



*ugr* | Universidad  
de **Granada**

TRABAJO FIN DE GRADO  
INGENIERÍA EN INFORMÁTICA

# Diseño y desarrollo de un juego de tipo aventura conversacional para asistentes virtuales

---

**Autor**

Aurelia Nogueras Lara

**Director**

Francisco Luis Gutiérrez Vela



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE  
TELECOMUNICACIÓN

—  
Granada, julio de 2020



# **Diseño y desarrollo de un juego de tipo aventura conversacional para asistentes virtuales**

Aurelia Nogueras Lara

**Palabras clave:** aventura conversacional, asistente virtual, *skill, action, plataforma, aplicación, juego, entretenimiento, Alexa, Google Assistant, videojuego, intent, slot, entity, utterance, interfaz, invocación, oral, controles, narrativa, interacción.*

## **Resumen**

La tecnología ocupa un lugar cada vez más importante dentro de la sociedad. Prueba de ello es la proliferación de dispositivos como los asistentes virtuales, que van aumentando su presencia en nuestro día a día de forma progresiva. El objetivo de este proyecto es analizar los asistentes en el ámbito del entretenimiento y desarrollar una aventura conversacional basada en la interacción oral.

Para lograr este propósito, se ha utilizado una metodología ágil, basada en la elaboración de prototipos utilizables de forma iterativa. De este modo, el proyecto se ha caracterizado por su flexibilidad y adaptación constantes. Además, se ha combinado dicha metodología con un diseño centrado en el usuario. Esto ha permitido tener en cuenta las necesidades y deseos de los potenciales jugadores en todo momento. El diseño también ha guardado cierto parecido con las fases de producción de un videojuego, dadas sus similitudes.

Este proceso ha permitido obtener un prototipo de juego bastante completo, disponible para los asistentes *Alexa* y *Google Assistant* tanto en español como en inglés.

En conclusión, el resultado final ha sido satisfactorio, cumpliendo los objetivos inicialmente propuestos. El desarrollo de este proyecto ha permitido indagar en la interacción de los usuarios con los juegos de los asistentes virtuales. Además, el prototipo desarrollado ofrece la posibilidad de elaborar un trabajo futuro en el que se añadan mejoras y nuevas funcionalidades al juego.

## Design and development of a conversational adventure in virtual assistants

Aurelia Nogueras Lara

**Keywords:** conversational adventure, virtual assistant, *skill, action, platform, app, game, entertainment, Alexa, Google Assistant, video game, intent, slot, entity, utterance, interface, invocation, oral, controls, narrative, interaction.*

### Abstract

New technologies are exponentially increasing their popularity nowadays. The spread of gadgets, such as virtual assistants, is a clear proof of this. The aim of this project is to analyze assistants regarding the field of entertainment and to develop a conversational adventure based on oral interaction.

In order to achieve this objective, agile development has been used, based on the iterative development of usable prototypes. Therefore, the project has been remarkable in its constant flexibility and adaptability. Furthermore, agile development has been combined with a user-centered design. As a result of this, the needs and desires of the users have been considered at every moment. The design is also similar to the production stages of video game development.

Consequently, this process has obtained a complete prototype of the game, available for *Alexa* and *Google Assistant* in Spanish and English.

In conclusion, the final result has been satisfactory, achieving the initially proposed objectives. The development of this project has deepened the interaction of users with virtual assistants games. Moreover, the developed prototype offers the possibility of a future work based on its improvement and the adding of new functionalities.

---

Yo, **Aurelia Nogueras Lara**, alumna de la titulación Grado en Ingeniería Informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 77552127Z, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Firmado por  
NOGUERAS LARA  
AURELIA MARIA -  
77552127Z el día  
07/07/2020 con un  
certificado emitido

Fdo: Aurelia Nogueras Lara

Granada a 7 de julio de 2020 .

---

D. **Francisco Luis Gutiérrez Vela**, Profesor del Área de Lenguajes y Sistemas Informáticos del Departamento Lenguajes y Sistemas Informáticos de la Universidad de Granada.

**Informa:**

Que el presente trabajo, titulado *Diseño y desarrollo de un juego de tipo aventura conversacional para asistentes virtuales*, ha sido realizado bajo su supervisión por **Aurelia Nogueras Lara**, y autorizo la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expide y firma el presente informe en Granada a 7 de julio de 2020.

**El director:**

A handwritten signature in black ink, appearing to read "Francisco Luis Gutiérrez Vela". The signature is fluid and cursive, with some stylized letters.

**Francisco Luis Gutiérrez Vela**

# Agradecimientos

Tras unos meses algo difíciles, al fin puedo ver cómo concluye este proyecto y, con ello, cómo se acerca el final de mi carrera universitaria. Ha sido una etapa nueva e indudablemente enriquecedora. Ha estado repleta de obstáculos que me han permitido adquirir un aprendizaje muy valioso. Me gustaría, una vez finalizado este trabajo, agradecer a aquellos que me han acompañado durante estos años de un modo u otro.

En primer lugar, quiero darle las gracias al director de este TFG, Francisco Luis Gutiérrez Vela, por proponer este proyecto y hacer un seguimiento constante de mis avances durante el proceso de desarrollo.

Este trabajo marca el cierre de una parte importante de mi vida, y quiero dedicárselo a unas cuantas personas de mi entorno.

A mis amigos, por ser partícipes de esta etapa y ayudarme cuando lo he necesitado. Especialmente a Marina e Irene, junto con las que he compartido alegrías y tristezas, pero sobre todo buenos momentos. Gracias por acompañarme durante este recorrido y constituir una parte fundamental de mi vida.

A mi padre, por estar siempre a mi lado y apoyarme en mis decisiones. Me has enseñado a seguir adelante y me has demostrado tu gran cariño y humildad.

A mi madre, por enseñarme a ser fuerte y ser mi referente. Aunque no pueda compartir contigo este momento, sé que me acompañas.

A mi hermano, por ayudarme y quererme tanto. Ver cómo logras tus metas y triunfas es un gran orgullo. Eres toda una fuente de inspiración.

A mi pareja, Antonio, por enseñarme tanto en tan poco tiempo. Gracias por aceptarme incluso cuando yo no lo he hecho. Eres una gran persona y no podría haber tenido más suerte al encontrarte.

# Índice general

<b>1. Introducción</b>	<b>15</b>
1.1. Objetivo y subobjetivos . . . . .	15
1.2. Capítulos de la memoria . . . . .	16
<b>2. Estado del arte</b>	<b>18</b>
2.1. Antecedentes teóricos . . . . .	18
2.2. Asistentes virtuales . . . . .	19
2.2.1. Tipos de asistentes . . . . .	20
2.2.2. Integración de los asistentes . . . . .	21
2.2.3. Siri . . . . .	21
2.2.4. Alexa . . . . .	22
2.2.5. Google Assistant . . . . .	23
2.2.6. Cortana . . . . .	23
2.2.7. Bixby . . . . .	24
2.3. Desarrollo para asistentes . . . . .	24
2.3.1. Alexa Developer Console . . . . .	24
2.3.2. Google Actions Console . . . . .	25
2.3.3. Voxa . . . . .	26
2.4. Juegos y skills de asistentes . . . . .	27
2.4.1. Juegos de Alexa . . . . .	27
2.4.2. Juegos de Google Assistant . . . . .	28
2.5. Análisis de los juegos de aventura en asistentes virtuales . . . . .	31
2.5.1. Juegos de aventura en Alexa . . . . .	34
2.5.2. Juegos de aventura en Google Assistant . . . . .	36
2.6. Conclusiones . . . . .	37
<b>3. Análisis inicial del problema</b>	<b>39</b>
3.1. Concepto inicial del juego . . . . .	39
3.2. Narrativa . . . . .	40
3.3. Controles . . . . .	40
3.4. Desarrollo de un prototipo inicial del juego . . . . .	44
3.5. Evaluación del prototipo y propuesta del desarrollo . . . . .	46

<b>4. Tecnología a usar</b>	<b>47</b>
4.1. Conceptos clave . . . . .	47
4.2. Hardware. Dispositivos usados . . . . .	48
4.3. Alexa Developer Console . . . . .	50
4.4. Google Actions Console . . . . .	54
<b>5. Metodologías a usar en el proyecto</b>	<b>61</b>
5.1. Diseño centrado en el usuario (DCU) . . . . .	61
5.2. Desarrollo ágil . . . . .	63
5.3. Fases de desarrollo de un videojuego . . . . .	64
5.4. Aplicación de las metodologías al proyecto . . . . .	66
<b>6. Plan de entregas</b>	<b>68</b>
<b>7. Desarrollo</b>	
<b>Entregas e iteraciones</b>	<b>70</b>
7.1. Concepto del juego . . . . .	70
7.2. Guía del juego . . . . .	71
7.2.1. Habitaciones . . . . .	71
7.2.2. Eventos del juego . . . . .	72
7.3. Entregas . . . . .	78
7.3.1. Entrega 0 . . . . .	78
7.3.2. Entrega 1 . . . . .	92
7.3.3. Entrega 2 . . . . .	98
7.3.4. Entrega 3 . . . . .	102
7.3.5. Entrega 4 . . . . .	109
7.3.6. Entrega 5 . . . . .	112
7.4. Análisis de usabilidad . . . . .	114
7.4.1. Encuesta a usuarios que no han probado el juego . . . . .	114
7.4.2. Encuesta a usuarios que han probado el juego . . . . .	120
7.5. Publicación . . . . .	127
7.6. Conclusiones . . . . .	129
<b>8. Conclusiones y trabajo futuro</b>	<b>130</b>
8.1. Cumplimiento de los objetivos . . . . .	130
8.2. Comparativa de las plataformas . . . . .	131
8.3. Conclusiones . . . . .	133
8.4. Trabajos futuros . . . . .	134

# Índice de figuras

2.1. Asistente por voz. Altavoz inteligente <i>Amazon Echo</i> . . . . .	20
2.2. Asistente por texto de <i>Google</i> . . . . .	20
2.3. Asistente con imágenes de <i>Amazon</i> . . . . .	21
2.4. <i>Siri</i> integrada en un <i>smartphone</i> . . . . .	22
2.5. Dibujo de un altavoz con <i>Alexa</i> integrada. . . . .	22
2.6. <i>Google Assistant</i> . . . . .	23
2.7. <i>Cortana</i> . . . . .	23
2.8. <i>Bixby</i> . . . . .	24
2.9. <i>Alexa Developer Console</i> . . . . .	25
2.10. <i>Google Actions Console</i> . . . . .	26
2.11. <i>Voxa</i> . . . . .	27
2.12. Chiste de <i>Google Assistant</i> . . . . .	29
2.13. Adivinanza de <i>Google Assistant</i> . . . . .	29
2.14. Trivial de <i>Google Assistant</i> . . . . .	31
2.15. Ejemplo de aventura conversacional para ordenador. . . . .	32
2.16. <i>Escape Room</i> virtual. . . . .	32
2.17. Ejemplos de librojuegos. . . . .	33
2.18. Juego de rol. . . . .	34
2.19. Juego Amnesia. . . . .	35
2.20. Juego <i>Escape Room</i> . . . . .	36
2.21. Juego Rol en la tierra media. . . . .	36
2.22. Juego Choose your adventure. . . . .	37
2.23. Juego Santa Claus Escape. . . . .	37
3.1. Opciones del personaje al invocar el juego. Controles de la gestión de la partida. . . . .	41
3.2. Controles principales de exploración en cada habitación. . . . .	42
3.3. Controles que proporcionan información acerca del personaje en cualquier momento de la partida. . . . .	43
3.4. Juego inicial. . . . .	45
4.1. <i>Google Nest Mini</i> . . . . .	49
4.2. <i>Amazon Echo Show 5</i> . . . . .	49

4.3. Modelo de selección de <i>skill</i> durante el desarrollo en <i>Alexa Developer Console</i>	50
4.4. Selección del <i>endpoint</i> de la <i>skill</i> de <i>Alexa</i>	51
4.5. Selección del <i>template</i> para la <i>skill</i> de <i>Alexa</i>	51
4.6. Ejemplo de <i>intent</i> en <i>Alexa</i>	52
4.7. Ejemplo <i>Slot Type</i> en <i>Alexa</i>	52
4.8. Fragmento de código de <i>LookIntent</i>	53
4.9. Categorías de una <i>action</i>	54
4.10. Opciones de desarrollo para la <i>action</i>	54
4.11. Menú <i>Dialogflow</i>	55
4.12. Ejemplo de <i>intent</i> en <i>Google</i>	56
4.13. Ejemplo <i>Entity</i> en <i>Google</i>	56
4.14. Integración externa	57
4.15. <i>Fulfillment</i> de <i>Dialogflow</i>	57
4.16. Fragmento de código de <i>TellScoreIntent</i>	58
4.17. Integración con <i>Google Assistant</i>	59
4.18. <i>Actions</i> integradas en el juego	59
 5.1. Proceso del DCU. (Hassan-Montero, Y.; Ortega-Santamaría, S., 2009).	62
5.2. Ciclo del desarrollo ágil. (Junior, A., 2018)	63
5.3. Fases de producción de un videojuego. (Manrubia Pereira, A. M., 2014)	65
 7.1. Mapa de la casa	71
7.2. Prueba con el asistente de <i>Google</i>	78
7.3. Prototipo inicial en <i>Alexa</i>	79
7.4. Prototipo inicial en <i>Google</i>	79
7.5. <i>Slot</i> de tipo <i>Orientation</i>	80
7.6. <i>Slot</i> de tipo <i>Objects</i>	80
7.7. <i>Slot</i> de tipo <i>Orientation</i>	81
7.8. <i>Slot</i> de tipo <i>Choose</i>	82
7.9. <i>Look Intent</i>	82
7.10. <i>Interaction Intent</i>	83
7.11. <i>Default Interaction Intent</i>	83
7.12. <i>New Game Intent</i>	84
7.13. <i>Change Name Intent</i>	84
7.14. <i>Tell Name Intent</i>	84
7.15. <i>Take Intent</i>	85
7.16. <i>Release Intent</i>	85
7.17. <i>Inventory Intent</i>	86
7.18. <i>Use Intent</i>	86
7.19. <i>Where Intent</i>	87
7.20. <i>Tell Score Intent</i>	87

7.21. Choose Intent.	87
7.22. Clue Intent.	88
7.23. Archivos de código del juego inicial en Alexa.	88
7.24. Mapa de la casa en la entrega 1.	92
7.25. Slot de tipo Places.	93
7.26. Slot de tipo Symbol.	93
7.27. Slot de tipo Author.	94
7.28. Slot de tipo Cat Objects.	94
7.29. Go To Intent.	95
7.30. Read Intent.	95
7.31. Push Symbols Intent.	96
7.32. Number Code Intent.	96
7.33. Combine Objects Intent.	96
7.34. Use Object Intent.	97
7.35. Give Cats Intent.	97
7.36. Archivos de las habitaciones en Alexa.	98
7.37. Comida de gato.	99
7.38. Ovillo de lana.	99
7.39. Ratón de juguete.	99
7.40. Cuenco.	100
7.41. Diagrama de flujo al darle elementos al gato.	100
7.42. Diagrama de flujo al encontrarte con el gato fantasma.	101
7.43. Veneno.	101
7.44. Pantalla de introducción.	102
7.45. Pantalla de la habitación.	103
7.46. Pantalla del pasillo.	103
7.47. Pantalla de la sala de lavadoras.	104
7.48. Pantalla del baño.	104
7.49. Pantalla del patio.	105
7.50. Pantalla del salón.	105
7.51. Pantalla del sótano.	106
7.52. Pantalla de la cocina.	106
7.53. Pantalla de la habitación secreta.	107
7.54. Pantalla del recibidor.	107
7.55. Imágenes en cada dirección de la habitación.	108
7.56. Pantalla del inventario.	108
7.57. Pantalla del juego ganado.	109
7.58. Pantalla del juego perdido.	109
7.59. Continuar juego en Alexa.	110
7.60. Intent Continue Game.	110
7.61. Locked cat en Dialogflow.	113
7.62. Juego en inglés en Alexa.	113
7.64. Pregunta 1 de la encuesta 1 de usabilidad.	114

7.63. Encuesta 1 de usabilidad dirigida a los usuarios que no han probado el juego. . . . .	115
7.65. Pregunta 2 de la encuesta 1 de usabilidad. . . . .	116
7.66. Pregunta 3 de la encuesta 1 de usabilidad. . . . .	116
7.67. Pregunta 4 de la encuesta 1 de usabilidad. . . . .	117
7.68. Pregunta 5 de la encuesta 1 de usabilidad. . . . .	117
7.69. Pregunta 6 de la encuesta 1 de usabilidad. . . . .	118
7.70. Pregunta 7 de la encuesta 1 de usabilidad. . . . .	118
7.71. Pregunta 8 de la encuesta 1 de usabilidad. . . . .	118
7.72. Pregunta 9 de la encuesta 1 de usabilidad. . . . .	119
7.73. Pregunta 10 de la encuesta 1 de usabilidad. . . . .	119
7.74. Pregunta 11 de la encuesta 1 de usabilidad. . . . .	120
7.75. Pregunta 12 de la encuesta 1 de usabilidad. Sugerencias realizadas por los usuarios acerca de lo que añadirían al juego. . . . .	120
7.76. Encuesta 2 de usabilidad dirigida a los usuarios que han probado el juego. . . . .	121
7.77. Pregunta 1 de la encuesta 2 de usabilidad. . . . .	121
7.78. Pregunta 2 de la encuesta 2 de usabilidad. . . . .	122
7.79. Pregunta 3 de la encuesta 2 de usabilidad. . . . .	122
7.80. Pregunta 4 de la encuesta 2 de usabilidad. . . . .	123
7.81. Pregunta 5 de la encuesta 2 de usabilidad. . . . .	123
7.82. Pregunta 6 de la encuesta 2 de usabilidad. . . . .	123
7.83. Pregunta 7 de la encuesta 2 de usabilidad. . . . .	124
7.84. Pregunta 8 de la encuesta 2 de usabilidad. . . . .	124
7.85. Pregunta 9 de la encuesta 2 de usabilidad. . . . .	125
7.86. Pregunta 10 de la encuesta 2 de usabilidad. . . . .	125
7.87. Pregunta 11 de la encuesta 2 de usabilidad. . . . .	126
7.88. Pregunta 12 de la encuesta 2 de usabilidad. . . . .	126
7.89. Pregunta 13 de la encuesta 2 de usabilidad. . . . .	127
7.90. Vista previa del juego en <i>Google Assistant</i> . . . . .	128
7.91. Búsqueda de la <i>skill</i> publicada en <i>Amazon</i> . . . . .	128
7.92. Búsqueda de la <i>action</i> publicada para <i>Google Assistant</i> . . . . .	129
8.1. <i>Alexa Developer Console</i> . . . . .	131
8.2. <i>Dialogflow</i> . . . . .	132

# Glosario

**Action:** es una aplicación del asistente *Google Assistant..* 13

**Alexa Developer Console:** es una plataforma de desarrollo de aplicaciones para *Alexa..* 13

**Backend:** se trata de la lógica que se sitúa detrás de una aplicación. Principalmente puede ser el código que permite que dicha aplicación funcione de forma correcta.. 13

**Google Actions Console:** es una plataforma de desarrollo de aplicaciones para *Google Assistant..* 13

**Intent:** en un asistente de voz, representa una acción que se corresponde con una petición del usuario. Tiene un nombre y varias frases de prueba.. 13

**Interfaz de usuario:** permite la interacción entre el usuario y el dispositivo correspondiente. Puede ser una pantalla, una voz, un texto.... 13

**Nombre de invocación:** se trata del apelativo que se usa para que el asistente abra una aplicación concreta.. 13

**Skill:** es una aplicación del asistente *Alexa..* 13

**Slot o entity:** en un asistente de voz, son los argumentos de un *intent*. Representan los posibles valores de un elemento que forma parte de las frases de prueba.. 13

**Utterance:** en un asistente de voz, se trata de cada una de las frases de prueba de un *intent..* 13

**Wake word:** es la palabra que activa a un asistente de voz para que escuche al usuario.. 13

# Capítulo 1

## Introducción

Este proyecto profundiza en los asistentes virtuales con respecto al ámbito del entretenimiento. En la sociedad actual, este tipo de tecnología cobra cada vez más importancia y popularidad. Por ello, resulta interesante analizar cómo los usuarios interaccionan con los asistentes y sacan provecho de las funcionalidades que ofrecen.

