protagonista, con: $q_1 = 0$ alucinaciones; $q_2 = 1$; $q_3 = 2$; $q_4 = 3$; $q_5 = 4$. Y por el hecho de si la caja de cerillas está o no abierta; conjunto de estados: $\{q_0, q_{2b}, q_{3b}, 1_{4b}\}$, puesto que las alucinaciones se muestran cada vez que la protagonista enciende un fósforo y sólo puede hacerlo en tanto la caja esté abierta.

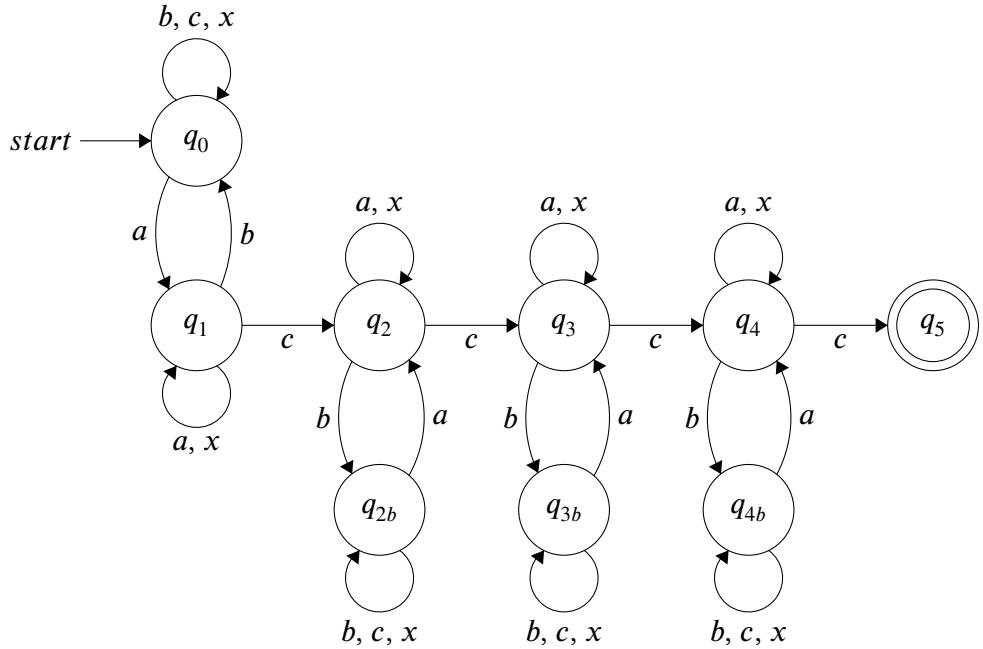
La figura 4.2 muestra una representación gráfica de la estructura narrativa modelada como un DFA, contemplando los estados anteriores. Además:

- a es el comando «ABRE CAJA DE CERILLAS».
- $b = \text{«CIERRA CAJA DE CERILLAS»}$.
- $c = \text{«ENCIENDE CERILLA»}$.
- x = cualquier otro comando válido, con $x \in \Sigma$.

4.2.2 Estructura de *Historia de Grisóstomo y Marcela*

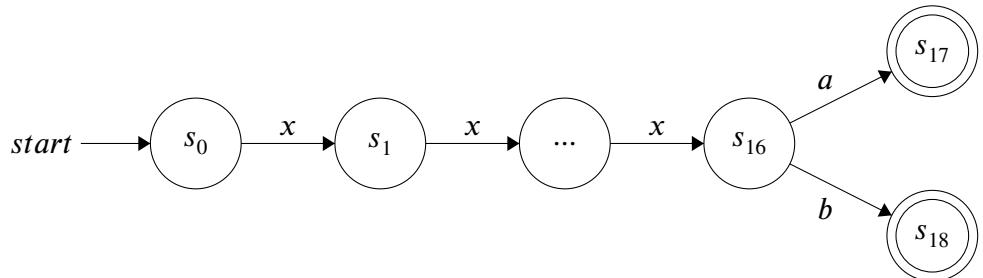
Historia de Grisóstomo y Marcela es también una adaptación literaria, en esta ocasión de un episodio narrado en la primera parte de *El Ingenioso Hidalgo Don Quijote de la Mancha*, de Miguel de Cervantes.

De nuevo se opta por utilizar una estructura narrativa esencialmente lineal como en el caso de *La pequeña cerillera*, pero se modifica en gran medida el papel que juega aquí a interacción de usuario. Siguiendo la dinámica apuntada por el material original, en el que el personaje principal cede el protagonismo de la escena a los personajes

Figura 4.2: DFA con la estructura de *La pequeña cerillera*.

secundarios, en *Historia de Grisóstomo y Marcela* se decide reforzar esta idea convirtiendo al usuario en un simple espectador que no puede modificar en nada la acción que se desarrolla automáticamente ante él; los comandos a su disposición sirven tan sólo para controlar el ritmo de la narración y para estudiar los objetos de la simulación y conocer más detalles sobre el entorno mientras que la trama avanza ajena a sus acciones, salvo en el momento justo del desenlace. Aquí se subvierte el material original y se fuerza al usuario a tomar una decisión que afectará al final de la historia –en un ejemplo de utilización del módulo Choice sets (ver 3.4.3)–.

De esta manera los estados: $\{s_0, s_1, s_2, \dots, s_{16}\}$ se corresponden con una división granular de la acción que se desarrolla automáticamente en la escena cada vez que el usuario introduce un comando, independientemente de qué comando sea éste. Los estados finales: $\{s_{17}, s_{18}\}$, sin embargo, sí dependen de la entrada introducida por el usuario, tal y como se puede ver en la figura 4.3. Así, el usuario puede optar por hacer que el personaje que controla actúe y afecte al desenlace (tal y como ocurre en el relato original), o puede continuar manteniéndose al margen de la escena y alcanzar un final alternativo.

Figura 4.3: DFA con la estructura de *Historia de Grisóstomo y Marcela*.