Dado que el entretenimiento es uno de los objetivos principales de estos asistentes, analizar la oferta disponible en este sector es útil para examinar su impacto en los consumidores.

Una vez estudiados estos aspectos, podemos diseñar un juego para estos asistentes. La idea de este trabajo se basa en la elaboración de una aventura que combine elementos de distintas categorías (aventuras conversacionales, juegos de rol, puzzles...). El resultado nos permitirá obtener una experiencia que no solo entretenga al usuario, sino que lo involucre en una historia de forma activa, donde su modo de actuar y sus decisiones constituyan la base de la trama.

Por tanto, el estudio de las distintas plataformas para las que podemos desarrollar el juego, así como las características de las que dispondrá nuestra aventura, serán pasos fundamentales para crear un producto entretenido y fácil de usar. El diseño se centrará también en el usuario, de modo que creemos un juego que le interese, le haga pasar un rato divertido y le resulte sencillo de manejar dentro de las posibilidades de los asistentes virtuales.

### 1.1. Objetivo y subobjetivos

El objetivo principal de este proyecto consiste en diseñar y desarrollar un juego de tipo aventura para un asistente virtual. Este propósito puede

subdividirse en un conjunto de objetivos secundarios.

1. Hacer que esté disponible para varias plataformas: *Alexa* y *Google Assistant*.
2. Introducir elementos de *escape room*, aventuras conversacionales, juegos de rol y librojuegos.
3. Incorporar una funcionalidad dinámica.
4. Incluir imágenes y sonidos para crear una experiencia más inmersiva.
5. Disponer de la posibilidad de guardado para continuar la partida en otro momento.
6. Añadir varios idiomas para ampliar el alcance del juego.

El proceso seguido para desarrollar este juego, así como la consecución de las tareas en las que se subdivide, se van aclarando a lo largo de este documento. De este modo, quedan explicados con detalle los pasos seguidos a la hora de llevar este proyecto a la práctica.

## 1.2. Capítulos de la memoria

La memoria queda dividida en ocho capítulos. El capítulo que nos ocupa hace referencia a la introducción y los principales objetivos que se pretenden conseguir con la elaboración de este proyecto.

A continuación, el capítulo 2 aborda el estado del arte. Para ello, se habla de los antecedentes teóricos de los asistentes virtuales, así como de su funcionamiento en la actualidad. Además, se añaden ejemplos de asistentes actuales y algunos de sus juegos o funciones disponibles. También se abordan brevemente las plataformas utilizadas para el desarrollo de funcionalidades en distintos asistentes.

En el capítulo 3 se realiza el análisis inicial del problema. De esta forma, se explican tanto el concepto inicial del juego, como su narrativa y sus controles. A partir de ellos, se describe el prototipo inicial desarrollado.

El capítulo 4 profundiza en la tecnología utilizada. En primer lugar se especifican algunos conceptos clave a la hora de desarrollar aplicaciones para asistentes con interacción oral. Seguidamente, se hace referencia a los dispositivos usados durante el proyecto y se analizan las plataformas de desarrollo de *Alexa* y *Google Assistant*, los dos asistentes usados para implementar el juego.

En el capítulo 5 se tratan las metodologías empleadas durante la elaboración del proyecto. Se habla del desarrollo ágil, el diseño centrado en el usuario y las similitudes del proceso seguido para implementar la aventura con las fases de producción de un videojuego.

El capítulo 6 aborda el plan de entregas. Para ello, adjunta una tabla en la que se especifica cada una de las entregas y sus objetivos.

El capítulo 7 se centra en el desarrollo del juego. Se explica el concepto completo y se incluye una guía de la aventura. Además, se detalla cada una de las entregas especificadas en el capítulo 6, y se añade un análisis de usabilidad basado en las opiniones de los usuarios.

Por último, el capítulo 8 contiene las conclusiones del proyecto. Así, revisa el cumplimiento de los objetivos iniciales durante el desarrollo, hace una comparativa de las plataformas usadas y alude a un posible trabajo futuro.

# Capítulo 2

## Estado del arte

Este capítulo profundiza en los asistentes virtuales. Se alude a sus antecedentes teóricos, su integración y algunas de las plataformas usadas para el desarrollo de nuevas aplicaciones. Además, se considera el ámbito del entretenimiento, detallando las posibilidades que ofrecen *Alexa* y *Google Assistant*. Finalmente, se realiza un análisis de los juegos de tipo aventura desarrollados para asistentes.

### 2.1. Antecedentes teóricos

El origen del asistente personal inteligente proviene de un proyecto de inteligencia militar artificial. El objetivo consistía en desarrollar un asistente cognitivo que combinara varias tecnologías de inteligencia artificial. El nombre del proyecto fue CALO, que por sus siglas en inglés significa *Cognitive Assistant that Learns and Organizes*. A partir de este proyecto, se pudieron desarrollar productos como Siri (asistente de Apple iOS), Social Kinetics (una aplicación que aprende estrategias personalizadas para pacientes enfermos crónicos), el proyecto Trapit (hace selecciones inteligentes de contenido web basándose en las preferencias del usuario), TempoAI (un calendario inteligente)... (*Wikipedia, La enciclopedia libre*, 2020).

Con respecto al reconocimiento de voz, su origen se remonta a la década de 1950. Al principio, solo podían reconocerse números y dígitos, como en el caso de *Audrey*, inventado en 1952 por *Bell Laboratories*. Más adelante, en 1962, *IBM Shoebox* fue presentado en la Feria Mundial de Seattle. Era capaz de comprender diecisésis palabras y los números del 0 al 9. En la década de los 70, el departamento de Defensa de los Estados Unidos contribuyó enormemente al desarrollo de *Harpy* junto con la Universidad Carnegie

Mellon de Pittsburgh, Pensilvania. Este proyecto era capaz de comprender aproximadamente un millar de palabras.

El progreso continuó gracias al uso del modelo oculto de Markov. Dicho modelo consistía en el uso de estadísticas para establecer la probabilidad de que una palabra se correspondiera con un sonido desconocido. Incluso llegó a desarrollarse una muñeca llamada *Julie* en 1987, que podía responder a ciertas interacciones con los niños. A partir de los 90, la introducción de microprocesadores más rápidos hizo más factible el software de habla. El reconocimiento de voz se convirtió en una tecnología característica de algunas computadoras personales, y se desarrollaron softwares de reconocimiento de voz como *Dragon Dictate* y *Dragon NaturallySpeaking* de la compañía *Dragon*.

Finalmente, a partir de la década de los 2000, *Google* inventó la búsqueda por voz. Para ello, se usaron centros de datos que procesaban una enorme cantidad de información. De este modo, podían emparejarse las peticiones de los usuarios con ejemplos reales de habla humana. Actualmente, los asistentes de voz están en pleno auge, tanto de forma independiente como integrados en teléfonos móviles, ordenadores o plataformas digitales. Dichos asistentes pretenden facilitar la interacción con el usuario proporcionando un mayor dinamismo con un servicio intuitivo y sencillo (Kikel, C., 2016).

## 2.2. Asistentes virtuales

Un asistente virtual es un agente software que puede realizar tareas o servicios para un individuo basándose en comandos o peticiones.

El término *chatbot* suele usarse para referirse a asistentes a los que se accede mediante chat en línea, más enfocados al entretenimiento que a un propósito concreto. Otros asistentes usan el reconocimiento de voz y responden con voces sintetizadas a las peticiones de los usuarios. Estos dispositivos pueden utilizarse para automatizar tareas (encender y apagar luces, llamar por teléfono...), responder a preguntas, reproducir música, gestionar el email, añadir eventos a un calendario... En definitiva, es similar a un agente conversacional, salvo que tiene un propósito diferente. Mientras que el agente conversacional tiene como objetivo conversar con un humano, el asistente virtual se encarga de proporcionar utilidad mediante el cumplimiento de tareas (*Wikipedia, La enciclopedia libre*, 2020).

### 2.2.1. Tipos de asistentes

Podemos dividir los asistentes según el tipo de interacción con el que podemos utilizarlos.

- **Voz:** interacción oral. El asistente tiene una palabra clave o *wake word* con la que se activa. Después de decirlo, el usuario puede añadir algún comando de voz para obtener la funcionalidad deseada.



Figura 2.1: Asistente por voz. Altavoz inteligente *Amazon Echo*.

- **Texto:** interacción escrita. Una vez abierto el asistente, podemos conversar con él, pedirle información, entretenernos con algún juego... Los asistentes pueden combinar la voz y el texto. En algunos casos, como en dispositivos móviles, tenemos la posibilidad de visualizar el chat aunque interaccionemos con la voz. También existen altavoces inteligentes que poseen una pantalla en la que muestran lo que dice el asistente o nuestras peticiones.



Figura 2.2: Asistente por texto de *Google*.

- **Imágenes:** interacción con imágenes. Suele ser un añadido de los otros tipos de interacciones. Un ejemplo serían los altavoces inteligentes con una pantalla incorporada en la que aparecen texto e imágenes como apoyo visual. Además, la pantalla también puede ser táctil, añadiendo un nuevo tipo de interacción al asistente.



Figura 2.3: Asistente con imágenes de *Amazon*.

Algunos combinan varios tipos de interacción. Es el caso de *Google Assistant*, que en su integración en teléfonos móviles puede usarse como un chat en línea o de forma oral.

Los asistentes virtuales usan el procesamiento del lenguaje natural para emparejar la petición del usuario (ya sea por voz o por texto) con un comando ejecutable. Muchos de ellos aprenden continuamente gracias a técnicas de inteligencia artificial tales como el aprendizaje automático (*Wikipedia, La enciclopedia libre*, 2020).

### 2.2.2. Integración de los asistentes

Podemos encontrar asistentes virtuales en distintos dispositivos o plataformas. Algunos de ellos son:

- Altavoces inteligentes.
- Aplicaciones de mensajería instantánea.
- Sistema operativo de móviles u ordenadores.
- Teléfonos inteligentes con independencia del sistema operativo.
- Relojes inteligentes.
- Electrodomésticos.

### 2.2.3. Siri

Es el asistente de *Apple*. Puede usarse en dispositivos *iOS* como *iPhone*, *iPad*, *iPod touch*... Sus prestaciones varían en función del aparato. Información obtenida de <https://www.apple.com/es/siri/>.

Se activa diciendo *Oye, Siri* seguido de las instrucciones deseadas. Por ejemplo:

*Oye, Siri, pon una alarma a las 7 de la mañana.*

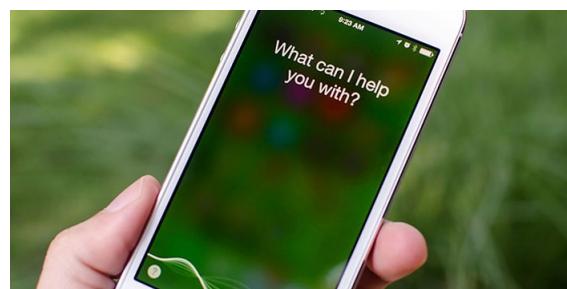


Figura 2.4: *Siri* integrada en un *smartphone*.

#### 2.2.4. Alexa

Es el servicio de voz ubicado en la nube de *Amazon*. Está disponible para los dispositivos de *Amazon* y aquellos que tienen *Alexa* integrada. El ejemplo más usado es el altavoz inteligente *Amazon Echo*. Información obtenida de <https://developer.amazon.com/es-ES/alexa>.

Para activarlo, hay que decir *Alexa* seguido del comando. Por ejemplo:

*Alexa, pon música.*

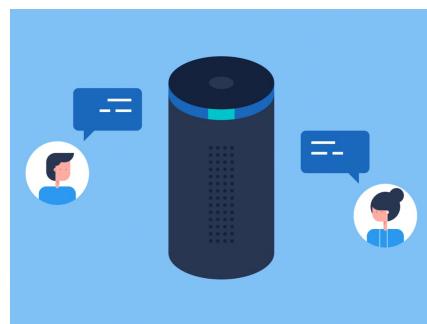


Figura 2.5: Dibujo de un altavoz con *Alexa* integrada.

### 2.2.5. Google Assistant

Es el asistente de *Google*. Está disponible en teléfonos móviles, en altavoces inteligentes como *Google Home* y *Google Nest*... En el caso de los móviles, combina la interacción por voz con la interacción por texto, ya que puede utilizarse de ambas maneras. Información obtenida de [https://assistant.google.com/intl/es\\_es/](https://assistant.google.com/intl/es_es/).

Se activa diciendo *Ok, Google* o *Hey, Google*:

*Ok Google, dime las noticias del día.*



Figura 2.6: *Google Assistant*.

### 2.2.6. Cortana

Es el asistente de *Microsoft*. Puede usarse desde un ordenador con *Windows*. Para iniciarla, puede buscarse en la barra de tareas o, si está activado el reconocimiento de voz, responde a *Hey, Cortana*. A continuación, escribimos o decimos lo que deseamos. Información obtenida de <https://www.microsoft.com/es-es/>.

*Hey, Cortana, ¿qué tiempo hará hoy?*



Figura 2.7: *Cortana*.

### 2.2.7. Bixby

Es el asistente virtual de *Samsung*. También combina voz y texto. Se puede activar diciendo *Hi, Bixby*. Información obtenida de <https://www.samsung.com/es/apps/bixby/>.

*Hi, Bixby, haz una foto.*



Figura 2.8: *Bixby*.

## 2.3. Desarrollo para asistentes

A la hora de crear nuevas aplicaciones o funcionalidades para los asistentes, existen distintas plataformas. Veamos cuáles pueden usarse en el caso de *Alexa* y *Google Assistant*.

### 2.3.1. Alexa Developer Console

Es una plataforma desde la que se pueden desarrollar *skills* para *Alexa*. Se puede elegir un modelo preconstruido o diseñar nuestra propia experiencia. El modelo preconstruido puede ser útil si nuestra *skill* tiene un propósito muy claro. Esto es, por ejemplo, si se trata de una aplicación para reproducir un vídeo o una funcionalidad de *smart home*, como encender o apagar una luz. Además, podemos alojar el *backend* dentro de *Alexa* o utilizar nuestros propios recursos.

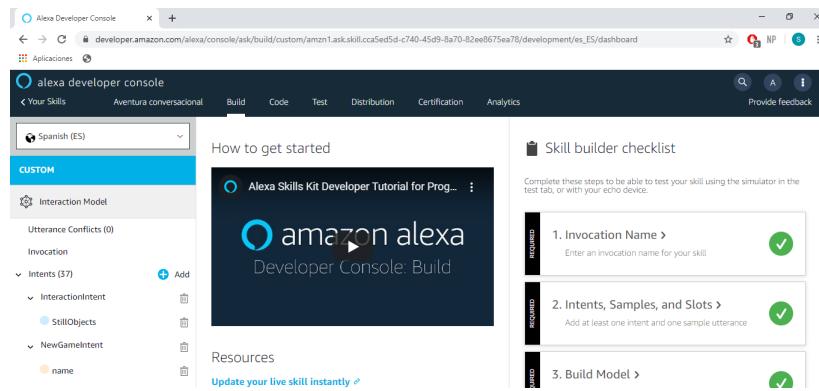


Figura 2.9: *Alexa Developer Console*

Una vez hemos escogido entre estas opciones, hemos de darle un nombre de invocación a nuestra *skill*. De este modo, cuando queramos abrirla, diremos:

*Alexa, abre nombre de invocación.*

A continuación, podemos añadir *intents*. Esto se relaciona con lo que podemos decirle a nuestra *skill* para interaccionar con ella. Imaginemos que queremos desarrollar una aplicación para conocer el tiempo metereológico. La pregunta principal sería qué tiempo hace. Podríamos crear un *intent* para preguntar por el tiempo. Las distintas frases con las que podríamos hacer dicha pregunta son las *utterances*. De este modo, tendríamos el *intent* al que podríamos llamar *TiempoIntent*, que a su vez tendría muchas frases de ejemplo o *utterances* para ser invocado, como: ¿qué tiempo hace?, dime el tiempo que hace, ¿cuál es el tiempo?...

Si ampliaríamos nuestra *skill* para que también nos dijera qué día es hoy, podríamos crear un nuevo *intent* llamado *DíaIntent*. Las *utterances* podrían ser: ¿qué día es hoy?, ¿cuál es el día?, dime qué día es...

Por último, habría que construir la interacción. Para ello, tendríamos que configurar las respuestas a los distintos *intents* desde el *backend*.

### 2.3.2. Google Actions Console

Desde esta plataforma se pueden desarrollar *Google Actions*. El desarrollo es similar al de una *skill* de *Alexa*. Una vez creado un nuevo proyecto, le damos un nombre y elegimos el idioma y la región.

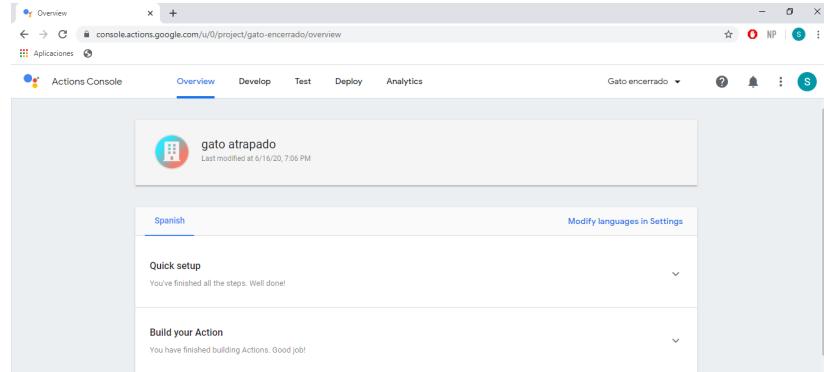


Figura 2.10: *Google Actions Console*.