Las transiciones de la figura 4.3 se corresponden con:

- $a = \text{«INTERVENIR»}$.
- $b = \text{«NO HACER NADA»}$.
- $x = \text{cualquier comando válido, con } x \in \Sigma$.

4.2.3 Estructura de *La zona*

Con el último ejemplo se intenta aprovechar las capacidades del computador para intentar crear una experiencia imposible de replicar exactamente a través de un relato convencional. Con este objetivo se opta por crear una historia original, en lugar de adaptar un relato ya escrito como en las obras anteriores.

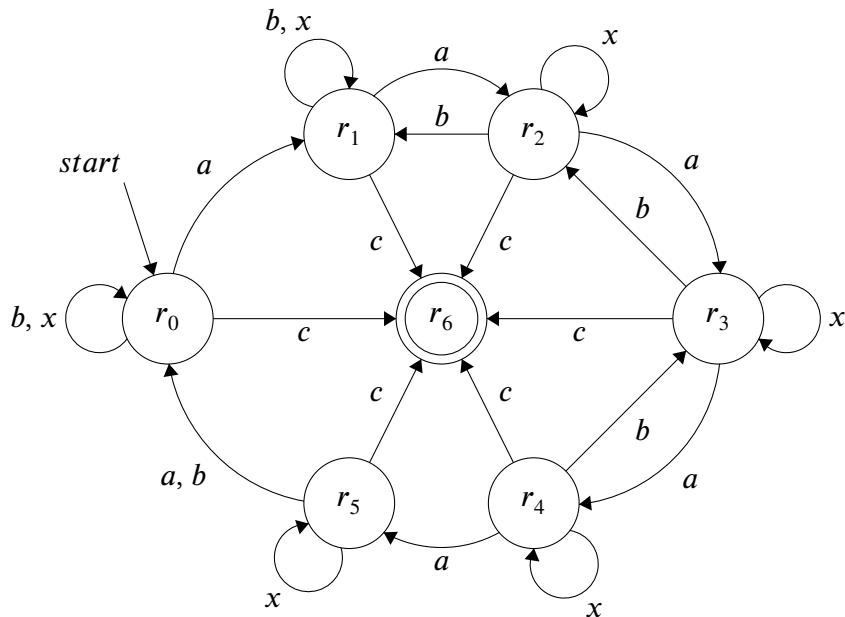


Figura 4.4: DFA con la estructura de *La zona*.

En primer lugar se plantea una estructura cíclica; llegado a un punto de la lectura la historia continuará sin cortes de nuevo desde el momento inicial. Como justificación narrativa para esta estructura se propone una historia de carácter surrealista-metáfísico que explorará temas de la mitología clásica asociados con la muerte y una vida más allá. Además, a fin de reforzar aún más el aspecto surrealista de la narración se podrán aprovechar las posibilidades abiertas por el módulo Spanish library messages3.4.8 para hacer que en cada ciclo de la historia tanto el tiempo verbal del narrador —presente, pasado y futuro— como el género —masculino y femenino— del personaje protagonista varíen de forma aleatoria.

La interacción del usuario se enfoca en la exploración del espacio de juego. De este modo, el conjunto de estados: $\{r_0, r_1, r_2, r_3, r_4, r_5\}$ se corresponde con diferentes localidades por las que el usuario puede desplazarse a través de los comandos «IR HACIA

ADELANTE» e «IR HACIA ATRÁS», de tal manera que la última localidad —asociada al estado r_5 — queda vinculada a la primera localidad —asociada a r_0 —.

Se introduce por último un desenlace —el estado r_6 — al que es posible llegar inmediatamente tras iniciar la historia. La pretensión es que mediante la exploración del espacio, lo cuál conforma el cuerpo principal de la narración, los usuarios puedan conseguir pistas sobre cuál es el comando que se espera que introduzcan para alcanzar el final de la trama —el comando «DISPÁRATE»—. En la figura 4.4 se puede ver una representación gráfica de esta estructura cíclica, donde:

- a es el comando «IR HACIA ADELANTE».
- $b = \text{«IR HACIA ATRÁS»}$.
- $c = \text{«DISPÁRATE»}$.
- $x = \text{cualquier otro comando válido, con } x \in \Sigma$.

4.3 Script de pre-compilación

Una parte importante de los módulos desarrollados en el capítulo anterior se centra en la adición de funcionalidades relacionadas con la impresión de texto; rutinas que facilitan la selección de estilos diferentes, por ejemplo, o encargadas de generar hipervínculos sobre ciertas cadenas de caracteres. A la hora de desarrollar una ficción interactiva con Inform 6, la utilización de este tipo de rutinas que afectan a la impresión de texto pueden dar lugar a secuencias de código confusas y difíciles de mantener.

Antes de abordar la implementación de las obras de ejemplo y con objeto de facilitar el proceso, se ha creado un *script* —`preprocesaTexto.pl`— para ser ejecutado antes de la compilación que traduce una serie de etiquetas (inspiradas en las utilizadas por lenguajes de marcado como *Markdown*) en rutinas del lenguaje Inform 6 y los módulos descritos en el capítulo 3:

- Las secuencias “\[”, “[”] se traducen por los caracteres ‘[’, ‘]’.
- Etiquetas `**texto**`, `*texto*` y ``texto`` para imprimir “texto” en estilo fuerte, enfatizado y código, respectivamente.
- Etiqueta `[plural: objeto]` para abrir secuencias condicionales si el “objeto” está definido como plural. `[else]` y `[fi]` para cerrar estas secuencias condicionales.
- Etiquetas para imprimir el nombre corto de un objeto junto con los artículos determinados, los artículos indeterminados y las contracciones al/del adecuados para la definición de dicho objeto.
- Etiqueta `[lista de objetos en contenedor]` para utilizar la rutina de la librería Inform encargada de listar los objetos contenidos por un objeto “contenedor”.

- Etiqueta [objeto] (texto) para crear hipervínculos asociados a un objeto que utilicen “texto” como texto alternativo.
- Etiqueta [objeto] para imprimir el nombre corto de un objeto.