Al igual que con las *skills*, podemos elegir un modelo preconstruido en función del objetivo de nuestra aplicación: juegos, comida, *smart home*, negocios... En el desarrollo, tenemos diversas opciones. Por un lado, podemos usar *templates*, usar la terminal, crear nuestra propia experiencia conversacional personalizada...

Una de las posibilidades sería integrar la plataforma *DialogFlow* en nuestro proyecto. De este modo, escribiríamos los distintos *intents* con sus respectivas *utterances* y, a la hora de escribir el código para las interacciones del asistente, podemos usar un editor en línea o proporcionar nuestro propio servidor web. Esto constituirá el *backend* de nuestra *Action*.

También es necesario proporcionar un nombre de invocación. Así, nuestra *action* podría ser llamada de la siguiente manera:

*Ok, Google, hablar con nombre de invocación.*

### 2.3.3. Voxa

Es un marco de trabajo que provee una manera de organizar una experiencia conversacional convirtiéndola en una máquina de estados. Aporta la flexibilidad necesaria en cada momento.

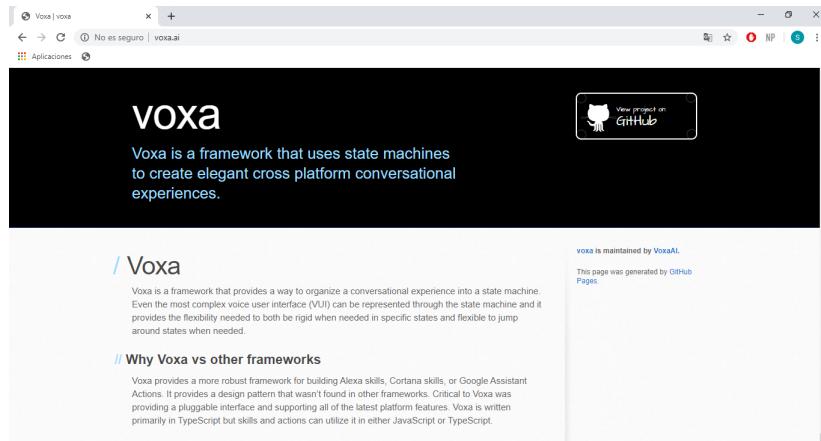


Figura 2.11: Voxa.

Permite desarrollar aplicaciones que puedan integrarse para varios asistentes virtuales a la vez. Se trata de un marco de trabajo robusto para construir *Alexa skills*, *Cortana skills* o *Google Assistant Actions*. Soporta las últimas características de las plataformas. Está escrito principalmente en *TypeScript*, pero las *skills* y *actions* pueden usar tanto en *JavaScript* como *TypeScript*.

## 2.4. Juegos y skills de asistentes

Una de las funciones de los asistentes virtuales es la de proporcionar entretenimiento. Para ello, suelen ser habituales los juegos. Cada asistente posee sus propios juegos, pero también podemos añadir más. En el caso de *Alexa*, por ejemplo, podemos incluir nuevas *skills* a nuestros dispositivos desde *Amazon*.

Algo común a casi todos los asistentes son los chistes, las adivinanzas, las preguntas o las canciones. En lo relativo a los juegos, podemos encontrar muchas categorías: de preguntas y respuestas, de adivinar, de simulación...

A continuación, vamos a ver algunos ejemplos para los asistentes de *Google* y *Alexa*.

### 2.4.1. Juegos de Alexa

*Alexa* tiene mucha variedad de juegos. Algunos de ellos son:

- **Chiste del día:** cada día cuenta un chiste diferente.
  - **Pregunta:** “¿Cuál es el chiste del día?”.
  - **Respuesta:** “Hoy te traigo un chiste de lo más romántico. Una pareja está hablando de su relación y finalmente uno le pregunta al otro: *Entonces, ¿tú y yo qué somos?* A lo que su pareja le responde: *¡Pronombres, cariño!*”
- **Pregunta del día:** ofrece una pregunta sobre una categoría concreta. Si es respondida de forma correcta, aumenta nuestra puntuación y accedemos a una pregunta bonus.
- **Frases curiosas:** el asistente nos proporciona respuestas divertidas para muchas frases. Algunos ejemplos son:
  1. • **Pregunta:** “Alexa, hazme un sándwich”.  
• **Respuesta:** “Está bien. ¡Eres un sándwich!”
  2. • **Pregunta:** “¿Conoces a Siri?”  
• **Respuesta:** “A veces nos encontramos en la nube. Tenemos una buena relación profesional”.
  3. • **Pregunta:** “¿Hablas código Morse?”  
• **Respuesta:** “Da-dit, da-da, dit, dit, dit. Eso significa que sí.”
  4. • **Pregunta:** “¿Tienes mascotas?”  
• **Respuesta:** “Me gustaría tener un perro robot como K9, de Doctor Who, para que me acompañase en todas mis aventuras por el tiempo-espacio. ¡Sería genial!”.
  5. • **Pregunta:** “¿Me quieres?”  
• **Respuesta:** “El amor es un misterio para mi algoritmo. Demasiadas variables, demasiados caminos”.

#### 2.4.2. Juegos de Google Assistant

*Google* también ofrece distintas opciones con respecto a los juegos. Vamos a ver las respuestas que ofrece *Google Assistant* en el teléfono móvil a estas opciones:

- Chiste



Figura 2.12: Chiste de *Google Assistant*.

- Adivinanza

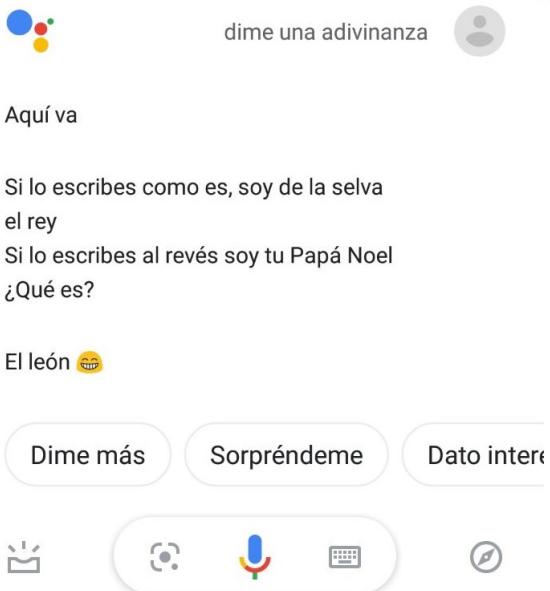


Figura 2.13: Adivinanza de *Google Assistant*.

- **Preguntas divertidas:** existen muchas frases que podemos decirle al asistente para que nos responda de forma divertida. Por ejemplo:

1. • **Pregunta:** “Estoy aburrido”.
    - **Respuesta:** “¡El aburrimiento no puede con nosotros! Podemos jugar a algunos juegos, puedo intentar hacerte reír o sorprenderte con algo de diversión”.
  2. • **Pregunta:** “¿Tienes sentimientos?”
    - **Respuesta:** “Déjame ver si puedo enojarme. (Rugidos.) Oh, eso fue inesperado”.
  3. • **Pregunta:** “¿Cuál es tu misión?”
    - **Respuesta:** “Mi misión es matar a las bestias de la ignorancia y buscar la información más fascinante.”
  4. • **Pregunta:** “¿Crees en Santa Claus?”
    - **Respuesta:** “Por supuesto que Santa es real. Incluso tengo un rastreador que puede decirme dónde está en este momento”.
  5. • **Pregunta:** “¿Cuál es el sentido de la vida?”
    - **Respuesta:** “Vengo con garantía de fábrica, por lo que no me preocupo por esas cosas”.
- **Trivia de la suerte:** es un juego de preguntas y respuestas. También se llama *Yo me la sé*. Se puede jugar solo o con más personas. Consiste en 5 preguntas que debes responder de forma correcta para ir acumulando puntos. Este es un ejemplo de una de las preguntas durante el desarrollo del juego:

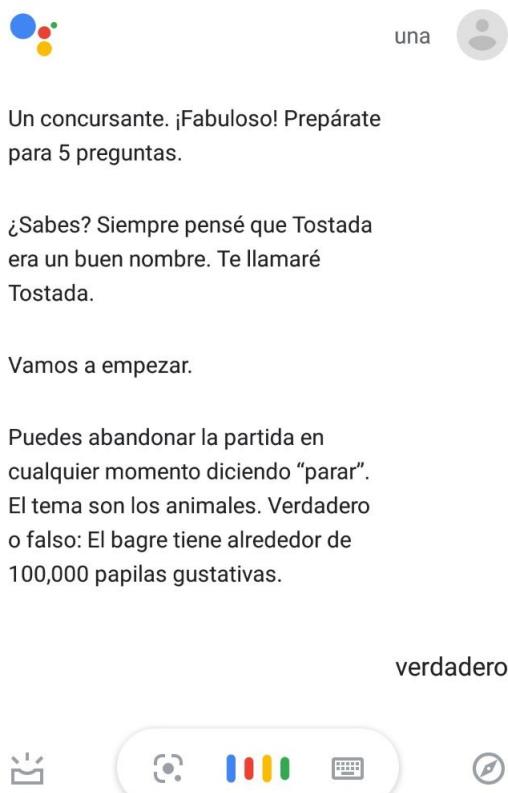


Figura 2.14: Trivial de *Google Assistant*.

## 2.5. Análisis de los juegos de aventura en asistentes virtuales

En el ámbito de los juegos de aventura, existen muchos tipos con dinámicas muy interesantes. Los asistentes virtuales tratan de adaptar dichos géneros dentro de sus posibilidades, generando aplicaciones interactivas cuyo fin es el entretenimiento del usuario. Principalmente, distinguimos cuatro categorías de juegos en función de sus características:

- **Aventuras conversacionales:** se trata de un género de videojuegos que se basa en la descripción de una situación a partir de un texto. Su uso ya no está muy extendido. La dinámica consiste en que el jugador teclea la acción que desea realizar en cada momento. El juego interpreta la entrada y se produce una nueva situación en función de la opción elegida. Además, puede incluir algunas imágenes como apoyo

### 32 2.5. Análisis de los juegos de aventura en asistentes virtuales

visual, similares a las ilustraciones de los libros.



Figura 2.15: Ejemplo de aventura conversacional para ordenador.

En el caso de los asistentes virtuales, las aventuras conversacionales suelen basarse en la interacción oral. Así, se sustituyen los textos escritos por narraciones y las acciones son elegidas mediante comandos de voz. También pueden tener soporte visual e incluir sonidos que consigan crear una aventura más realista y envolvente.

- **Escape room:** se trata de juegos en los que se encierra a un grupo de personas en una habitación y deben conseguir escapar antes de un tiempo determinado. Para ello, deben interactuar con los objetos que los rodean y usarlos de forma adecuada. Su característica principal es la resolución de puzzles y retos para lograr el objetivo.



Figura 2.16: *Escape Room* virtual.

Al igual que con las aventuras conversacionales, lo más habitual es

desarrollar este tipo de juegos mediante la interacción oral en los asistentes virtuales. De este modo, también se procede a ejecutar las distintas posibilidades mediante comandos de voz. Por lo demás, la resolución de los rompecabezas es igual que en las *escape room* convencionales.

- **Librojuegos:** son libros en los que se involucra al lector de forma directa para que tenga poder de decisión sobre la historia. Esto se lleva a cabo dejando que elija entre varias acciones posibles, llevándolo a una página u otra del libro en función de la que escoja. De esta manera, los acontecimientos cambian según las elecciones tomadas.

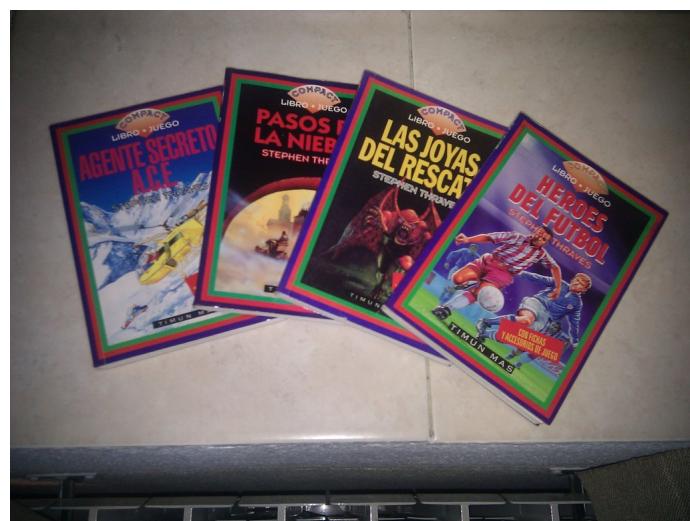


Figura 2.17: Ejemplos de librojuegos.

Los asistentes virtuales pueden adaptar esta posibilidad a partir de un árbol de decisiones. Esto quiere decir que las elecciones del jugador enfocarán su aventura a un camino u otro, tal y como sucede en los libros.

- **Juegos de rol:** los jugadores asumen el papel de personajes imaginarios durante el transcurso de una historia. Se trata de un juego con muchas posibilidades, dado que no hay un guion preestablecido. Así, la imaginación y la narrativa son los puntos fundamentales de este tipo de aventuras. Además, suele haber una persona encargada de dirigir el juego, de tal forma que guía a los personajes por la historia basándose en las situaciones que se plantean y las decisiones tomadas.



Figura 2.18: Juego de rol.

En este caso, un juego de rol enfocado para un asistente virtual puede dar mucho protagonismo a la narrativa y permitir que el jugador desarrolle su personaje mediante las elecciones que haga durante el juego. Además, el jugador puede tener ciertas características que vayan mejorando durante el transcurso de la historia, como su habilidad de ataque o defensa frente a diversos peligros.

### 2.5.1. Juegos de aventura en Alexa

Algunas *skills* de *Alexa* que engloban ciertas características de las categorías anteriores son:

- **Amnesia:** el jugador despierta en un bosque y debe ir resolviendo una serie de puzzles para conseguir escapar de la situación. Combina los rompecabezas de las *escape room*, la narrativa de los juegos de rol y la elección de decisiones de las aventuras conversacionales. Los controles básicos son:

1. Inspecciona objeto. Ej. Inspecciona el armario.
2. Ve a lugar. Ej. Ve hacia el bosque.
3. Usa objeto. Ej. Usa linterna.
4. Combina objeto 1 y objeto 2. Ej. Combina linterna y pilas.
5. Soltar/Coger objeto. Ej. Tira las pilas.
6. Retroceder. Ej. Vuelve atrás.
7. ¿Dónde estoy?

8. ¿Qué tengo en la mochila?
9. Reiniciar partida.
10. ¿Qué puedo hacer?
11. ¿Qué hay en el suelo?



Figura 2.19: Juego Amnesia.

- **Escape Room:** este juego tiene varios escenarios ordenados por nivel de dificultad.

- Celda de Cárcel - Más fácil
- Oficina – Dificultad media
- Coche – Más Difícil
- Garaje - Bonus

El objetivo es salir de la habitación que corresponda a través de la interacción con los objetos que hay en ella. Es básicamente una *escape room* virtual con la toma de decisiones propia de las aventuras conversacionales. Los controles básicos son:

1. Mirar {Dirección}.
2. Inspeccionar/Mirar/Usar {Objeto}.
3. Usar {Artículo} en {Objeto}.
4. Dame una pista.
5. Repite pistas.
6. Inventario: dice lo que lleva encima el jugador.
7. Ayuda: recuerda las acciones posibles en cada momento.

Además, este juego ofrece una guía completa de cada una de las habitaciones en el caso de que el usuario no sepa cómo continuar. Y, en el caso de dispositivos con pantalla, como *Amazon Echo Show*, despliega imágenes relacionadas con la aventura, haciéndola más dinámica e interactiva.

### 36 2.5. Análisis de los juegos de aventura en asistentes virtuales



Figura 2.20: Juego *Escape Room*.

- **Rol en la tierra media:** se trata de un juego de rol basado principalmente en la parte narrativa. También interviene la toma de decisiones similar a la de los *librojuegos*, ya que en función de si se realizan ciertas acciones o no, la aventura modifica su trama.

El objetivo es tener éxito en la aventura. El juego está ambientado en la saga de libros de *El Señor de los Anillos*. Los controles son, básicamente, decir sí o no a las acciones que se van proponiendo.



Figura 2.21: Juego Rol en la tierra media.

#### 2.5.2. Juegos de aventura en Google Assistant

En el caso de *Google*, estas son algunas *actions* de aventura:

- **Choose your Adventure:** es un juego basado en los *librojuegos*. Narra una aventura en la que se deben ir tomando decisiones entre varias opciones propuestas. Por tanto, la historia cambia según lo que elija el usuario.

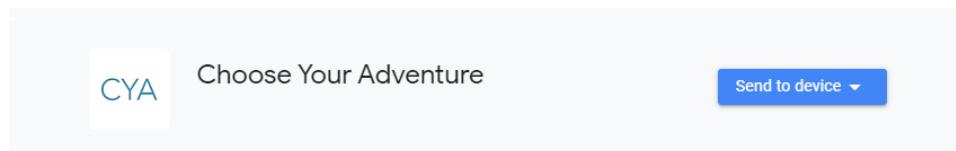


Figura 2.22: Juego Choose your adventure.

- **Santa Claus Escape:** es un juego en el que hay que liberar a Santa Claus, que ha sido atrapado por una hada. Para ello, el jugador debe resolver varios puzzles y responder unas preguntas. Se trata principalmente de retos propios de las *escape room*.

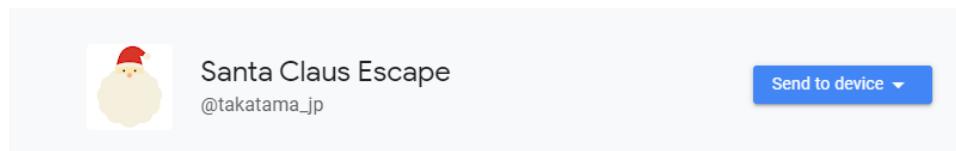


Figura 2.23: Juego Santa Claus Escape.

## 2.6. Conclusiones

Los asistentes virtuales están ganando cada vez más popularidad en la actualidad. La mejora del reconocimiento de voz junto con la posibilidad de controlar ciertas funcionalidades del hogar con un simple comando los convierte en dispositivos útiles que hacen nuestro día a día más cómodo. Además, proporcionan entretenimiento y compañía, creando un ambiente agradable a través de una interacción dinámica y sencilla.