La figura 4.5 muestra un ejemplo de implementación con la descripción de un objeto que hace uso de estilos de texto diferentes, varios hipervínculos, fragmentos condicionales de texto y una rutina encargada de listar cualesquiera otros objetos contenidos por el primero:

```
print (A) Comoda;
print " de madera oscura.";
if (IsPluralNoun(Cajones)) {
    print " Tiene varios cajones";
} else {
    print " Tiene ", (emph) "un único", " cajón";
}
print ", de ";
PRT_("pomos");
print " agresivamente ", (strong) "feos.";
print " Sobre ella puedes ver ";
WriteListFrom(child(Comoda), ENGLISH_BIT);
print ".";
```

Figura 4.5: Implementación por defecto con la descripción de un objeto.

La figura 4.6 muestra una implementación de esta misma descripción, pero utilizando las etiquetas del *script*:

```
print "[Una Comoda] de madera oscura.[plural:Cajones] Tiene varios
cajones[else] Tiene *un único* cajón[fi], de [](pomos) agresivamente
**feos**. Sobre ella puedes ver [lista de objetos sobre Comoda].";
```

Figura 4.6: Descripción de un objeto usando el *script* de pre-compilación.

4.4 Sistema de sintaxis simplificada

De acuerdo a la especificación de requisitos (ver 3.3), a través de los módulos no fue posible terminar de implementar el sistema de sintaxis simplificada. En este apartado se detalla el proceso seguido para completarlo.

Para empezar se hace necesario definir una nueva acción *falsa*, que no puede ser invocada directamente por el usuario, pero que se lanzará por defecto al introducir el

nombre de un objeto del modelo de mundo sin incluir con él un verbo. En la figura 4.7 se adjunta una posible implementación en la que la acción por defecto se limita a lanzar la acción ##Examine¹:

```
Verb 'no.verb'
    * noun -> DefaultAction
;

[ DefaultActionSub;
    <<Examine noun>>;
];
;
```

Figura 4.7: Posible implementación de la acción ##DefaultAction.

Como se ha señalado, este comando no puede ser utilizado directamente por el usuario. En su lugar debe ser lanzado desde el punto de entrada UnknownVerb() de la librería Inform; una rutina que es invocada por el sistema cuando se encuentra ante un verbo desconocido, de forma que los autores puedan hacer uso de ella para reinterpretar la entrada de usuario por un verbo definido en la obra. Es en este punto en el que entra en juego el módulo Parsing_preprocessor3.4.6 diseñado en el capítulo anterior, tal y como se puede observar en la figura 4.8.

```
[ UnknownVerb
    obj;
    obj = ParsingPreprocessor.get_selected_object();
    if (TestScope(obj, player)) {
        verb_wordnum = 0;
        return 'no.verb';
    }

    // [...] La rutina puede implementar lógica adicional.

    return false;
];
;
```

Figura 4.8: Posible implementación de la rutina UnknownVerb.

Por último hace falta modificar un segundo punto de entrada: PrintVerb(), utilizado por el sistema para imprimir los verbos de los comandos al formar mensajes del tipo: «*Intenta de nuevo porque sólo se pudo entender que quieres <verbo>.*». Una vez más se adjunta una posible implementación en la figura 4.9.

¹De esta manera, que con el sistema de sintaxis simplificado el comando «EXAMINA CASA» podría simplificarse y utilizar, en su lugar, simplemente «CASA».

```
[ PrintVerb word;
  if (word == 'no.verb') {
    print "hacer algo con";
    return true;
  }

  // [...] La rutina puede implementar lógica adicional.

  return false;
];
```

Figura 4.9: Posible implementación de la rutina PrintVerb.

4.4.1 Hipervínculos que lanzan la sintaxis simplificada

Como último añunte al sistema de sintaxis simplificada, si se está utilizando el módulo Hyperlinks3.4.5 es posible configurarlo sencillamente de manera que los hipervínculos invoquen la acción por defecto definida en el sistema anterior. Para hacer esto basta con establecer el comando por defecto de los hipervínculos como vacío, tal y como se puede observar en la figura ??.

```
Global _hyperlinks_command = "";
```

Figura 4.10: Ejemplo de definición del comando por defecto para los hipervínculos.

Las tres ficciones interactivas desarrolladas durante esta fase implementan estas funcionalidades.

4.5 Sistema de ayuda

Los sistemas de ayuda en tiempo de ejecución son otro tipo de funcionalidad que pueden resultar muy útiles de cara a mejorar la accesibilidad de las ficciones interactivas. Es posible implementar estos sistemas de diferentes maneras: ofreciendo un nuevo comando «AYUDA» o «PISTA» dedicado, a través de alguna entrada de menú, u observando la actividad del usuario con la obra y mostrando ayudas útiles en función de su forma de interactuar con la obra, por ejemplo. Tanto en *La pequeña cerillera* como *Historia de Grisóstomo y Marcela*² se incluye un comando «AYUDA» con indicaciones útiles para el usuario.

²En el caso de *La zona* no ha sido posible desarrollar un sistema de ayuda debido a una limitación temporal. La versión inicial de la obra se ha creado a contrarreloj para poder presentarla a *Rayuela de arena I*; un evento y competición por internet de videojuegos narrativos celebrado en agosto de 2018. El desarrollo del sistema de ayuda queda propuesto junto a otras correcciones generales de la obra para una revisión futura.

En las figuras 4.11 y 4.12 se puede ver un ejemplo de este comando en acción.

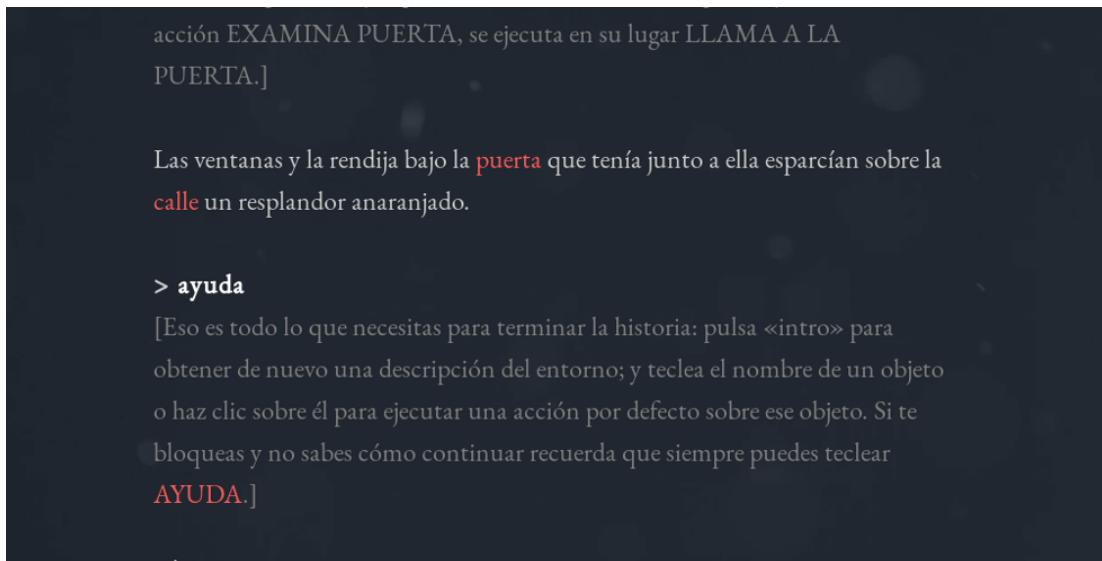


Figura 4.11: Captura de pantalla de *La pequeña cerillera* (2018).



Figura 4.12: Captura de pantalla de *Historia de Grisóstomo y Marcela* (2018).

5 | Conclusiones y trabajo futuro

You have ensured that others will hear the tape – *Would you like to RESTART, RESTORE a saved game, QUIT, UNDO the last command, or see DESIGN notes?*

«Te has asegurado de que otros vayan a oír la cinta – *¿Deseas REINICIAR, RECUPERAR una partida guardada, SALIR, DESHACER el último comando, o ver notas de DISEÑO?*»

—COLORATURA (Lynnea Glasser, 2013)

Se recogen a continuación un conjunto de conclusiones extraídas tras la elaboración de proyecto y se señalan algunas líneas posibles para continuar con el trabajo.

5.1 Grado de consecución de objetivos

Más allá de los objetivos planteados en 1.3, el trabajo ha servido como pretexto para poner en práctica y profundizar en un amplio conjunto de conocimientos y habilidades. A este respecto cabría citar:

- Experimentación con la aplicación de dos enfoques distintos de desarrollo software: el modelo en cascada (tal y como se ha descrito en el capítulo 3) y el modelo iterativo incremental (ver capítulo 4).
- Familiarización con el estilo de programación de otros autores. Al desarrollar el software descrito en el trabajo se ha seguido el ejemplo de estilo utilizado en la librería base del sistema de autoría Inform 6, de forma que se ha procurado adoptar un enfoque más estructurado que orientado a objetos (con el que previamente se tenía más experiencia).
- Oportunidad de trabajar en la construcción colectiva de conocimiento a través de un proyecto colaborativo de gran envergadura, sustentado por una comunidad numerosa de desarrolladores. Los módulos descritos en el capítulo 3 se han publicado con un licencia libre y se añaden al amplio ecosistema de herramientas publicadas bajo diferentes licencias para extender las funcionalidades del sistema de autoría Inform 6.

- Profundización en la naturaleza de las principales barreras de accesibilidad con las que se pueden encontrar los usuarios de un determinado software, y explotación de posibles formas de minimizarlas. Si bien el estudio se ha centrado en el software de ficción interactiva, muchas de las barreras y decisiones de diseño adoptadas para atenuarlas podrían aplicarse en otros proyectos distintos.
- Investigación del medio de la ficción interactiva: su caracterización, historia y evolución, analizando algunas de las obras más reseñables y el modo en que otros autores exploran las posibilidades narrativas abiertas por el lenguaje interactivo; desde las aventuras conversacionales clásicas como la ya citada *Zork* (Infocom, 1979) —convertidas hoy en día en parodia¹— a experimentos interactivos como *Nemean Lion* (A. Cadre, 2008)².

En cuanto a los dos objetivos concretos del TFG, de acuerdo al trabajo descrito en los capítulos 3 y 4 se estima que han sido cumplidos con éxito. Dos de los módulos desarrollados, en especial, pueden resultar en una mejora importante de la accesibilidad y la usabilidad de la ficción interactiva creada con Inform 6 con apenas esfuerzo adicional por parte de los autores:

- Módulo *Hyperlinks* (3.4.5) para la interacción con la obra de ficción interactiva a través de hipervínculos. Si bien cualquier obra compilada para la máquina virtual *Glulx* podría potencialmente implementar esta funcionalidad, en la práctica se ha observado que son muy pocas las obras que lo hacen efectivamente³. Este módulo simplifica significativamente al autor la utilización de hipervínculos, tal y como se demuestra en las tres obras descritas en el capítulo 4.
- Módulo *GWindows GUI* (3.4.4), con la definición de una interfaz de ficción interactiva que adapta el ancho de los párrafos de texto a las dimensiones de la ventana principal y es configurable por el autor. Habitualmente los autores pueden configurar cuestiones como el tamaño de letra, interlineado, contrastes de colores,... pero no el ancho de los párrafos de texto, y esto afecta en gran medida a la legibilidad de una ficción interactiva⁴.

Por otra parte, entre los objetivos se ponía un énfasis especial en las cuestiones de distribución y demostración del trabajo llevado a cabo entre la comunidad de autores

¹En la serie de televisión *The Big Bang Theory* (episodio 6 de la cuarta temporada) se hace comedia a partir de las mecánicas de juego y las importantes limitaciones técnicas de estas primeras ficciones interactivas.