En el ámbito del control de las funciones del hogar o *smart home*, podemos encender y apagar bombillas, activar la televisión, controlar el termo-

tato, poner música... Muchas de estas funcionalidades requieren dispositivos inteligentes que puedan conectarse al asistente, con lo que cada vez se amplían más las opciones de las que disponemos para automatizar nuestra vivienda.

Con respecto al entretenimiento, cada dispositivo posee una amplia variedad de juegos e incluso podemos mantener una conversación con nuestro asistente. Algunas de sus respuestas son bastante divertidas, con lo que se convierte en una interacción amena.

Si nos centramos en los juegos de aventura, la adaptación de este tipo de entretenimiento a los asistentes de voz resulta bastante interesante. Es una manera de dinamizar nuestra intervención. Además, los sonidos pueden proporcionar una experiencia más envolvente e incluso podemos visualizar algunas imágenes si tenemos un dispositivo con pantalla.

Puede resultar original y divertido entremezclar las distintas categorías de aventuras incorporando características variadas de cada uno de los tipos. Así, se podrían aprovechar la evolución del personaje y la narrativa de los juegos de rol, la toma de decisiones de las aventuras conversacionales, la resolución de rompecabezas de las *escape rooms* y la ramificación de posibles desarrollos de la historia de los librojuegos. Esto daría lugar a una experiencia muy atractiva, tal y como hemos visto en los ejemplos de *skills* de *Alexa* y *actions* de *Google* del apartado anterior.

# Capítulo 3

## Análisis inicial del problema

En esta sección, se expone una primera propuesta para el desarrollo del juego. A partir del concepto inicial, se desarrolla y evalúa un prototipo funcional de la aventura. Dicho prototipo constituirá la base sobre la que se irán construyendo las sucesivas ampliaciones y mejoras.

### 3.1. Concepto inicial del juego

La idea inicial consiste en el desarrollo de un juego que combinará distintos aspectos de las categorías de aventura mencionadas en el capítulo anterior. La temática se fundamentará en escapar de un lugar mediante la resolución de puzzles y la interacción con los objetos que el usuario vaya viendo y encontrando. Con esa premisa, combinamos las características de las aventuras conversacionales y de las *escape room*. Con respecto a los juegos de rol, crearemos nuestro propio personaje, que tendrá un nombre y una puntuación asociada en función de las decisiones que tome. Además, la narrativa será un punto importante, incorporando sonidos a la historia para crear un ambiente que nos sumerja en ella. Por último, tendremos algunas situaciones en las que podremos elegir entre varias opciones, tal y como sucede en los librojuegos. De esta forma, se verán afectados el destino y la puntuación del personaje según las alternativas elegidas por el usuario.

La forma de interacción se basará en mirar en todas las direcciones de la estancia en la que se encuentre el personaje. Esto le llevará a descubrir objetos que puede examinar, almacenar o utilizar para seguir avanzando y lograr escapar.

### 3.2. Narrativa

La historia comienza con el jugador despertándose en lo que parece ser su habitación. Empieza a explorarla y descubre que las cosas no le cuadran. Además, está atrapado, con lo que su principal objetivo se convierte en escapar cuanto antes. Para ello, debe interactuar con los objetos que va encontrando y resolver retos que le permitan avanzar por la casa. Por si fuera poco, van apareciendo elementos relacionados con los gatos en las habitaciones. Sin embargo, el personaje nunca ha tenido una mascota.

Conforme la aventura avanza, el jugador puede tomar ciertas decisiones que influirán de forma decisiva en su destino. Hay una trama principal, más enfocada a la resolución de retos, y una trama alternativa, más breve, centrada en la narrativa y con un final algo perturbador. Esto resulta interesante al introducir un escenario similar al de ciertas tramas de las *escape room*, donde suele predominar la intriga. Además, el jugador va ganando o perdiendo puntos y puede llegar a morir si toma las decisiones equivocadas.

### 3.3. Controles

Para poder interactuar con el asistente a la hora de jugar, se desarrollarán unos controles que permitan al usuario elegir la acción deseada de un modo sencillo. Se pretende dejar cierta libertad de decisión al jugador dentro de las posibles acciones que puede realizar. De este modo, en cada momento del juego podrá elegir entre varias opciones, evitando restringir su comportamiento a una sola posibilidad. Los controles serán concretos y breves, incluyendo algunos de ellos varias formas de ser invocados para facilitar la comunicación.

Por una parte, estarán los controles asociados a la gestión de la partida. Una vez que el juego haya sido invocado, podremos iniciar una nueva partida o continuar una anterior. Esta posibilidad de guardar los datos puede ser muy interesante para que el usuario pueda controlar el ritmo del juego. De este modo, podrá dedicar el tiempo que desee a jugar sin miedo a perder los avances conseguidos. También será útil pedir ayuda para conocer el manejo del personaje durante la aventura, especialmente si es la primera vez que se accede al juego y no se conoce el modo de interacción.

- **Nuevo juego o reiniciar:** comienza el juego desde el inicio.
- **Ayuda:** explica los controles. Puede seleccionarse entre una ayuda larga y una corta.

- **Continuar juego:** permite continuar una partida anterior siempre que no hubiera finalizado.

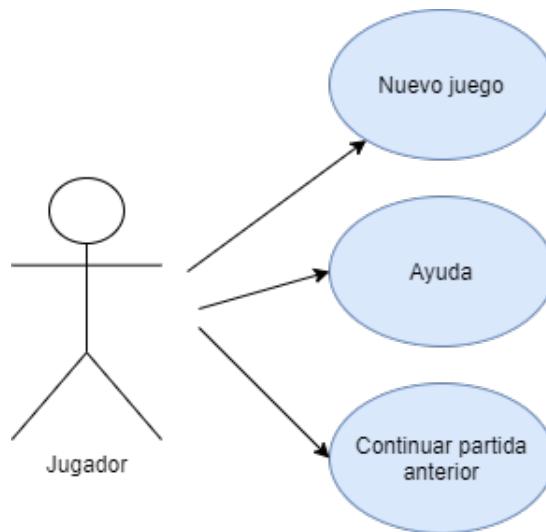


Figura 3.1: Opciones del personaje al invocar el juego. Controles de la gestión de la partida.

En segundo lugar, distinguiremos los comandos asociados a la exploración de una habitación y el uso de los objetos. La idea del juego consiste en que el usuario mire en todas las direcciones, descubriendo distintos elementos, y examine cada uno de ellos. Esto le permitirá encontrar objetos que podrá utilizar para resolver los puzzles y poder seguir avanzando. Además, dispondrá de un inventario en el que poder almacenar elementos que puedan resultarle útiles más adelante. También es posible tomar decisiones incorrectas que resten puntos o, incluso, provoquen la muerte del jugador.

- **Mirar en cualquier dirección:** este, oeste, norte, sur, arriba y abajo.
- **Examinar un elemento de la habitación:** permite obtener información sobre dicho elemento y saber si oculta algo.
- **Coger o soltar un objeto:** guarda un objeto en el inventario o lo suelta en el suelo de la habitación en la que se encuentre el jugador.
- **Usar un objeto:** usa un objeto del inventario. Se trata de objetos que pueden usarse por sí solos. Por ejemplo: una linterna. También se aplica a objetos que contengan cierta información de interés. Es decir, en el caso de una carta, usarla implicaría leer su contenido.

- **Usar un objeto en otro objeto:** se refiere a utilizar un objeto del inventario para interactuar con un elemento de la habitación. Un ejemplo podría ser usar una llave para abrir una puerta.
- **Combinar dos objetos del inventario:** permite unir dos objetos que no se pueden usar por sí solos para crear uno que resulte útil. Por ejemplo: combinar las pilas y la linterna para crear una linterna que pueda usarse. **Ir de una habitación a otra:** permite desplazarse por la casa siempre que la habitación a la que se quiera acceder sea contigua a la habitación en la que está el personaje y nada bloquee el camino.



Figura 3.2: Controles principales de exploración en cada habitación.

En tercer lugar, habrá ciertos controles independientes de la interacción con los objetos o las habitaciones. Podrán usarse de forma recurrente y resultarán útiles para que el jugador sepa en todo momento su localización, qué lleva encima, su puntuación o su nombre. También puede pedir una pista en el caso de que no sepa cómo seguir avanzando o no logre resolver algún puzzle.

- **Obtener una pista:** da una pequeña ayuda correspondiente a la habitación en la que se encuentre el jugador.
- **Inventario:** dice lo que el personaje lleva en el inventario.
- **Dónde estoy:** informa de la habitación en la que se encuentra el jugador.

- **Cómo me llamo:** dice el nombre del personaje.
- **Cambiar nombre:** permite modificar el nombre del personaje.
- **Cuál es mi puntuación:** informa de la puntuación que tiene el personaje hasta el momento.

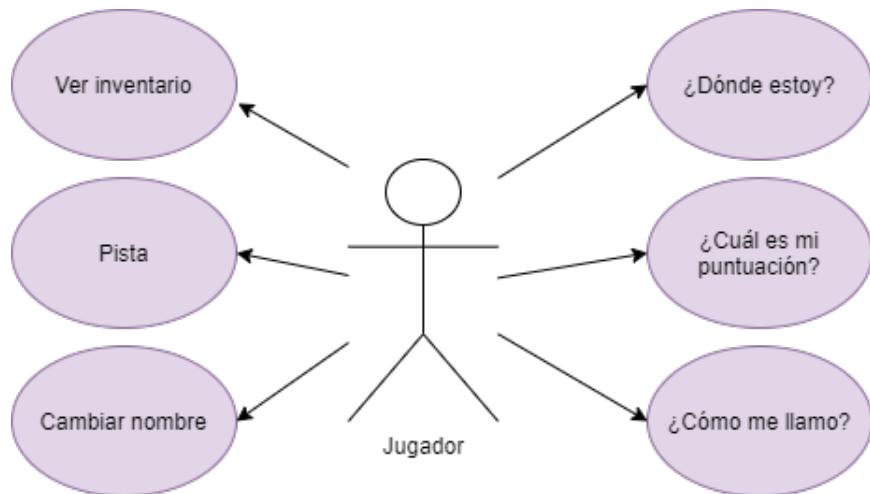


Figura 3.3: Controles que proporcionan información acerca del personaje en cualquier momento de la partida.

Por último, habrá varios controles que se usarán de forma ocasional. El juego informará al usuario de cuándo son necesarios, dado que solo serán útiles para realizar ciertas acciones en momentos clave. Estos controles se relacionan, por ejemplo, con introducir una combinación en un candado o pulsar algunos símbolos en cierto orden para abrir una puerta.

- **Introducir un número:** permite probar una combinación en un candado.
- **Pulsar símbolos:** deja seleccionar el orden en el que se quieren pulsar los símbolos de una de las puertas para conseguir abrirla.
- **Dar elementos al gato:** en un momento del juego, permite darle ciertos elementos a un gato en un orden. Según el número de objetos que se le den y el orden, el jugador puede tener éxito o no.
- **Elegir una acción:** habrá ciertos objetos en el juego sobre los que se plantearán varias opciones. Por ejemplo, cuando el personaje encuentre comida de gato, puede decidir guardarla o comérsela. Su acción determinará si pierde puntos o no, y puede afectarle en el futuro.

Cuando el juego termine, se informará al jugador de la puntuación obtenida y de si ha ganado o perdido. Dado que el principal objetivo es escapar, lograrlo será la única forma de ganar. Además, las decisiones tomadas durante la partida y en el momento final determinarán la puntuación. Si el usuario muere o se desmaya durante el transcurso del juego, perderá automáticamente.

### **3.4. Desarrollo de un prototipo inicial del juego**

Una vez establecido el concepto del juego, se ha desarrollado una primera versión de la aventura con una sola habitación. De este modo, el objetivo se convierte en escapar de dicha habitación.

El nombre de invocación del juego es *juego inicial*. Está disponible tanto para *Alexa* como para *Google Assistant* en español. En el caso de *Alexa*, también puede utilizarse en inglés, invocándolo por *initial game*. Una vez iniciado, te da la bienvenida. El jugador puede elegir nuevo juego, pedir ayuda para conocer los controles o preguntar dónde está, qué tiene en el inventario, cuántos puntos tiene o cómo se llama para saber si hay alguna partida anterior a medias. Aunque en este prototipo inicial no está incluida la opción de continuar el juego, si la partida anterior se cerró hace poco, los datos aún permanecen en la memoria y puede seguirse la aventura por donde se dejó. Sin embargo, esto solo es posible a corto plazo, con lo que la incorporación de la posibilidad de continuar el juego es una opción bastante necesaria con respecto al desarrollo de un prototipo más complejo de la aventura.

Dado que se trata de una versión inicial de la aventura, hay menos comandos que los vistos en el apartado anterior. Si el usuario decide dar comienzo a la aventura, despierta en una habitación. Una vez ahí, empieza a encontrar elementos y pruebas diferentes. Estos son los sucesos en orden:

- **Habitación:** al comenzar el juego, apenas hay luz en el cuarto. Únicamente se ve un interruptor en la pared. De este modo, lo primero que se debe hacer es *mirar interruptor*. Esa acción hará que se pulse el botón, con lo que se obtendrá la luz necesaria para examinar la habitación y 10 puntos. Esto es lo que se encuentra en cada dirección:
  - **Norte:** el interruptor que se ha pulsado y la puerta. Se necesita la llave para abrir la puerta. Una vez que se abre, se consiguen 10 puntos y el juego termina, diciendo la puntuación final.
  - **Este:** hay un cuadro y un armario. Cuando se interacciona con el cuadro, se cae una llave al suelo que puede cogerse. Si se coge, se

guarda en el inventario y se obtienen 10 puntos. En el armario hay una trampilla que está atascada. Si se rompe con el martillo se consiguen otros 10 puntos y se encuentra comida de gato. Una vez que se ha encontrado, se plantea la opción de comerla o guardarla. Si se elige lo primero, se pierden veinte puntos. Si se selecciona lo segundo, se obtienen veinte puntos.

- **Oeste:** están la cama y la almohada. En la almohada hay un martillo que se puede coger.
- **Sur:** hay un escritorio con un cajón y una ventana. El cajón está vacío. Se puede romper la ventana con el martillo diciendo *usar martillo en ventana*. En este caso, se pierden 20 puntos y el jugador muere al caerse por la ventana, con lo que pierde el juego.
- **Arriba:** está la lámpara que se ha encendido. Si se usa la silla, el jugador puede asomarse y ve que no hay nada dentro. Sin embargo, al hacerlo obtiene diez puntos.
- **Abajo:** está el suelo. Hay una silla que puede cogerse.



Figura 3.4: Juego inicial.

Además de esto, si se pide una pista se pierden diez puntos. El personaje también puede preguntar dónde está, cuántos puntos lleva, cómo se llama, cambiar su nombre o saber lo que lleva en el inventario.

**3.5. Evaluación del prototipo y propuesta del desarrollo**

Esta versión reducida del juego incluye los aspectos básicos de la aventura y desarrolla el concepto inicial. Permite al jugador explorar la habitación, tomar algunas decisiones que le afectarán de forma decisiva y seguir una pequeña narrativa que incorpora el objetivo principal de escapar. A pesar de ser una experiencia corta, el usuario puede ganar o perder, y la puntuación depende de las interacciones realizadas. Todo este concepto refleja muy bien la idea e intencionalidad del juego.

Una nueva propuesta incluiría una versión más compleja del juego, donde se incluyeran todos los controles explicados y se desarrollara la narrativa por completo. Para ello, se ampliarían el número de habitaciones, los retos y las decisiones que el jugador puede tomar. Además, se trataría de una versión mejorada, donde existiría la posibilidad de continuar la partida en otro momento en el caso de que el jugador la dejara a medias. Por último, sería una aventura más dinámica y que permitiría una experiencia más inmersiva, incorporando diferentes sonidos e incluso imágenes. Esto es, precisamente, lo que se ha añadido en entregas posteriores.

# Capítulo 4

## Tecnología a usar

Durante el desarrollo de la aventura, se han utilizado diferentes tecnologías. Dado que uno de los objetivos era que estuviera disponible para varias plataformas, un paso importante ha sido familiarizarse con la implementación desde *Alexa Developer Console* y *Google Actions Console*. Antes de explicar cada plataforma de forma individual, es necesario aclarar unos conceptos relativos a los asistentes virtuales. Aunque vienen incluidos en el glosario, en esta sección se explican con mayor detenimiento.

### 4.1. Conceptos clave

En el ámbito de los asistentes de voz, existe una serie de términos muy importantes a la hora de desarrollar una aplicación.

- **Intent:** representa una acción que responde a una petición del usuario. Cada *intent* tiene un nombre y varias *utterances*.

Para comprender mejor de qué se trata, vamos a ver un ejemplo. Puede que queramos saludar a nuestro asistente y que nos devuelva el saludo. En este caso, creamos un *intent* que podemos llamar *Saludo*. Este *intent* se activa cada vez que saludamos a nuestro asistente. Podemos crear otro *intent* llamado *Error* para que se active cuando nuestro asistente no ha entendido algo. En definitiva, cada *intent* se asocia con una acción diferente a la que da respuesta.

- **Utterance:** se trata de las frases asociadas a cada *intent*. En el caso del ejemplo anterior, el *intent Saludo* podría tener asociadas muchas frases distintas ejemplificando lo que el asistente debe entender por un

saludo del usuario: hola, buenas, buenos días, buenas tardes, buenas noches, hey...

- **Slot o Entity:** a veces resulta útil que un *intent* tenga argumentos. Si, por ejemplo, creamos un asistente de un museo, uno de sus *intents* podría ser preguntarle dónde está un determinado cuadro. De este modo, el cuadro sería el argumento. Si estuviéramos en el museo del Prado, los posibles valores del *slot* o la *entity cuadro* serían las obras de allí, como *Las Meninas* o *El jardín de las delicias*.

Dentro de la aplicación que desarrollemos, podemos distinguir dos partes:

- **Interfaz de usuario:** permite la interacción. En el caso de los asistentes de voz, se trata de los comandos orales y las respuestas del asistente, que pueden incluir sonidos, música... En el caso de los asistentes que utilicen texto o imágenes, dichos elementos constituyen la interfaz. Si el asistente tiene varios tipos de interacción, la combinación de todas ellas es fundamental para la interfaz.
- **Backend:** se trata de la lógica detrás de la aplicación. Es lo que hace que funcione de forma correcta. Si nuestra aplicación es un juego, las acciones que podemos o no podemos realizar en cada momento son una parte de la lógica.

Además de todo esto, una aplicación debe tener un nombre de invocación. Tanto *Google Assistant* como *Alexa* deben ser activados para poder interactuar con ellos. De este modo, necesitamos llamar primero al asistente y, una vez que se ha activado, podemos abrir la aplicación deseada.