²En *Nemean Lion*, Cadre juega con la granularidad de la acción y la idea preconcebida por parte del usuario sobre cómo se desarrolla una aventura —usa un escenario similar al de *Zork*, intercambiando el troll por un león—. El jugador puede finalizar la historia con una secuencia de comandos del tipo: “empuja roca”, “entra en la cueva”, “mira al león”, “golpea al león con la maza” y “corta al león con la espada” (que es, de hecho, la secuencia habitual de cualquiera en una primera partida); sin embargo, el usuario podría haber empezado tecleando simplemente “mata al león” al inicio de la historia, y ésta se hubiese desarrollado automáticamente hasta su desenlace.

³En obras creadas con Inform no se han encontrado más ejemplos aparte de *El anillo 3* (J. Antonio Paz, 2009) que los utilicen de manera general.

⁴J. Antonio Paz —autor de *El anillo 3*, citada anteriormente—, se ha puesto en contacto con el deseo de adaptar el módulo *GWindows GUI* desde Inform 6 a la versión 7 del sistema.

y aficionados. Con propósito de cubrir este punto se han puesto a disposición pública a través de Internet y bajo una licencia de software libre GNU GPLv3 tanto los módulos como las obras desarrolladas:

- Espacio web de *La pequeña cerillera*:
<https://jomali.itch.io/la-pequenna-cerillera>.
- Espacio web de *Historia de Grisóstomo y Marcela*:
<https://jomali.itch.io/grisostomo-marcela>.
- Espacio web de *La zona*:
<https://jomali.itch.io/la-zona>.
- Repositorio de código fuente (*La pequeña cerillera*):
<https://github.com/jomali/la-pequenna-cerillera>.
- Repositorio (*Historia de Grisóstomo y Marcela*):
<https://github.com/jomali/grisostomo-marcela>.
- Repositorio (*La zona*):
<https://github.com/jomali/la-zona>.
- Repositorio (módulos):
<https://github.com/jomali/i6-libs/tree/master/Extensions>.

Se ha dedicado además un esfuerzo significativo en participar y hacerse eco del trabajo en diferentes foros y espacios que tratan el diseño y desarrollo de videojuegos en general, y la ficción interactiva en particular: desde portales web⁵, chats públicos y redes sociales⁶ a la participación en varios eventos y competiciones (que se detallan a continuación). Fruto de la experiencia obtenida en el proceso se opina que **las obras de ficción interactiva son capaces de competir en igualdad de condiciones con otros géneros populares de videojuegos** por despertar el interés de los jugadores.

Este hecho, el que los usuarios encuentren atractiva la ficción interactiva independientemente de si cuentan o no con algún tipo de discapacidad, se aventura fundamental de cara a utilizarlas como herramienta de inclusión digital.

5.1.1 *Rayuela de Arena I*

Rayuela de Arena I fue una *gamejam* dedicada a los juegos narrativos celebrada a lo largo de los meses de agosto y septiembre de 2018. Siguiendo la temática mitológica del evento se participó con *La zona*, junto a otros juegos de naturaleza diversa; desde otras ficciones interactivas, a juegos de mesa, RPGs y videojuegos de aventura en 3D. Durante la fase de votación popular *La zona* obtuvo la posición 4 de 21 [Aguilera and Sánchez, 2018].

⁵Se ha participado en las comunidades de habla inglesa y francesa: <https://intfiction.org/> y <http://www.fiction-interactive.fr/>, respectivamente. Y, por supuesto, en la comunidad de habla hispana: <http://www.caad.es/>.

⁶Además de en *Facebook* y *Twitter*, se ha mantenido contacto con diversos grupos de aficionados a través de servidores dedicados a la ficción interactiva y los videojuegos en las plataformas *Euphoria* y *Discord*.

5.1.2 *Jam Cultura Abierta*

Historia de Grisóstomo y Marcela se desarrolló entre los días 16 al 18 de noviembre de 2018 coincidiendo con la celebración de la *Jam Cultura Abierta*, un evento/competición de creación de videojuegos a partir de contenido cultural de dominio público organizada por la Asociación Española de Empresas Productoras y Desarrolladoras de Videojuegos y Software de Entretenimiento (DEV) con el respaldo del Ministerio de Cultura y Deporte[DEV, 2018a], y del que se hicieron eco diferentes medios especializados⁷.

De entre los 30 juegos desarrollados durante la *jam*, un jurado de expertos «formado por Tatiana Delgado, fundadora y Game Designer en *Out of the Blue Games*; Juanma Moreno, codirector de *Nivel Oculto* y director de *Indie[MAD]*, Rocío Tomé, psicóloga Clínica y programadora de VR/AR en *Minsait*. Cofundadora de *Femdevs*; y Luis Oliván, productor en *Fictiorama Studios* y Vocal de la junta directiva de DEV»[Ruete, 2018a] seleccionó 8 para ser presentados ante el público. *Historia de Grisóstomo y Marcela* formó parte de esta selección.

La exposición, inaugurada el día 23 de enero de 2019 con un acto presentado por Eduardo A. Leyva, vocal asesor de la Subdirección General de Propiedad Intelectual del Ministerio de Cultura y Deporte, y por el escritor y diseñador británico de videojuegos Greg Buchanan, se mantuvo abierta al público hasta el día 30 de ese mismo mes.

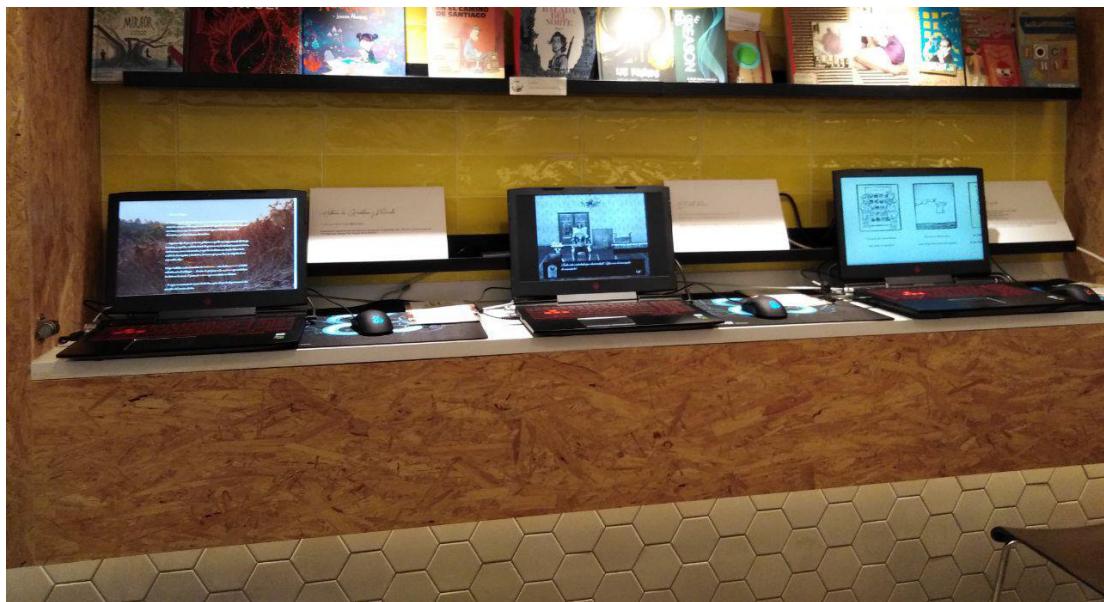


Figura 5.1: *Historia de Grisóstomo y Marcela* en la exposición de *Jam Cultura Abierta*.

⁷Blogs de noticias sobre videojuegos como *Anait Games*[Trivi, 2018], *Hobby Consolas*[Ortega, 2018] o *Meristation*[Ruete, 2018b] publicaron información sobre el evento.

5.1.3 Exposición: Ficción interactiva y juegos narrativos

La pequeña cerillera fue seleccionada por la organización junto a un conjunto de otras cinco ficciones interactivas como exponentes del medio en lengua española para una exposición celebrada el día 19 de marzo de 2019 y organizada por la Facultad de Filosofía y Letras y E.T.S. de Ingenierías Informática y de Telecomunicación de la Universidad de Granada[UGR, 2019].

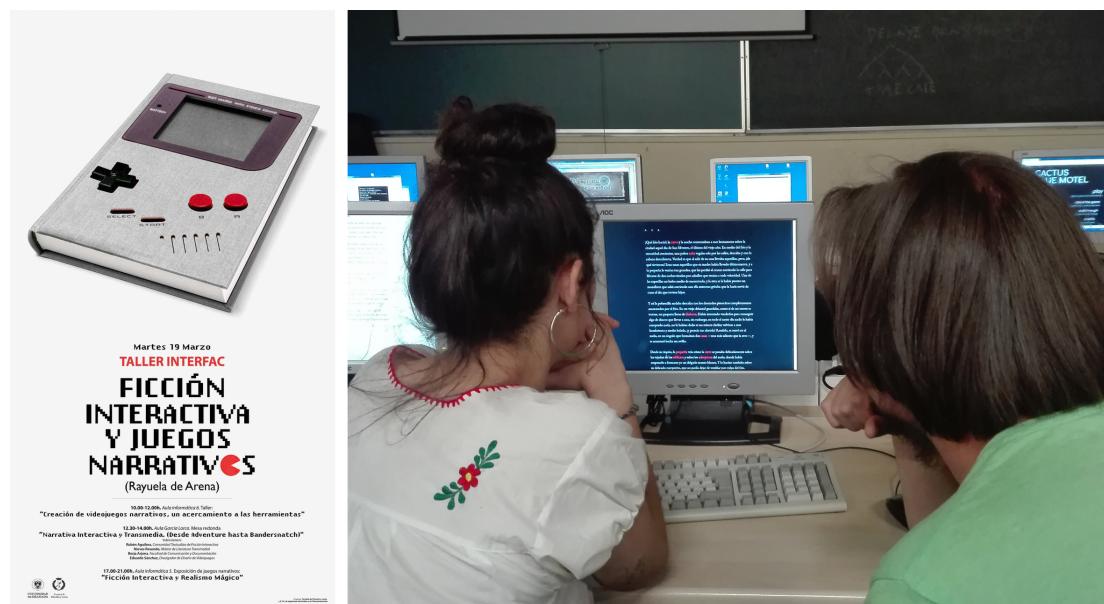


Figura 5.2: *La pequeña cerillera* en la exposición *Ficción interactiva y juegos narrativos*.

5.2 Líneas de trabajo futuro

Actualmente se sigue trabajando en varias ficciones interactivas iniciadas junto a las tres obras descritas en el TFG, y que se encuentran en diferentes fases de desarrollo:

- La primera de estas obras, titulada provisionalmente *El palacio de la memoria*, pretende continuar la línea de experimentación apuntada por *La zona* —así como por algunos videojuegos como *The Path* (2009), de la compañía belga Tale of Tales— y utilizar un planteamiento narrativo estrictamente basado en la exploración libre por parte del usuario; se le ofrece la opción de investigar un cierto entorno a través del que se desarrolla el nudo narrativo, pero desde un primer momento se ofrece igualmente la oportunidad de abandonarlo y alcanzar de esta manera el desenlace de la historia. El usuario es libre así de decidir hasta qué punto desea profundizar en la obra e, idealmente, alcanzar una experiencia narrativa satisfactoria sea cuál sea su nivel de implicación.
- Adaptación de *De Baron* (2006) de Victor Gijsbers, profesor de Filosofía en la

universidad de Leiden⁸. *De Baron*, que actualmente cuenta con versiones en inglés y holandés, es una de las ficciones interactivas mejor valoradas por la comunidad[IFDB, 2011][IFDB, 2015]. No persigue tanto el mero entretenimiento como reflexionar y hacer reflexionar a través del juego de identificación de roles y la interactividad sobre cuestiones como la moralidad de la violencia, el abuso infantil o el concepto de la maldad, entre otros.