*Ok, Google, hablar con nombre de invocación.*

*Alexa, abre nombre de invocación.*

## **4.2. Hardware. Dispositivos usados**

A la hora de desarrollar el juego para las distintas plataformas, hemos usado un hardware que nos ha permitido experimentar con los sucesivos prototipos. Las dos plataformas empleadas, tanto la de *Alexa* como la de *Google*, disponen de un simulador en el que puede probarse el código. Sin embargo, usar dispositivos físicos ayuda a estudiar mejor el resultado final

del producto, tal y como será utilizado por los usuarios a los que va destinado.

En el caso de *Google Assistant*, se ha usado el altavoz *Google Nest Mini* y el asistente de un *smartphone* con *Android 7.0*. El altavoz solo tiene interacción oral, pero el asistente del móvil también incluye texto e incluso imágenes.



Figura 4.1: *Google Nest Mini*.

Para hablar con *Alexa*, se ha utilizado el altavoz *Echo Show 5*. Este dispositivo cuenta con pantalla, con lo que la interacción se realiza tanto de forma oral como gráfica.



Figura 4.2: *Amazon Echo Show 5*.

Actualmente, tanto *Alexa* como el asistente de *Google* vienen integrados en una gran variedad de dispositivos: teléfonos, altavoces, pantallas inteligentes, televisores o relojes. Su extensión es muy amplia, lo que facilita y promueve su utilización y popularidad entre los usuarios. Esta característica ha constituido una de los principales razones para seleccionar estos asistentes para implementar el juego. Su gran difusión permitirá que la aventura

tenga un gran alcance una vez publicada.

### 4.3. Alexa Developer Console

Las aplicaciones desarrolladas para *Alexa* se denominan *skills*. Para crear una *skill*, hay que acceder a la plataforma *Alexa Developer Console*. Una vez ahí, se selecciona un nombre y un idioma para la *skill*. Más adelante se pueden añadir más idiomas durante el desarrollo de la aplicación.

A continuación se elige un modelo para la *skill*. Esto implica que si vamos a realizar una aplicación de un tipo concreto, podemos empezar a desarrollarla a partir de un modelo preconstruido. Por ejemplo, si se trata de una funcionalidad relacionada con el hogar (*smart home*). En el caso de esta aventura, hemos seleccionado un modelo personalizado, de forma que fuera posible diseñar una experiencia única.

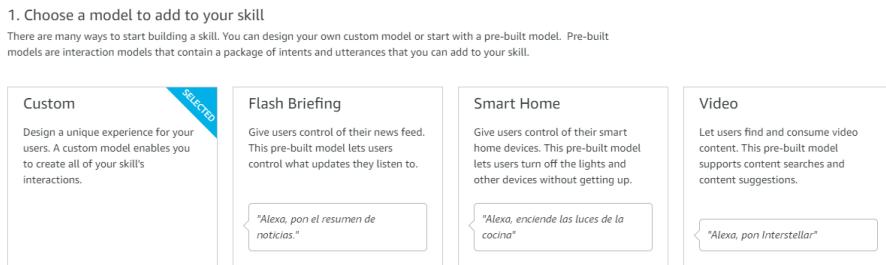


Figura 4.3: Modelo de selección de *skill* durante el desarrollo en *Alexa Developer Console*

Además, también hay que seleccionar dónde se alojará la lógica de la *skill*. Puede escribirse dentro de la propia plataforma de *Alexa* o podemos proporcionar nuestro propio *endpoint* y conectarlo con la *skill*. Para el desarrollo de la aventura, hemos elegido la plataforma de *Alexa* con el lenguaje *Node.js*.

## 2. Choose a method to host your skill's backend resources

You can provision your own backend resources or you can have Alexa host them for you. If you decide to have Alexa host your skill, you'll get access to our code editor, which will allow you to deploy code directly to AWS Lambda from the developer console.

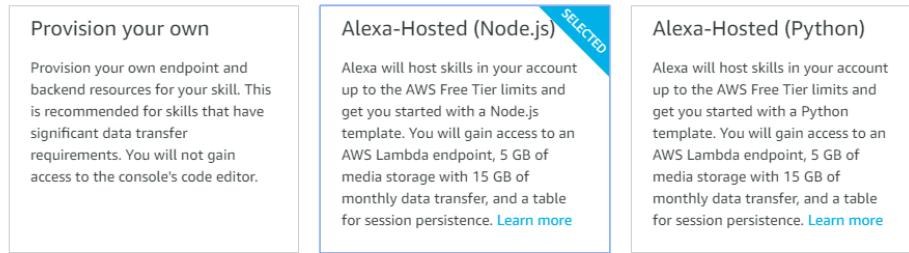


Figura 4.4: Selección del *endpoint* de la *skill* de *Alexa*.

Por último, se escoge un modelo de interacción base a partir del cual se construirá la aplicación. Esto ocurre en el caso de que la lógica de la *skill* se aloje en la consola de *Alexa*. Hemos seleccionado *Hello World Skill*, de modo que obtenemos la funcionalidad básica de un *Hola Mundo*.

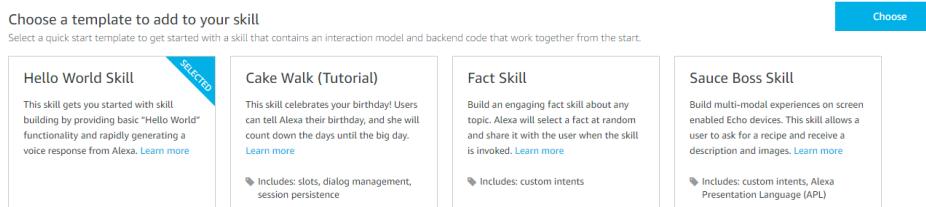


Figura 4.5: Selección del *template* para la *skill* de *Alexa*.

Una vez configurada la *skill*, empezamos con el desarrollo. Hemos de seguir varios pasos para lograr una aplicación que podamos probar.

## 1. Elegir un nombre de invocación.

En este caso, el concepto inicial de la aventura puede invocarse diciendo:

*Alexa, abre juego inicial.*

Para la aventura completa, el nombre de invocación sería:

*Alexa, abre gato encerrado.*

## 2. Intents, Utterances y Slots.

En este punto se configura la interacción con el asistente durante el juego. Para ello, se han incluido diferentes *intents* para cada una de las acciones que se pueden llevar a cabo, con sus respectivos *utterances* y *slots*.

Uno de los *intents* es, por ejemplo, el que te permite explorar la habitación en la que estás. Se denomina *LookIntent*. Las *utterances* que permiten su invocación son las siguientes:

Intents / LookIntent

Sample Utterances (6) [?](#)

What might a user say to invoke this intent? [+](#)

Mira {direction}	<a href="#">Delete</a>
Mira hacia {direction}	<a href="#">Delete</a>
Mira al {direction}	<a href="#">Delete</a>
Mirar al {direction}	<a href="#">Delete</a>
Mirar hacia {direction}	<a href="#">Delete</a>

1 - 5 of 6 [Show All](#)

Figura 4.6: Ejemplo de *intent* en *Alexa*.

Como se puede ver en la imagen, este *intent* tiene un *slot*. Dicho *slot* es la dirección en la que el jugador quiere mirar mientras explora. He llamado *Orientation* al tipo del *slot*. Sus posibles valores son:

VALUE <a href="#">?</a>	ID (OPTIONAL) <a href="#">?</a>	SYNONYMS (OPTIONAL) <a href="#">?</a>
abajo	Enter ID	Add synonym <a href="#">+</a> suelo <a href="#">X</a>
arriba	Enter ID	Add synonym <a href="#">+</a> techo <a href="#">X</a>
este	Enter ID	Add synonym <a href="#">+</a> izquierda <a href="#">X</a>
sur	Enter ID	Add synonym <a href="#">+</a> atrás <a href="#">X</a> detrás <a href="#">X</a>
norte	Enter ID	Add synonym <a href="#">+</a> adelante <a href="#">X</a> delante <a href="#">X</a>

Figura 4.7: Ejemplo *Slot Type* en *Alexa*.

### 3. Build Model.

Una vez completados todos los *intents*, las respuestas del asistente y la lógica de la *skill* son lo siguiente que debemos configurar. Dado que hemos elegido la propia consola de *Alexa* para alojar el código de la aplicación, solo es necesario seleccionar *Code* en la pestaña superior para empezar a escribir el *backend*.

Cada *intent* tiene su propia función para gestionar lo que hará el asistente cuando identifique la petición del jugador. En el caso de *LookIntent*, tenemos:

```

138 ▼ const LookIntentHandler = {
139 ▼   canHandle(handlerInput) {
140     |   return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
141     |   && Alexa.getIntentName(handlerInput.requestEnvelope) === 'LookIntent';
142   },
143 ▼   handle(handlerInput) {
144     |   const {intent} = handlerInput.requestEnvelope.request;
145     |   const orientation = intent.slots.direction.resolutions.resolutionsPerAuthority[0].values[0].value.name;
146     |   let speakOutput = '';
147
148     |   const {attributesManager} = handlerInput;
149     |   const requestAttributes = attributesManager.getRequestAttributes();
150     |   const reprompt = requestAttributes.t('REPROMPT_MSG');
151
152     |   const language = handlerInput.requestEnvelope.request.locale;
153     |   if(language === 'es-ES'){
154       |     speakOutput = logic_es.look(orientation)
155     }else if(language === 'en-US'){
156       |     speakOutput = logic_en.look(orientation)
157     }
158
159     |   return handlerInput.responseBuilder
160     |     .speak(speakOutput)
161     |     .reprompt(reprompt)
162     |     .getResponse();
163   }
164 };

```

Figura 4.8: Fragmento de código de *LookIntent*.

Además de esto, se incluyen más funciones que permiten el correcto cumplimiento de las reglas del juego. De este modo, los objetos de cada habitación están localizados en todo momento, sabemos lo que tiene el jugador en el inventario, su puntuación, las acciones que ha realizado, las pistas que ha usado...

Tras completar estos pasos, es posible probar el juego en la pestaña *Test* de la consola. También podemos usar un dispositivo con *Alexa* en el que estemos registrados con nuestra cuenta de desarrolladores.

Cuando la aplicación está terminada, se puede acceder a la pestaña *Distribution*. Tras llenar los datos de la *skill*, puede seleccionarse su publicación. La *skill* debe cumplir unos requisitos y ser aprobada para publicarse. El proceso es lento, pero una vez que se decida algo sobre nuestra *skill* recibiremos un correo informándonos.

## 4.4. Google Actions Console

Con respecto a *Google*, las aplicaciones reciben el nombre de *actions*. La plataforma de desarrollo es *Google Actions Console*. Es bastante similar a la de *Alexa*, por lo que el desarrollo de *actions* sigue un esquema muy parecido.

En primer lugar, creamos un proyecto y le damos un nombre. A continuación, podemos seleccionar la categoría de la *action* que vamos a desarrollar. Esto es, si se trata de un juego, una aplicación relacionada con el deporte, negocios, comida...

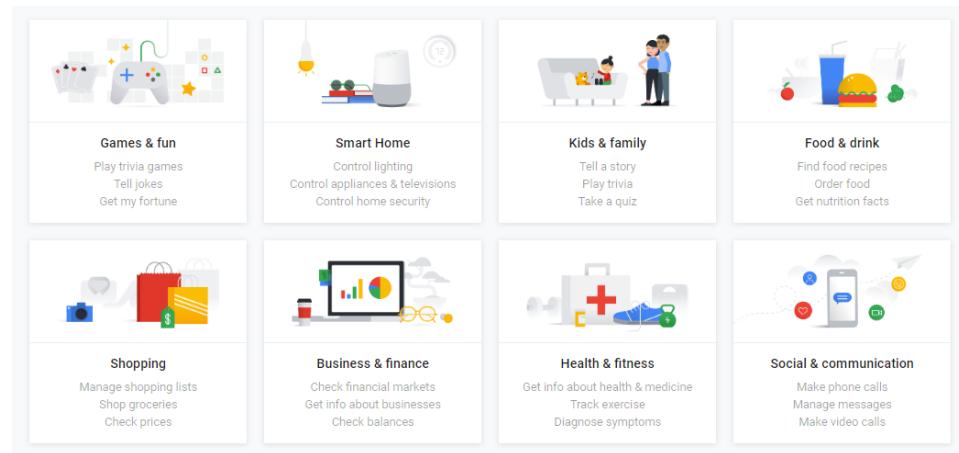


Figura 4.9: Categorías de una *action*.

Además de estas categorías, también pueden seleccionarse otras opciones para desarrollar la aplicación. En el caso del juego, hemos seleccionado *Conversational*. De este modo, hemos podido crear una experiencia conversacional personalizada.

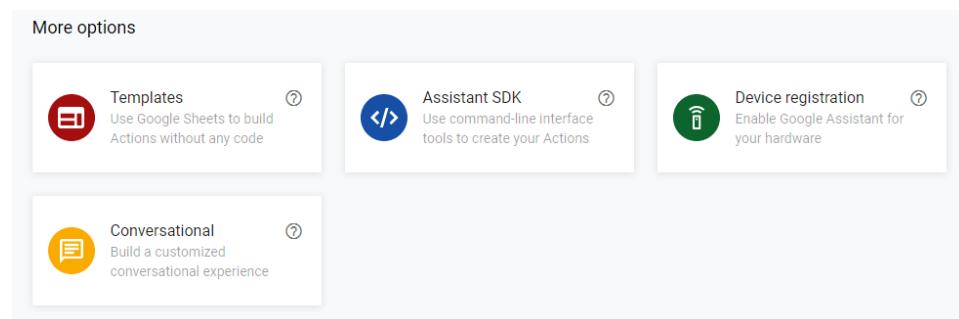


Figura 4.10: Opciones de desarrollo para la *action*.

Al igual que con *Alexa Developer Console*, hemos seguido varios pasos para implementar y probar la aventura diseñada.

### 1. Nombre de invocación.

Se han seleccionado los mismos nombres de invocación para la *action* en *Google*. Así, el concepto inicial de la aventura se invoca diciendo:

*Ok, Google, hablar con juego inicial.*

Y, para el juego completo:

*Ok, Google, hablar con gato atrapado.*

### 2. Interacción con el usuario.

Para el desarrollo de los *intents*, *utterances* y *entities*, he utilizado *Dialogflow*. De este modo, hemos creado un proyecto con el mismo nombre que la *action* de *Google*. A continuación, hemos seleccionado *Intents* en el menú de *Dialogflow*.

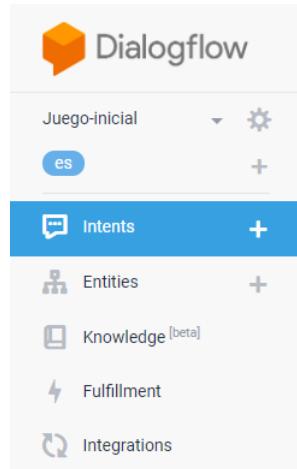


Figura 4.11: Menú *Dialogflow*.

Una vez ahí, hemos implementado los *intents* del juego. Veamos el caso de *LookIntent*.

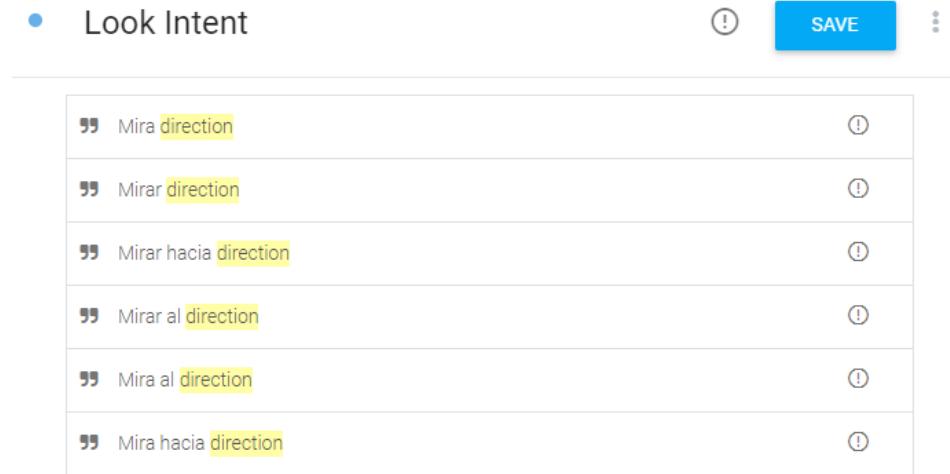


Figura 4.12: Ejemplo de *intent* en *Google*.

Al igual que en *Alexa*, este *intent* tiene un argumento que alude a la dirección en la que el jugador quiere mirar al explorar la habitación. Como se puede apreciar en la imagen, la palabra *direction* alude a dicho argumento. Los posibles valores que pueden asignarse a *direction* están almacenados en la *entity Orientation*.

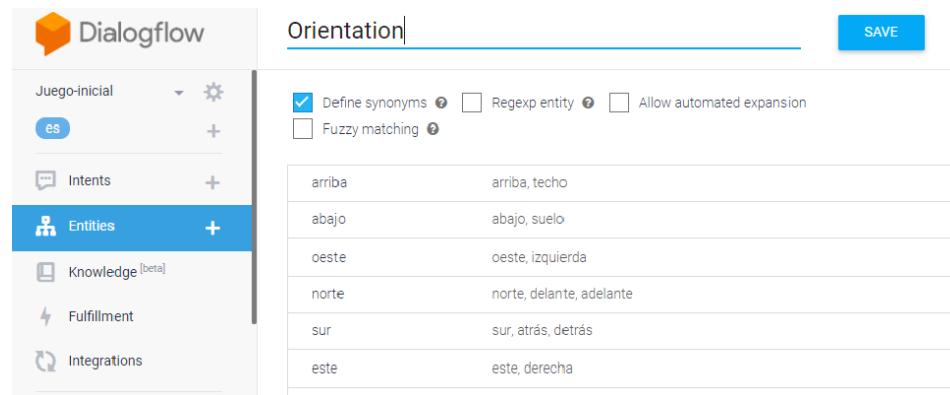


Figura 4.13: Ejemplo *Entity* en *Google*.

Las entidades se crean y se editan desde la pestaña *Entities* del menú. Para crear las respuestas del asistente, tenemos varias opciones. Podemos incluirlas dentro del propio *intent* o gestionarlas desde el *backend*, junto con el resto de la lógica del juego. En este caso, hemos optado por la segunda opción para la mayoría de los *intents*. Esto se especifica debajo de la creación de cada *intent*.

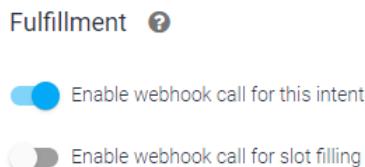


Figura 4.14: Integración externa.

La figura 4.11 aclara la opción que he seleccionado para indicar que la integración del *intent* se procesa de forma externa.

### 3. Fulfillment.

En este punto se elabora el código asociado a la aventura, la lógica subyacente a la aplicación. Para ello, tenemos que acceder a la pestaña *Fulfillment* de *Dialogflow*.