- Proyecto de colaboración con el estudio independiente de videojuegos *cKolmos* para la creación de una obra titulada *La Roka*. A través de un enfoque lúdico se pretende hablar de la cuestión de la mujer en el deporte (representatividad, prejuicios, tabús —la menstruación en el deporte de alto nivel—, financiación, etc.).

En adición a estos proyectos para la creación de nuevas obras: durante el TFG los esfuerzos para dar a conocer y demostrar las posibilidades del trabajo desarrollado se han focalizado en las principales comunidades de aficionados a la ficción interactiva y los videojuegos narrativos, dado que es aquí donde más se concentra el mayor número de autores y mayores eran, por tanto, las posibilidades de incentivar el desarrollo de nuevas obras que utilicen los módulos generados. Ya es habitual encontrar personas con diversidad funcional dentro de estas comunidades, al tratarse de un tipo de videojuegos que en general suelen resultar más accesibles que otros géneros de mayor popularidad, aún así cabe explorar la posibilidad de ponerse en contacto con otras comunidades y asociaciones, como tal vez el Comité Español de Representantes de Personas con Discapacidad (CERMI), la red social accesible *Blindworlds* —<https://www.blindworlds.com/>— o el espacio web de *Tiflojuegos* —<http://www.tiflojuegos.com/>— dedicado a los audiojuegos y videojuegos accesibles para personas con discapacidad visual, por ejemplo, a través de las que llegar a más personas potencialmente interesadas en el trabajo.

Se considera igualmente un hilo de investigación interesante el profundizar en las posibilidades didácticas del medio, quizás en la línea de lo que apuntan desde *Anait Games* al hablar de *Historia de Grisóstomo y Marcela*: «recurre a la primera parte del socorrido Quijote y la convierte en una ficción interactiva que actualiza los textos⁹ y que bien podrían usarse como herramienta educativa en muchas clases de Lengua Castellana»[Trivi, 2019].

⁸El autor de *De Baron* ha dado su consentimiento para acometer la tarea de traducción y adaptación de la obra, facilitando por privado el código fuente original.

⁹En realidad el cuerpo principal del texto en *Historia de Grisóstomo y Marcela* está extraído directamente de la obra original sin ninguna actualización. Sólo se han añadido descripciones ante ciertas acciones introducidas por el usuario para las que Cervantes no contempla respuesta —lo cuál da fe de lo sorprendentemente actual que sigue resultando esta novela—.

Bibliografía

- [GAG, 2019] (2019). Game accessibility guidelines. <http://gameaccessibilityguidelines.com/>. (Web) Consultado: 2019/03/26.
- [Aguilera and Sánchez, 2018] Aguilera, R. and Sánchez, E. (2018). Resultados de Rayuela de Arena – Gamejam Narrativa. <https://itch.io/jam/rayuela-de-arena-gamejam-narrativa/results>. (Web) Consultado: 2019/05/25.
- [Alvarez, 2003] Alvarez, A. S. (2003). Behavioral and environmental correlates of digital inequality. *IT & Society*.
- [Bakal, 2005] Bakal, M. (2005). UML for C Programmers. <http://www.drdobbs.com/cpp/uml-for-c-programmers/184401948>. (Web) Consultado: 2019/05/17.
- [Cadre, 2002] Cadre, A. (2002). Gull. <http://adamcadre.ac/gull/gull-2o.html>. (Web) Consultado: 2018/04/19.
- [Cousins, 2019] Cousins, R. F. (2019). L'arroseur Arrose - Film (Movie) Plot and Review. <http://www.filmreference.com/Films-Am-Aw/L-Arroseur-Arrose.html>. (Web) Consultado: 2019/05/23.
- [DEV, 2018a] DEV (2018a). Game jam cultura abierta. <http://www.dev.org.es/es/noticias-a-eventos/eventos-dev/item/313-game-jam-cultura-abierta>. (Web) Consultado: 2019/05/30.
- [DEV, 2018b] DEV (2018b). *Libro Blanco del Desarrollo Español de Videojuegos 2018*. DEV - Desarrollo Español de Videojuegos, Madrid, España.
- [EA, 2018] EA (2018). EA: Accessibility Portal. <https://www.ea.com/able>. (Web) Consultado: 2019/03/26.
- [Eve, 2009] Eve, E. (2009). A Comparison of TADS 3 and Inform 7. <http://www.brasslantern.org/writers/iftheory/tads3andi7.html>. (Web) Consultado: 2018/09/06.
- [Eve et al., 2008] Eve, E., Roberts, M., Breslin, S., Nizette, M., and Sewe, A. (2008). TADS 3 Technical Manual: Choosing a Conversation System. <http://www.tads.org/t3doc/doc/techman/convbkg.htm>. (Web) Consultado: 2018/08/21.