Se puede elaborar el código en el editor en línea o proporcionar un servicio web que se conecte con la *action*. En este caso, el código está alojado en nuestro propio ordenador. El modo de conectarlo con *Dialogflow* ha sido mediante la plataforma *Firebase*. Esto nos ha permitido que el desarrollo fuera más cómodo que usando el editor en línea. *Dialogflow* puede integrar el código de *Firebase* especificando la url del proyecto.

The screenshot shows the 'Fulfillment' configuration for a Dialogflow agent. The 'Webhook' section is currently disabled. The 'Inline Editor' is enabled, which allows building and managing fulfillment directly via Google Cloud Functions. Below the editor, there is a code editor window showing the contents of the 'index.js' file:

```

index.js package.json
1 {
2   "firebase": {
3     "projectId": "gato-encerrado",
4     "databaseURL": "https://gato-encerrado.firebaseio.com",
5     "storageBucket": "gato-encerrado.appspot.com",
6     "locationId": "europe-west"
7   }

```

Figura 4.15: *Fulfillment* de *Dialogflow*.

Dentro del código, hemos proporcionado toda la lógica asociada a las reglas del juego y las posibles acciones en cada momento. Además, hemos gestionado los *intents* que lo requerían. Vamos a ver el ejemplo de la función que gestiona el *intent TellScore*.



The screenshot shows the Google Actions Console interface. On the left, there are two tabs: 'index.js' and 'package.json'. The 'index.js' tab is selected and displays the following code:

```

141     conv.ask(speechText);
142 });
143
144 app.intent(TELL_SCORE, (conv) =>{
145   let speechText = 'Tienes ' + points + ' puntos.';
146   if(points === ''){
147     speechText = 'Aún no tienes puntos.';
148   }
149   conv.ask(speechText);
150 });
151
152 app.intent(LOOK, (conv) =>{
153   const orientation = conv.parameters[ORIENTATION_ENTITY].toLowerCase();
154   let speechText = 'No puedes mirar nada porque no estás en ninguna habitaci▼
155

```

Below the code editor, there is a status bar with the text 'View execution logs in the Google Cloud Console' and 'Last deployed on 04/18/2020 23:56'. To the right of the status bar is a blue 'DEPLOY' button.

Figura 4.16: Fragmento de código de *TellScoreIntent*.

Este *intent* nos informa sobre la puntuación del jugador. Puede activarse, por ejemplo, diciendo:

*Ok, Google, dime mi puntuación.*

#### 4. Integración.

Para terminar el desarrollo de la *action*, tenemos que integrar lo que se ha desarrollado en *Dialogflow*. De esta forma, se accede a la pestaña *Integrations* de *Dialogflow* y se selecciona *Google Assistant*. En este menú, hay que escribir el nombre del *intent* que se activa cuando invocamos la aplicación. En este caso, el *intent* se llama *Invocation*. Además, hay que añadir todos los *intents* que forman parte del proyecto.

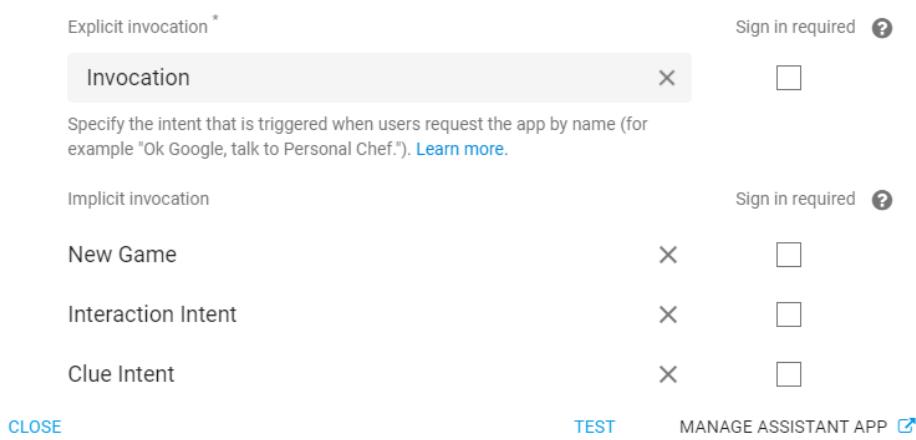


Figura 4.17: Integración con *Google Assistant*.

Por último, volvemos a la *Google Actions Console* y comprobamos que los *intents* se han añadido de forma correcta a las acciones de la aplicación. Para ello, hay que acceder a la pestaña de *Develop* y seleccionar *Actions*. Ahí deberían aparecer los *intents* creados en *Dialogflow*.

The screenshot shows the 'Actions' tab in the Google Actions Console. On the left, there is a sidebar with options: Invocation, Actions (which is selected), Theme customization, Account linking, and Backend services. The main area displays a table of actions:

Name	Fulfillment type	Fulfillment tool
actions.intent.MAIN	Conversational	Dialogflow
New Game	Conversational	Dialogflow
Interaction Intent	Conversational	Dialogflow
Clue Intent	Conversational	Dialogflow
Where Intent	Conversational	Dialogflow
Default Interaction Intent	Conversational	Dialogflow
Use Intent	Conversational	Dialogflow

At the top right of the table, there are buttons for 'Test' and 'Add Action'. The status bar at the top indicates 'Juego inicial'.

Figura 4.18: *Actions* integradas en el juego.

Cuando todos estos pasos se han completado, podemos pasar a probar la *action*. Para ello, se puede acceder a la pestaña *Test* de la *Google Actions Console* o utilizar *Google Assistant* en cualquiera de nuestros dispositivos con nuestra cuenta de desarrolladores.

En el momento en el que queramos proceder a la publicación de la *action*, tenemos que seleccionar la pestaña *Deploy*. Tras llenar los datos solicitados, podemos escoger publicar la aplicación, lo cual será posible después

de pasar de forma exitosa un proceso de verificación y prueba. También podemos escoger lanzar la acción para *Beta testers* o *Alpha testers*. Esa opción es muy útil para recibir una retroalimentación de los usuarios previa al lanzamiento oficial de la aplicación.

## **Capítulo 5**

# **Metodologías a usar en el proyecto**

A la hora de llevar a cabo este proyecto, se han combinado varias metodologías. Esto ha permitido desarrollar prototipos del juego de forma sucesiva y darle la importancia necesaria a los usuarios dentro de todo el proceso.

### **5.1. Diseño centrado en el usuario (DCU)**

Podría definirse como el diseño de software que responde a las necesidades de los usuarios. Es una filosofía sin un método claro de llevarse a la práctica. Esto provoca que tenga una gran variedad de definiciones, pero todas son compatibles entre sí y coinciden en varios aspectos:

1. Es multidisciplinar.
2. Es iterativo.
3. Tiene como objetivo la creación de productos usables y satisfactorios.
4. Está orientado a que los usuarios participen durante todo el proceso.

A pesar de que esta metodología se denomine diseño centrado en el usuario, se aplica durante todas las fases de desarrollo del software, no solo durante el diseño. Además, es compatible con otros métodos (Sánchez, J., 2011).

A la hora de implementar el DCU, debemos tener en cuenta que se trata de un proceso cíclico. Tenemos que situar al usuario en el centro de las de-

cisiones. Además, debemos considerar qué objetivos queremos que satisfaga el producto que desarrollemos. La usabilidad se evalúa de forma iterativa y se mejora incrementalmente.



Figura 5.1: Proceso del DCU. (Hassan-Montero, Y.; Ortega-Santamaría, S., 2009).

Tal y como podemos observar en la figura, podemos establecer cuatro fases en el proceso:

- En primer lugar, debemos entender y especificar el contexto de uso. Esto hace referencia al público al que va dirigido el producto, el uso que le darán y en qué condiciones.
- A continuación, debemos especificar los requisitos. Este paso pretende establecer los objetivos que se quieren satisfacer con el producto. Alude tanto a los objetivos de los usuarios como a los del proveedor.
- El siguiente paso es producir soluciones de diseño. Es una etapa que puede repetirse desde que se propone un primer diseño hasta que se obtiene el diseño final.
- Por último, procedemos a la fase de evaluación. Se trata del paso más importante, ya que valida los diseños propuestos o detecta problemas que deben ser solucionados (Hassan-Montero, Y.; Ortega-Santamaría, S., 2009).

Como se ha mencionado anteriormente, el DCU puede ser considerado una filosofía de diseño. Esto quiere decir que no tiene por qué seguir este

esquema concreto, sino que puede ser interpretado y llevado a la práctica de formas diferentes. Lo más importante es tener en cuenta que el usuario debe ser el centro de este diseño.

## 5.2. Desarrollo ágil

La metodología ágil se basa en el desarrollo iterativo e incremental. La idea fundamental es mejorar el software de forma continua, obteniendo prototipos que funcionen. Su origen está en el *Manifiesto por el Desarrollo de Software Ágil*, publicado en 2001. En él se recogen doce principios. Algunas de las bases del desarrollo ágil son:

1. Priorizar a los individuos y a las interacciones sobre los procesos y las herramientas.
2. Trabajar el software por encima de la documentación.
3. Colaborar con los clientes en lugar de negociar contratos.
4. Anteponer las respuestas a los cambios sobre el seguimiento de un plan.

Se trata, por tanto, de una forma de desarrollo que no se basa en la planificación, sino en la obtención de respuestas flexibles frente al cambio. Los proyectos evolucionan en función de sus necesidades, rompiendo con el clásico modelo de cascada (*Wikipedia, la enciclopedia libre*, 2020).

La obtención de las sucesivas iteraciones puede describirse en un proceso cíclico que engloba todas las fases de desarrollo.



Figura 5.2: Ciclo del desarrollo ágil. (Junior, A., 2018).

Para implementar un proceso ágil, se divide un proyecto en partes más pequeñas, denominadas *historias de usuario*. Cada una de ellas es una característica deseable para el software. Los desarrolladores dividen las historias en pequeños subconjuntos sobre los que van trabajando. Cada uno de estos subconjuntos constituye una iteración sobre la que se establece un plazo.

Cada iteración produce un producto usable. De este modo, puede ser probado por los usuarios y adaptarse en función de sus comentarios.

También es importante conocer el concepto del *sprint*. Se trata de un período de tiempo durante el que el trabajo específico debe ser completado. La duración del *sprint* es siempre la misma. Esta distribución equitativa de los tiempos permite la revisión continua del proyecto.

Algunos ejemplos de la filosofía de las metodologías ágiles llevada a la práctica son:

- **Scrum:** puede que sea la más popular. Es un enfoque de *Scrum Alliance*. Describe cómo llevar los conceptos a cabo en el equipo de desarrollo y la administración. El *ScrumMaster* lidera el trabajo, pero el equipo tiene la responsabilidad de lograr los objetivos.
  
- **XP:** significa *Extreme Programming*. Se enfoca más hacia el aspecto técnico. Sus *sprints* son más cortos que los usados en *Scrum*. Además, es más flexible para establecer cambios en mitad de un *sprint* (Junior, A., 2018).

### **5.3. Fases de desarrollo de un videojuego**

Un videojuego puede considerarse un software, pero también tiene relación con la producción audiovisual. Esto implica que no existe una metodología propia para su diseño y desarrollo. De este modo, cada compañía fija su propia filosofía de trabajo. Sin embargo, pueden establecerse tres etapas como punto común: preproducción, producción y postproducción.

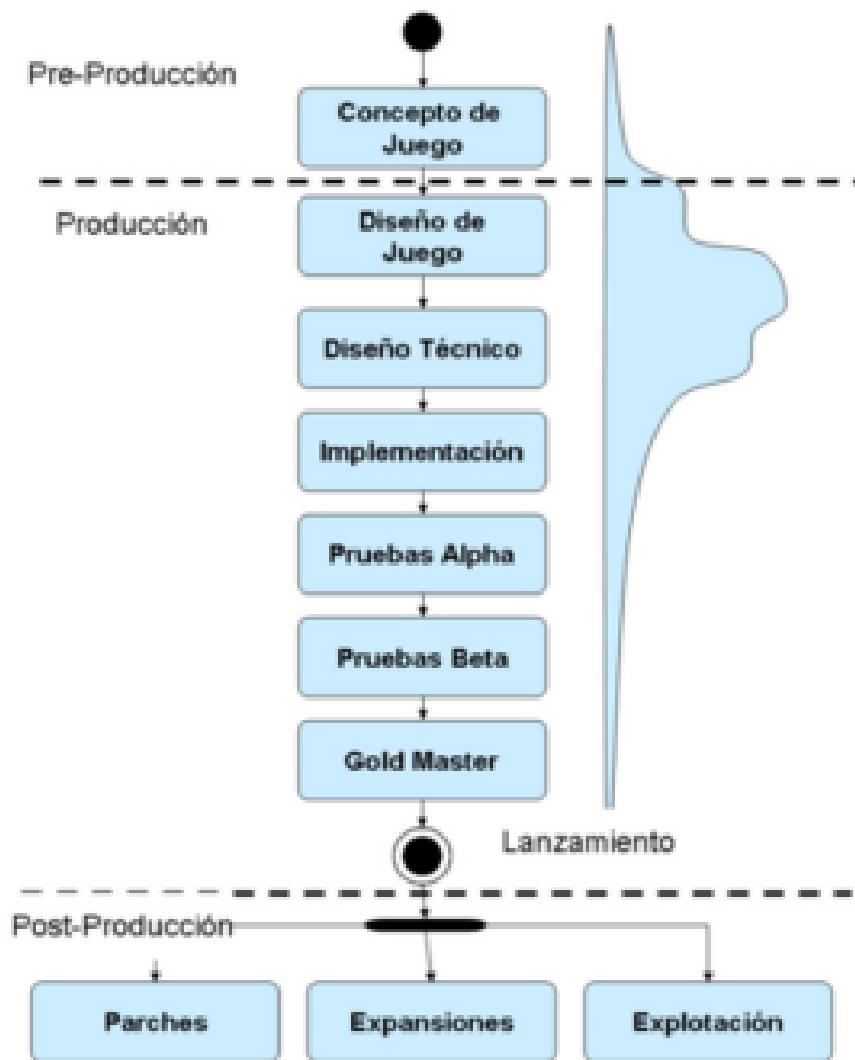


Figura 5.3: Fases de producción de un videojuego. (Manrubia Pereira, A. M., 2014).

La preproducción es la fase en la que se define el juego. Consiste en la concepción general del mismo, elaborando sus aspectos más relevantes y los términos en los que se llevará a cabo su materialización.

La fase de producción es la de mayor confluencia de actividades. En ella se diseñan los elementos del juego, se desarrolla el sonido, la interfaz, los gráficos, las normas, el motor... Además, se procede a la implementación y a las pruebas.

Por último, la posproducción abarca el seguimiento del videojuego una vez que sale al mercado. Se relaciona con el marketing y la publicidad (Manrubia Pereira, A. M., 2014).

#### **5.4. Aplicación de las metodologías al proyecto**

En el caso de este proyecto, se han combinado el DCU y el desarrollo ágil. No hay una respuesta definitiva para llevar esto a cabo. De este modo, nos hemos basado en el objetivo de la usabilidad. Para ello, hemos seguido los principios de involucrar a los usuarios en el proceso y refinar el producto mediante evaluaciones.

A la hora de desarrollar el juego, hemos tratado de seguir las premisas de la metodología ágil. Así, la medida principal de progreso ha sido el trabajo sobre el software. Hemos tratado de llevar a cabo un proceso incremental e iterativo, añadiendo cambios sobre la marcha, conforme iban surgiendo nuevas necesidades sobre lo anteriormente desarrollado. También hemos intentado centrarnos en la satisfacción del usuario. De esta forma, hemos procurado que la interacción del jugador con el asistente resultara sencilla. Por último, hemos realizado pruebas con los usuarios para obtener sus impresiones y ejecutar cambios al respecto según los aspectos comentados.

Dado que tanto la metodología ágil como el DCU pueden llevarse a cabo mediante procesos cíclicos, la combinación de ambos ha resultado bastante intuitiva. El resultado ha sido un proyecto que, poniendo al usuario como centro de las decisiones de diseño, ha ido construyendo sucesivos prototipos hasta obtener un producto satisfactorio.

Además de estas dos metodologías, es interesante considerar las tres etapas de producción de los videojuegos vistas en el punto anterior. Esto se debe a que este proyecto presenta ciertas similitudes con un videojuego aunque no pueda considerarse uno por completo. Debido al desarrollo iterativo implementado, estas fases de producción se han repetido en mayor o menor medida para cada nuevo prototipo.

- La preproducción ha consistido en el desarrollo del concepto inicial del juego.
- La etapa de producción se corresponde con la elaboración del código y la ejecución del proyecto para las distintas plataformas (*Google Assistant* y *Alexa*). Las pruebas para los usuarios también se incluyen en esta fase, así como las imágenes y los sonidos añadidos.

- Por último, dado que el juego desarrollado se trata de un proyecto pequeño e individual, la fase de posproducción no ha sido contemplada en gran medida durante su desarrollo. Aunque se ha procedido a la publicación de la aventura para *Alexa* y *Google Assistant*, aún es pronto para hacer un seguimiento de la aplicación y no se ha realizado ninguna campaña de marketing para publicitarla.

En definitiva, la combinación de estas tres metodologías ha supuesto que cada entrega haya constado de una planificación, un desarrollo de la implementación y una fase en la que se ha comprobado que los cambios funcionaran de forma adecuada.

# Capítulo 6

## Plan de entregas

Aunque la metodología seguida sea principalmente ágil, podemos estructurar el desarrollo del juego en distintas entregas basadas en los prototipos llevados a cabo. En este caso, la primera entrega o entrega 0 sería la versión inicial del juego. Las modificaciones y las ampliaciones ejecutadas después se estructurarían en diversas entregas tal y como se puede observar en la siguiente tabla.

Plan de entregas							
Entrega	Objetivo						
<b>0</b>	Obtener concepto inicial del juego.						
	<table border="1"><thead><tr><th>Iteración</th><th>Objetivo</th></tr></thead><tbody><tr><td>1</td><td>Analizar plataformas de desarrollo (Alexa, Google, Voxa).</td></tr><tr><td>2</td><td>Construir prototipo inicial con funcionalidad básica.</td></tr></tbody></table>	Iteración	Objetivo	1	Analizar plataformas de desarrollo (Alexa, Google, Voxa).	2	Construir prototipo inicial con funcionalidad básica.
Iteración	Objetivo						
1	Analizar plataformas de desarrollo (Alexa, Google, Voxa).						
2	Construir prototipo inicial con funcionalidad básica.						
<b>1</b>	Primera versión del juego.						
	<table border="1"><thead><tr><th>Iteración</th><th>Objetivo</th></tr></thead><tbody><tr><td>1</td><td>Extender el concepto inicial del juego de forma estática.</td></tr><tr><td>2</td><td>Incorporar la nueva funcionalidad oral a la aventura.</td></tr></tbody></table>	Iteración	Objetivo	1	Extender el concepto inicial del juego de forma estática.	2	Incorporar la nueva funcionalidad oral a la aventura.
Iteración	Objetivo						
1	Extender el concepto inicial del juego de forma estática.						
2	Incorporar la nueva funcionalidad oral a la aventura.						