- [Fisher, 1984] Fisher, W. R. (1984). Narration as a human communication paradigm: The case of public moral argument. *Communication Monographs*, 51(1):1–22.
- [Forbes, 2019] Forbes (2019). The Richest People in the World. <https://www.forbes.com/billionaires/>. (Web) Consultado: 2019/03/23.
- [GAconf, 2019] GAconf (2019). #GAconf: Advancing accessibility for gamers with disabilities. <https://www.gaconf.com/>. (Web) Consultado: 2019/03/26.
- [García Muñoz, 2012] García Muñoz, O. (2012). *Lectura fácil: métodos de redacción y evaluación*. Centro Español de Documentación sobre Discapacidad, Serrano, 140. 28006 Madrid.
- [Gómez et al., 2013] Gómez, D., Pardo, L. M., and Tîrnăucă, C. (2013). “*Lenguajes formales*” (para Ingenieros Informáticos). Universidad de Cantabria, Cantabria.
- [IFDB, 2011] IFDB (2011). Interactive fiction top 50 of all time (2011 edition). <https://ifdb.tads.org/viewcomp?id=oymvom4wrawhd4hr>. (Web) Consultado: 2019/05/30.
- [IFDB, 2015] IFDB (2015). Interactive fiction top 50 of all time (2015 edition). <https://ifdb.tads.org/viewcomp?id=p6s9uem6td8rfihv>. (Web) Consultado: 2019/05/30.
- [Lange et al., 2018] Lange, G.-M., Wodon, Q., and Carey, K. (2018). *The Changing Wealth of Nations 2018: Building a Sustainable Future*. World Bank, Washington, DC, USA.
- [Marcos and Macías, 2019] Marcos, L. and Macías, I. (2019). Desigualdad 1 – igualdad de oportunidades 0. *Informe de Oxfam Intermón*, (49).
- [Martin, 2003] Martin, R. C. (2003). *Agile Software Development: Principles, Patterns, and Practices*. Prentice Hall PTR, Upper Saddle River, NJ, USA.
- [Martín, 2018] Martín, J. F. (2018). Autómatas finitos para modelar la estructura narrativa en ficción interactiva. *Trabajo personal para la asignatura de Lenguajes Formales*. Universidad de Cantabria.
- [Meyer, 1997] Meyer, B. (1997). *Object-oriented Software Construction* (2Nd Ed.). Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- [Montfort, 2004] Montfort, N. (2004). *Twisty Little Passages: An Approach to Interactive Fiction*. MIT Press, Cambridge, MA, USA.
- [Nelson, 2001] Nelson, G. (2001). *The Inform Designer’s Manual* (4th Ed.). Dan Sanderson, UK.
- [Nelson, 2018] Nelson, G. (2018). Inform: Past, present, future. <http://www.emshort.com/ifmu/inform.html>. (Web) Consultado: 2018/09/07.

- [Nelson and Fillmore, 2014] Nelson, G. and Fillmore, D. (2014). The Z-Machine Standards Document. Version 1.1. <https://inform-fiction.org/zmachine/standards/z1point1/index.html>. (Web) Consultado: 2018/09/06.
- [Niesz and Holland, 1984] Niesz, A. J. and Holland, N. N. (1984). Interactive fiction. *Critical Inquiry*, 11(1):110–129.
- [Ortega, 2018] Ortega, J. L. (2018). La I Game Jam Cultura Abierta reúne a más de 100 desarrolladores. Hobby Consolas. <https://www.hobbyconsolas.com/noticias/i-game-jam-cultura-abierta-reune-mas-100-desarrolladores-332279>. (Web) Consultado: 2019/05/30.
- [Parsons and Hick, 2009] Parsons, C. and Hick, S. (2009). Moving from the digital divide to digital inclusion. *Currents: Scholarship in the Human Services*, 7(2).
- [Plotkin, 2014] Plotkin, A. (2014). Glulx. A 32-Bit Virtual Machine for IF. VM specification version 3.1.2. <https://www.eblong.com/zarf/glulx/>. (Web) Consultado: 2018/09/06.
- [Plotkin, 2017a] Plotkin, A. (2017a). Glk API Specification. API version 0.7.5. https://eblong.com/zarf/glk/glk-spec-075_4.html#s.8. (Web) Consultado: 2018/08/20.
- [Plotkin, 2017b] Plotkin, A. (2017b). Glk API Specification. API version 0.7.5. https://eblong.com/zarf/glk/glk-spec-075_5.html#s.5. (Web) Consultado: 2018/08/20.
- [Raszewski, 2004] Raszewski, L. R. (2004). Gwindows: A screen management framework for glulx inform. (Documentación de la librería).
- [Riedl et al., 2006] Riedl, M. O., Stern, A., and Dini, D. M. (2006). Mixing story and simulation in interactive narrative. In *The Artificial Intelligence for Interactive Digital Entertainment Conference (AIIDE)*.
- [Ruete, 2018a] Ruete, B. (2018a). Don Quijote y Lazarillo inspiran algunas de las obras seleccionadas en Game Jam Cultura Abierta. MeriStation. https://as.com/meristation/2018/11/26/noticias/1543251821_558724.html. (Web) Consultado: 2019/05/30.
- [Ruete, 2018b] Ruete, B. (2018b). Game Jam Cultura Abierta: juegos creados usando contenido cultural de dominio público. MeriStation. https://as.com/meristation/2018/11/07/noticias/1541612552_928259.html. (Web) Consultado: 2019/05/30.
- [Statista, 2016] Statista (2016). Number of smartphone users worldwide from 2014 to 2020 (in billions). <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>. (Web) Consultado: 2019/03/25.

- [Statista, 2018] Statista (2018). Share of households with a computer at home worldwide from 2005 to 2018. <https://www.statista.com/statistics/748551/worldwide-households-with-computer/>. (Web) Consultado: 2019/03/23.
- [Straub et al., 2019] Straub, J., Peeples, J., Hammond, M., and Matlock, M. (2019). DA-GERS. <https://dagersystem.com/>. (Web) Consultado: 2019/03/26.
- [THOCP, 2013] THOCP (2013). Univac: Universal automatic computer. <https://www.thocp.net/hardware/univac.htm>. (Web) Consultado: 2019/05/23.
- [Trivi, 2018] Trivi, M. (2018). La escena indie reivindica la importancia de la propiedad intelectual en la exposición Cultura Abierta. Anait Games. <https://www.anaitgames.com/noticias/la-game-jam-cultura-abierta-nos-invita-a-crear-juegos-usando-contenido-cultural-de-dominio-publico>. (Web) Consultado: 2019/05/30.
- [Trivi, 2019] Trivi, M. (2019). La escena indie reivindica la importancia de la propiedad intelectual en la exposición Cultura Abierta. Anait Games. <https://www.anaitgames.com/articulos/la-escena-indie-reivindica-la-importancia-de-la-propiedad-intelectual-en-la-exposicion-cultura-abier>. (Web) Consultado: 2019/05/30.
- [UGR, 2019] UGR (2019). Taller INTERFAC: “Ficción interactiva y Juegos Narrativos (Rayuela de Arena)”. https://filosofiayletras.ugr.es/pages/tablon*/actividades-culturales/taller-interfac-ficcion-interactiva-y-juegos-narrativos-rayuela-de-arena. (Web) Consultado: 2019/05/25.