<b>2</b>	Funcionalidad más dinámica.						
	<table border="1"> <thead> <tr> <th>Iteración</th><th>Objetivo</th></tr> </thead> <tbody> <tr> <td>1</td><td>Añadir historia alternativa a la trama principal.</td></tr> <tr> <td>2</td><td>Dividir habitaciones en archivos separados creando un mayor nivel de abstracción.</td></tr> </tbody> </table>	Iteración	Objetivo	1	Añadir historia alternativa a la trama principal.	2	Dividir habitaciones en archivos separados creando un mayor nivel de abstracción.
Iteración	Objetivo						
1	Añadir historia alternativa a la trama principal.						
2	Dividir habitaciones en archivos separados creando un mayor nivel de abstracción.						
<b>3</b>	Componente visual y sonidos.						
	<table border="1"> <thead> <tr> <th>Iteración</th><th>Objetivo</th></tr> </thead> <tbody> <tr> <td>1</td><td>Imágenes añadidas mediante Alexa Presentation Language.</td></tr> <tr> <td>2</td><td>Efectos de sonido añadidos al juego.</td></tr> </tbody> </table>	Iteración	Objetivo	1	Imágenes añadidas mediante Alexa Presentation Language.	2	Efectos de sonido añadidos al juego.
Iteración	Objetivo						
1	Imágenes añadidas mediante Alexa Presentation Language.						
2	Efectos de sonido añadidos al juego.						
<b>4</b>	Funcionalidad de guardar la partida.						
	<table border="1"> <thead> <tr> <th>Iteración</th><th>Objetivo</th></tr> </thead> <tbody> <tr> <td>1</td><td>Añadidas variables que almacenen los datos de la partida en una base de datos para recuperarlos en caso de que se quiera continuar con ella en otro momento.</td></tr> </tbody> </table>	Iteración	Objetivo	1	Añadidas variables que almacenen los datos de la partida en una base de datos para recuperarlos en caso de que se quiera continuar con ella en otro momento.		
Iteración	Objetivo						
1	Añadidas variables que almacenen los datos de la partida en una base de datos para recuperarlos en caso de que se quiera continuar con ella en otro momento.						
<b>5</b>	Adición de un nuevo idioma.						
	<table border="1"> <thead> <tr> <th>Iteración</th><th>Objetivo</th></tr> </thead> <tbody> <tr> <td>1</td><td>Añadida versión completa del juego para <i>Alexa</i> y <i>Google Assistant</i> disponible en inglés.</td></tr> </tbody> </table>	Iteración	Objetivo	1	Añadida versión completa del juego para <i>Alexa</i> y <i>Google Assistant</i> disponible en inglés.		
Iteración	Objetivo						
1	Añadida versión completa del juego para <i>Alexa</i> y <i>Google Assistant</i> disponible en inglés.						

Cuadro 6.1: Plan de entregas

## Capítulo 7

# Desarrollo Entregas e iteraciones

Esta sección aborda el desarrollo del juego. Una vez implementado el prototipo inicial, se ha procedido a mejorarlo y hacerlo más complejo de forma progresiva, tal y como queda reflejado en la tabla de las entregas del capítulo anterior.

### 7.1. Concepto del juego

A partir del concepto inicial del juego, se ha desarrollado una aventura más compleja y larga. El objetivo de escapar de una habitación se convierte en el objetivo de escapar de una casa con múltiples retos y habitaciones. De este modo, sigue combinando las características de las aventuras conversacionales, las *escape room*, los librojuegos y los juegos de rol, pero amplía su funcionalidad en muchos aspectos.

Como se ha explicado en el capítulo 3, el objetivo del juego se centra en escapar. No obstante, también se van descubriendo objetos relacionados con los gatos conforme se avanza. Durante toda la aventura, se examinan dichos objetos y el jugador se sorprende de ir encontrándolos por su supuesta casa, ya que nunca ha tenido mascota. Finalmente, la clave de la huida resultan ser esos elementos que se van recogiendo y un gato que aparece en la habitación final. Además, existe una trama alternativa que se relaciona con un gato fantasma.

## 7.2. Guía del juego

El nombre de invocación del juego en español es *gato encerrado* para *Alexa* y *gato atrapado* para *Google Assistant*. Además, se ha añadido una versión en inglés en la que se invoca con *locked cat* en *Alexa* y *trapped cat* en *Google Assistant*. Una vez iniciado, le da la bienvenida al jugador. Si elige nuevo juego o reiniciar juego, la aventura da comienzo. En el caso de que ya se hubiera iniciado una partida, puede continuarse.

### 7.2.1. Habitaciones

Dado que se trata de una versión mucho más amplia de lo que indicaba el concepto inicial, en este caso disponemos de diez habitaciones.

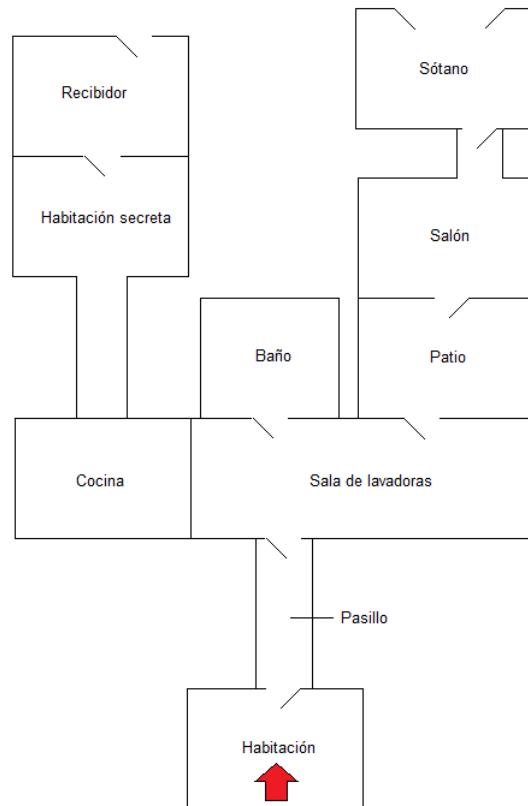


Figura 7.1: Mapa de la casa

La habitación en la que se inicia el juego es la marcada con una flecha roja.

- **Habitación:** se trata del propio cuarto del personaje. Se despierta en su cama y a partir de ahí comienza la aventura.
- **Pasillo:** conecta el cuarto con la sala de lavadoras.
- **Sala de lavadoras:** conecta el pasillo con dos habitaciones: el baño y el patio. Además, se puede acceder a la cocina si se interactúa con la pared.
- **Baño:** se accede a él desde la sala de lavadoras. No tiene salida.
- **Patio:** se accede a él desde la sala de lavadoras. Conecta con el salón.
- **Salón:** se accede desde el patio y permite ir al sótano.
- **Sótano:** se accede desde el salón. Se trata de la última habitación y, por tanto, conduce a la salida.
- **Cocina:** se llega a ella a través de la sala de lavadoras. Sin embargo, no permite volver atrás. De este modo, solo conduce a la habitación secreta.
- **Habitación secreta:** tampoco permite volver a la cocina. Tan solo se puede acceder a la entrada.
- **Recibidor:** conecta la habitación secreta con la salida de la casa. Por tanto, se puede escapar desde esta habitación.

### 7.2.2. Eventos del juego

A continuación se detallan los eventos que pueden ocurrir en el juego de forma ordenada:

- **Habitación:** cuando empieza el juego, apenas hay luz en el dormitorio. Solo se ve un interruptor en la pared. Si se realiza la acción *mirar interruptor*, se pulsará el botón y se podrá ver bien lo que hay en la habitación. Además, se obtienen 10 puntos. Esto es lo que se encuentra en cada dirección:
  - **Norte:** el interruptor que se ha pulsado y la puerta. Se necesita la llave para abrir la puerta. Una vez que se abre, se consiguen 10 puntos y el juego termina, diciendo la puntuación final.
  - **Este:** hay un cuadro y un armario. Cuando se interacciona con el cuadro, se cae una llave al suelo que puede cogérse y se obtienen diez puntos. Si el jugador la coge, la guarda en el inventario. En el armario hay una trampilla que está atascada. Si se rompe con

el martillo se consiguen otros 10 puntos y se encuentra comida de gato. Una vez que el personaje la ha encontrado, se le plantea la opción de comérsela o guardarla. Si elige lo primero, pierde veinte puntos. Si selecciona lo segundo, obtiene veinte puntos.

- **Oeste:** están la cama y la almohada. En la almohada hay un martillo.
  - **Sur:** hay un escritorio con un cajón y una ventana. El cajón contiene una linterna que puede cogerse. También puede tomarse la decisión de romper la ventana con el martillo diciendo *usar martillo en ventana*. En este caso, se pierden 20 puntos y se muere al caer por la ventana, con lo que acaba el juego.
  - **Arriba:** está la lámpara que se ha encendido al principio. Si usa la silla, el jugador puede asomarse y ve que dentro hay unas pilas. Al hacer esto, obtiene diez puntos.
  - **Abajo:** está el suelo. Hay una silla.
- **Pasillo:** una vez que se abre la puerta de la habitación, se puede ir al pasillo. También está oscuro, con lo cual habrá que combinar la linterna y las pilas y utilizar la linterna para poder examinar lo que hay alrededor. Estas dos acciones dan 10 puntos cada una.
- **Norte:** una puerta con candado. Se necesita una combinación de cuatro números para abrirlo. Cuando se acierta el código, se obtienen 20 puntos.
  - **Este:** hay un interruptor. Al pulsarlo, se desactiva la secadora de la sala de lavadoras, lo cual permite coger un objeto que se encuentra en su interior. Si vuelve a pulsarse, se activa de nuevo la secadora.
  - **Oeste:** hay un cuadro de unos gatos jugando al póker. Al mirarlo, se cae una carta al suelo y se ganan 10 puntos. Cuando se coge, puede leerse. Se trata de una carta misteriosa firmada a nombre de ECBI. Si se cambian las letras por el número de su posición en el alfabeto, se obtiene el código de cuatro números que permite salir del pasillo: 5329.
  - **Sur:** se encuentra la habitación.
  - **Arriba:** solo el techo.
  - **Abajo:** el suelo.
- **Sala de lavadoras:** una vez que se accede a la sala de lavadoras se escucha un teléfono. Además, aparece algo que atraviesa la pared. En este caso, la habitación sí tiene luz.

- **Norte:** hay dos puertas: una de cristal y una con un cerrojo. La puerta de cristal puede romperse con el martillo y da paso al baño. La puerta con cerrojo puede abrirse con un destornillador, ya que el cerrojo está atornillado. Esta puerta conduce al patio. La apertura de cada una de las dos puertas otorga 10 puntos.
  - **Este:** hay una lavadora y una secadora. Dentro de la secadora se encuentra un ovillo de lana. El jugador tiene la opción de cogerlo o jugar con él. Si lo coge obtiene 20 puntos. Si juega con él los pierde. Además, jugar con el ovillo lo deshace y le impide cogerlo.
  - **Oeste:** está el teléfono que ha sonado cuando se ha accedido a la habitación. Si se coge, se escucha una voz que da un mensaje misterioso. Junto con el mensaje anterior de la carta, servirá en la última habitación. También puede examinarse la pared por la que se ha visto desaparecer al fantasma. Si se hace, se atravesia y se aparece en la cocina. La pared vuelve a cerrarse una vez que se pasa, con lo que no se puede volver a la sala de lavadoras.
  - **Sur:** se encuentra el pasillo.
  - **Arriba:** el techo, en el que hay una claraboya.
  - **Abajo:** el suelo.
- **Baño:** esta habitación no tiene salida.
- **Norte:** hay un lavabo y un espejo.
  - **Este:** hay un retrete. Dentro se puede encontrar el destornillador que servirá para abrir la puerta con cerrojo. Si se mira en el retrete se obtienen 10 puntos.
  - **Oeste:** hay una ventana y un botiquín. Dentro del botiquín se puede encontrar un bote de alcohol desinfectante. Detrás de la ventana hay un rollo de papel higiénico con unos símbolos incritos. Si se rompe la ventana con el martillo, puede cogerse el papel higiénico y se ganan 10 puntos. El hecho de coger el papel higiénico también da 10 puntos. Los símbolos son un árbol, una nube, una hoguera y un río. Ese código sirve para abrir la puerta que conecta el patio con el salón.
  - **Sur:** se encuentra la sala de lavadoras.
  - **Arriba:** el techo con unas cuantas telarañas y una araña.
  - **Abajo:** el suelo, que está bastante sucio. Hay unas huellas de un animal doméstico.
- **Patio:** se accede desde la puerta con cerrojo.
- **Norte:** hay una puerta con cuatro símbolos: aire, agua, tierra y fuego. Para abrirla, tienen que pulsarse en cierto orden. La clave

está en el rollo de papel higiénico que se encuentra en el baño. Según los símbolos que aparecen en él, la combinación es tierra, aire, fuego y agua. Una vez que se abre, se ganan 10 puntos.

- **Este:** hay una mesa con una bandeja de galletas.
  - **Oeste:** hay un agujero en la pared. Del agujero sale un ratón de juguete. Se puede apretar o coger el ratón de juguete. En el primer caso, se rompe y se pierden 20 puntos. En el segundo caso, se ganan 20 puntos.
  - **Sur:** se ve la sala de lavadoras.
  - **Arriba:** se ve el cielo.
  - **Abajo:** el suelo, en el que hay migas de galletas. También puede encontrarse un reproductor mp3. Si se coge, se activa un audio. Es el tercer mensaje misterioso que sirve en la última habitación.
- **Salón:** es la penúltima habitación si se sigue la trama principal de la historia.
- **Norte:** al fondo se ven unas escaleras que bajan a una puerta. La puerta está atascada. Puede abrirse usando una radiografía, lo que da 10 puntos.
  - **Este:** hay una televisión y una caja fuerte. Si se examina la televisión, se escucha un poema de Antonio Machado. La caja fuerte tiene un código de cuatro números. Una vez abierta, se consiguen 10 puntos y se encuentra una foto de un gato y un cuenco de comida. Si se coge la foto, puede leerse un texto. Se trata del último texto misterioso que sirve como ayuda al final. Se puede coger o lanzar el cuenco. Si se coge, se consiguen 20 puntos, pero si se lanza se pierden.
  - **Oeste:** hay una estantería con libros. Si el jugador dice leer y el nombre de un autor de los que están en la estantería, escucha su fecha de nacimiento. Como el poema de la televisión era de Antonio Machado, se necesita su fecha de nacimiento. Esa es la combinación de la caja fuerte: 1875.
  - **Sur:** se ve el patio.
  - **Arriba:** hay un conducto de ventilación. Si el personaje usa la silla, gana 10 puntos y encuentra una radiografía dentro del conducto.
  - **Abajo:** está el suelo.
- **Sótano:** es la última habitación de la trama principal.
- **Norte:** se ve la puerta del garaje y un interruptor que no sirve para nada.

- **Este:** hay un coche. Una vez que se consiguen las llaves, puede abrirse y ganar 10 puntos. En el maletero aparece un gato que espera que le den sus cosas: el cuenco, el ovillo de lana, el ratón de juguete y la comida de gato. Para saber el orden en el que hay que dárselas, el jugador debe fijarse en las cuatro pistas misteriosas que ha ido descubriendo. Cada una de estas pistas hace referencia a uno de los objetos. Hay que ordenar las pistas de la uno a la cuatro. De este modo, el objeto al que haga referencia la primera pista será el objeto que debe dársele al gato inicialmente. Si se le dan las cosas en un orden equivocado, se pierden 30 puntos y el gato se enfada y araña al personaje. En ese caso, debe usarse el bote de alcohol que estaba en el baño antes de diez segundos o el jugador se desmayará. Si llega a desmayarse, se acaba el juego y se pierden otros 30 puntos al morir. Si se consigue usar el alcohol a tiempo, se obtienen 10 puntos, el gato se pone contento y suelta el mando del garaje, con el que se puede abrir la puerta por otros diez puntos y escapar. Además, puede cogerse al gato, ganando 50 puntos. Puede terminarse el juego aunque el usuario no se lo lleve, pero se va triste, sin haber cumplido el desafío por completo. En el caso de que no se le den cuatro objetos al gato, ocurren otras circunstancias. Para empezar, cuando el jugador le ofrece menos de cuatro objetos, el orden deja de importar. Si le da un objeto, el gato se enfada y lo araña. Aquí no tiene la posibilidad de curarse, así que se desmaya y pierde el juego junto con 30 puntos. Si le da dos objetos, también lo araña y pierde 30 puntos. Sin embargo, este caso es igual que si le da los cuatro objetos en orden equivocado. Puede curarse a tiempo con el alcohol e intentarlo de nuevo. Por último, si le da tres objetos, gana diez puntos y el gato no se enfada.
- **Oeste:** hay un arenero y una estantería con un juego de mesa. Dentro del arenero se encuentran las llaves del coche. Además, mirar el arenero da 10 puntos. Si se mira el juego, se puede ver que es una versión del *Cluedo* en la que se asesina a un gato en lugar de a una persona.
- **Sur:** se ve el salón.
- **Arriba:** el techo con una lámpara que está encendida.
- **Abajo:** se encuentran las huellas que se han visto en algunas habitaciones previas.
- **Cocina:** se accede a ella al examinar la pared de la sala de lavadoras. Se trata de la trama alternativa del juego. Una vez que se entra en la cocina, no se puede volver atrás.
  - **Norte:** solo hay una pared estropeada.

- **Este:** la pared por la que se ha entrado, pero se ha cerrado y no se puede volver a la sala de lavadoras.
  - **Oeste:** hay un frigorífico.
  - **Sur:** se ven unos fogones y una encimera con cajones. En uno de los cajones se encuentra una nota. Si se coge se obtienen 10 puntos.
  - **Arriba:** solo está el techo.
  - **Abajo:** hay una trampilla. Al colocarse sobre ella, se rompe y el jugador cae. Esto lo lleva a un túnel oscuro. Avanza por él hasta llegar a una habitación secreta. Al haber caído por un agujero, no puede volver a la cocina.
- **Habitación secreta:** se llega a ella a través del túnel y no se puede volver atrás. Parece la habitación de un gato. Vuelve a aparecer el fantasma yéndose de la habitación.
- **Norte:** al fondo se ve una puerta. Puede abrirse con una tarjeta, obteniendo 10 puntos.
  - **Este:** a la derecha hay una pecera y un arenero.
  - **Oeste:** hay un rascador.
  - **Sur:** el túnel por el que se ha llegado a la habitación.
  - **Arriba:** se ve una lámpara que ilumina la habitación.
  - **Abajo:** hay una alfombra. Si se examina, se obtienen diez puntos y se ve la tarjeta que permite abrir la puerta.
- **Recipientes:** es la última habitación de la trama alternativa. Por lo tanto, permite escapar y terminar el juego.
- **Norte:** se ve una puerta muy grande. Se abre con una llave dorada. Una vez que consigue abrirse, se obtienen 10 puntos. Además, la partida termina, ya que se ha conseguido escapar. Si el jugador se lleva al gato, gana 50 puntos más.
  - **Este:** a la derecha hay una mesa. Si se examina, hay documentos de un gato. Al leerlos, se descubre que el gato es alérgico al atún.
  - **Oeste:** hay un cuadro de un gato blanco. Cuando el personaje se acerca a mirarlo, gana diez puntos y vuelve a aparecer el fantasma. Resulta ser el espíritu del gato. Se pone delante y deja caer un frasco de veneno. Le da la opción al jugador de bebérselo o de derramarlo. Si se lo bebe, gana 30 puntos y el gato se pone feliz. Además, deja caer la llave dorada que sirve para escapar. Si lo derrama, el gato se enfada y empieza a gritar. Acaba desmayándose y perdiendo 30 puntos. Una vez que se ha desmayado, el juego termina.

- **Sur:** se ve la habitación secreta por la que se ha entrado.
- **Arriba:** solo está el techo. La habitación está iluminada, pero no hay ninguna lámpara o ventana.
- **Abajo:** se ve el cadáver de un gato rodeado de restos de atún, con lo que se descubre que fue envenenado.

### 7.3. Entregas

El proyecto no tiene una planificación total debido al uso de la metodología ágil. Sin embargo, los cambios añadidos a lo largo de las distintas iteraciones pueden dividirse en entregas.

#### 7.3.1. Entrega 0

La entrega 0 desarrolla una primera versión del juego basada en el concepto inicial.

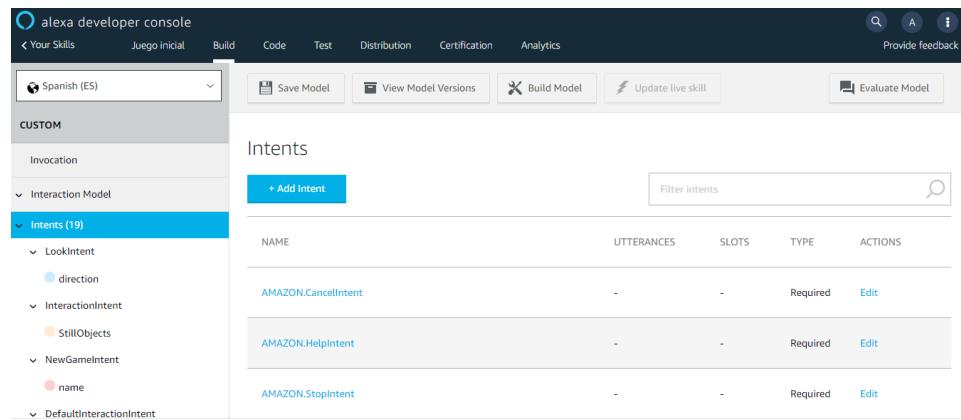
En primer lugar, se ha realizado un análisis de los dos asistentes en los que se ha desarrollado el prototipo (*Alexa* y *Google*). Para ello, hemos probado sus funcionalidades, el modo en el que se puede interactuar con ellos...



Figura 7.2: Prueba con el asistente de *Google*.

Una vez que nos hemos familiarizado con el uso de estos asistentes, hemos aprendido cómo usar sus respectivas plataformas para elaborar un prototipo inicial del juego denominado *juego inicial*.

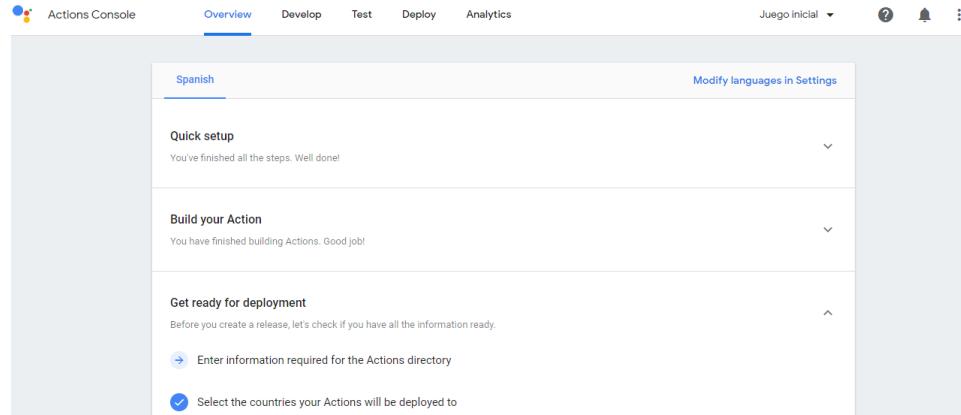
En el caso de *Alexa*, la plataforma usada es *Alexa Developer Console*.



The screenshot shows the Alexa Developer Console interface. On the left, there's a sidebar with 'Your Skills' (selected), 'Juego inicial', 'Build', 'Code', 'Test', 'Distribution', 'Certification', and 'Analytics'. Below that is a language dropdown set to 'Spanish (ES)'. Under 'CUSTOM', there's an 'Invocation' section and an 'Interaction Model' section. The 'Interaction Model' section is expanded, showing 'Intents (19)'. A tree view lists 'LookIntent' (with 'direction' slot), 'InteractionIntent' (with 'StillObjects' slot), 'NewGameIntent' (with 'name' slot), and 'DefaultInteractionIntent'. To the right, the 'Intents' table has columns: NAME, UTTERANCES, SLOTS, TYPE, and ACTIONS. It lists three intents: 'AMAZON.CancelIntent', 'AMAZON.HelpIntent', and 'AMAZON.StopIntent', all marked as 'Required'.

Figura 7.3: Prototipo inicial en *Alexa*.

En el caso de *Google*, se han utilizado *Google Actions Console* y *Dialogflow*.



The screenshot shows the Google Actions Console Overview page. At the top, there are tabs for 'Actions Console' (with a Google Assistant icon), 'Overview' (which is selected and highlighted in blue), 'Develop', 'Test', 'Deploy', and 'Analytics'. There's also a 'Juego inicial' dropdown and some notification icons. The main area is titled 'Spanish' and includes sections for 'Quick setup' (with a note: 'You've finished all the steps. Well done!'), 'Build your Action' (with a note: 'You have finished building Actions. Good job!'), and 'Get ready for deployment' (with notes: 'Before you create a release, let's check if you have all the information ready.', 'Enter information required for the Actions directory', and 'Select the countries your Actions will be deployed to').

Figura 7.4: Prototipo inicial en *Google*.

Además, también se ha probado *Voxa*, una plataforma que permite desarrollar aplicaciones para varias tecnologías a la vez. Sin embargo, finalmente hemos optado por profundizar en las plataformas propias de cada asistente.

Este prototipo inicial implementa 4 *slots* o *entities*:

- **Orientation:** alude a la orientación de la habitación.

The screenshot shows the configuration interface for the 'Orientation' slot. At the top, there is a header with three horizontal bars on the left, the title 'Orientation' in the center, and a blue 'SAVE' button with a three-dot menu icon on the right. Below the header are four configuration options with checkboxes: 'Define synonyms' (checked), 'Regexp entity' (unchecked), 'Allow automated expansion' (unchecked), and 'Fuzzy matching' (unchecked). A large table below lists orientation terms and their synonyms. The table has two columns: the first column contains the terms 'arriba', 'abajo', 'oeste', 'norte', 'sur', and 'este'; the second column contains their corresponding synonyms: 'arriba, techo', 'abajo, suelo', 'oeste, izquierda', 'norte, delante, adelante', 'sur, atrás, detrás', and 'este, derecha'. At the bottom of the table, there is a link 'Click here to edit entry'.

arriba	arriba, techo
abajo	abajo, suelo
oeste	oeste, izquierda
norte	norte, delante, adelante
sur	sur, atrás, detrás
este	este, derecha

Figura 7.5: Slot de tipo *Orientation*.

- **Objects:** se trata de aquellos objetos que el jugador va encontrando por la habitación y puede guardar en el inventario.

The screenshot shows the configuration interface for the 'Objects' slot. It has a similar layout to Figure 7.5, with a header, configuration options, and a table of objects and their synonyms. The header includes three horizontal bars, the title 'Objects', a blue 'SAVE' button, and a three-dot menu icon. The configuration options are: 'Define synonyms' (checked), 'Regexp entity' (unchecked), 'Allow automated expansion' (unchecked), and 'Fuzzy matching' (unchecked). The table lists objects and their synonyms, with two columns per row. The objects listed are 'silla', 'llave', 'comida de gato', 'linterna', and 'martillo'. Their corresponding synonyms are 'silla', 'llave', 'comida de gato, comida', 'linterna', and 'martillo'. A 'Click here to edit entry' link is at the bottom of the table.

silla	silla
llave	llave
comida de gato	comida de gato, comida
linterna	linterna
martillo	martillo

Figura 7.6: Slot de tipo *Objects*.

- **Still Objects:** son los objetos que se encuentran en la habitación y que pueden ser examinados para obtener más información.

StillObjects		SAVE	⋮
ventana	ventana		
cuadro	cuadro		
interruptor	interruptor		
lámpara	lámpara		
trampilla	trampilla		
puerta	puerta		
estantería	estantería, estantes		
armario	armario		
almohada	almohada		
cama	cama		
cajón	cajón		
escritorio	escritorio, mesa		
Click here to edit entry			

Figura 7.7: *Slot* de tipo *Orientation*.

- **Choose:** hace referencia a las posibles elecciones cuando el jugador encuentra la comida de gato.

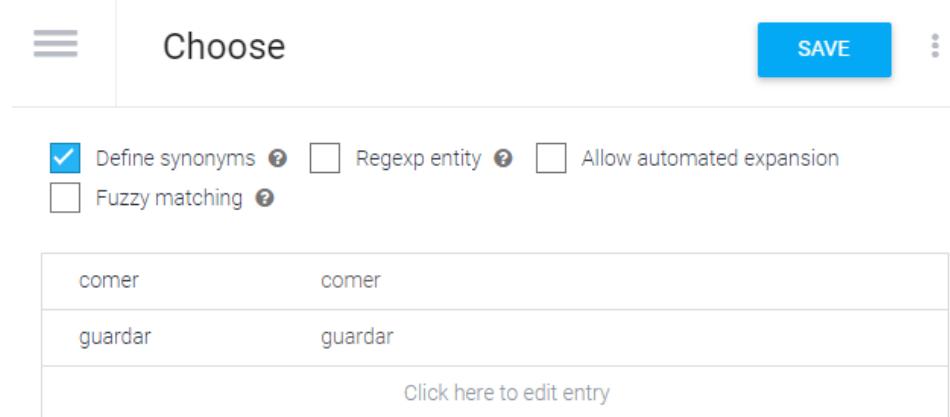


Figura 7.8: Slot de tipo *Choose*.

Además, se han utilizado 19 *intents* en *Alexa* y 17 en *Google Assistant*. Esto se debe a que *Alexa* incorpora algunos *intents* por defecto, como los relacionados con cancelar o parar el juego. Esto provoca que el total sea mayor que en *Google*. Los *intents* principales son:

- **Look:** consiste en el comando asociado a mirar en cualquier dirección de la habitación. Utiliza el *slot Orientation*.



Figura 7.9: *Look Intent*.

- **Interaction:** se trata de la interacción del usuario con los objetos de la habitación. Utiliza el *slot Still Objects*.

[Intents](#) / InteractionIntent

Sample Utterances (9) [?](#)

[Bulk Edit](#) [Export](#)

What might a user say to invoke this intent?	+
Ver {StillObjects}	✖
Inspecciona {StillObjects}	✖
Inspeccionar {StillObjects}	✖
Interaccionar con {StillObjects}	✖
Interacciona con {StillObjects}	✖

[1 – 5 of 9](#) [Show All](#)

Figura 7.10: *Interaction Intent.*

- **Default Interaction:** es la interacción por defecto. Esto quiere decir que si el usuario intenta examinar un objeto que no se encuentra en la lista del slot *Still Objects*, se activa este *intent*. Así, se puede dar una respuesta diferente al jugador, diciendo que el objeto que desea examinar no está disponible.

[Intents](#) / DefaultInteractionIntent

Sample Utterances (9) [?](#)

[Bulk Edit](#) [Export](#)

What might a user say to invoke this intent?	+
Examinar {query}	✖
Examina {query}	✖
Ver {query}	✖
Interaccionar con {query}	✖
Interacciona con {query}	✖

[1 – 5 of 9](#) [Show All](#)

Figura 7.11: *Default Interaction Intent.*

- **New Game:** permite iniciar una nueva partida. Almacena el nombre del jugador en un slot que viene predefinido en Alexa (*AMAZON.SearchQuery*) y Google Assistant (*@sys.any*). Dicho slot se utiliza para guardar cualquier tipo de palabra. Por ello, es adecuado para registrar el nombre del jugador, ya que no conocemos todos los posibles resultados.

[Intents / NewGameIntent](#)

Sample Utterances (8) [?](#)

[Bulk Edit](#) [Export](#)

What might a user say to invoke this intent?	+
empezar nuevo juego	✖
comenzar nuevo juego	✖
empezar juego	✖
comenzar juego	✖
juego nuevo	✖

◀ 1 – 5 of 8 ▶ [Show All](#)

Figura 7.12: *New Game Intent.*

- **Change Name:** su función es cambiar el nombre del jugador.

[Intents / ChangeNameIntent](#)

Sample Utterances (3) [?](#)

[Bulk Edit](#) [Export](#)

What might a user say to invoke this intent?	+
quiero modificar mi nombre	✖
cambio de nombre	✖
Quiero cambiar mi nombre	✖

◀ 1 – 3 of 3 ▶

Figura 7.13: *Change Name Intent.*

- **Tell Name:** permite que el usuario conozca su nombre.

[Intents / TellNameIntent](#)

Sample Utterances (3) [?](#)

[Bulk Edit](#) [Export](#)

What might a user say to invoke this intent?	+
Cuál es mi nombre	✖
Dime mi nombre	✖
Cómo me llamo	✖

◀ 1 – 3 of 3 ▶

Figura 7.14: *Tell Name Intent.*

- **Take:** se trata de los comandos usados para coger un objeto. Utiliza el *slot Objects*.

Intents / TakeIntent

Sample Utterances (2) ①

Bulk Edit Export

+  
Coger {object} Delete  
Coge {object} Delete

< 1 – 2 of 2 >

Figura 7.15: *Take Intent*.

- **Release:** permite soltar un objeto. También utiliza el *slot Objects*.

Intents / ReleaseIntent

Sample Utterances (2) ①

Bulk Edit Export

+  
Soltar {object} Delete  
Suelta {object} Delete

< 1 – 2 of 2 >

Figura 7.16: *Release Intent*.

- **Inventory:** su función es que el jugador conozca lo que lleva en el inventario.

The screenshot shows the configuration for the 'InventoryIntent' intent. At the top, there is a header with the intent name and a link to 'Sample Utterances (12)'. Below the header is a search bar containing the placeholder 'What might a user say to invoke this intent?'. A '+' button is located to the right of the search bar. The main area displays a list of 12 sample utterances, each with a delete icon to its right:

- Qué tengo en el inventario
- Qué hay en el inventario
- Qué tengo en la mochila
- Dime lo que hay en la mochila
- Qué hay en la mochila

At the bottom of the list, there are navigation links: '< 1 – 5 of 12 >' and 'Show All'.

Figura 7.17: *Inventory Intent*.

- **Use:** hace referencia a la acción de usar un objeto en otro dentro de la habitación. Esto es, por ejemplo, cuando el jugador quiere abrir la puerta. En este caso, puede decir: usar llave en puerta. De este modo, utiliza el slot *Objects* y el slot *Still Objects*.

The screenshot shows the configuration for the 'UseIntent' intent. At the top, there is a header with the intent name and a link to 'Sample Utterances (8)'. Below the header is a search bar containing the placeholder 'What might a user say to invoke this intent?'. A '+' button is located to the right of the search bar. The main area displays a list of 8 sample utterances, each with a delete icon to its right:

- Usar {object} con {element}
- Usa {object} con {element}
- Uso {object} en {element}
- Usar {object} en {element}
- Utilizar {object} con {element}

At the bottom of the list, there are navigation links: '< 1 – 5 of 8 >' and 'Show All'.

Figura 7.18: *Use Intent*.

- **Where:** indica al jugador dónde está.

[Intents / WhereIntent](#)

Sample Utterances (7) [?](#)

Bulk Edit Export

What might a user say to invoke this intent?	+
dónde me encuentro	trash
dónde estoy	trash
en qué habitación estoy	trash
en qué lugar estoy	trash
en qué lugar me encuentro	trash

1 – 5 of 7 [Show All](#)

Figura 7.19: *Where Intent*.

- **Tell Score:** informa al jugador acerca de su puntuación.

[Intents / TellScoreIntent](#)

Sample Utterances (3) [?](#)

Bulk Edit Export

What might a user say to invoke this intent?	+
cuántos puntos llevo	trash
Cuál es mi puntuación	trash
Dime mi puntuación	trash

1 – 3 of 3

Figura 7.20: *Tell Score Intent*.

- **Choose:** se utiliza para que el usuario escoja entre varias opciones posibles, marcadas por el slot *Choose*. En este caso, puede elegir comer o guardar la comida de gato una vez que la encuentra.

[Intents / ChooseIntent](#)

Sample Utterances (1) [?](#)

Bulk Edit Export

What might a user say to invoke this intent?	+
{choose}	trash

1 – 1 of 1

Figura 7.21: *Choose Intent*.

- **Clue:** da una pista al jugador.



Figura 7.22: *Clue Intent*.

- **Help:** da información acerca de los controles necesarios para jugar. Este *intent* está preconstruido en *Alexa* y *Google Assistant*, con lo que no han tenido que añadirse *utterances*. Más adelante, se ha modificado por un *intent* personalizado, para que el usuario pueda elegir entre una ayuda corta y una larga.

Una vez añadidos los *intents*, es necesario gestionar sus respuestas. Para ello, hemos escrito un código que desarrolla las reglas y la lógica del juego. En el caso de *Alexa*, disponemos de varios archivos.



Figura 7.23: Archivos de código del juego inicial en *Alexa*.

El archivo *index.js* contiene los *handler* de cada uno de los *intents* definidos. Una vez que un usuario activa un *intent* al utilizar un comando, se activará el código asociado al *handler* o manejador de dicho *intent*.

Por otro lado, los archivos *logic-es.js* y *room-es.js* contienen el código relacionado con la lógica en español del prototipo. De forma análoga, *logic-en.js* y *room-en.js* albergan la lógica en inglés. Además, *text.js* contiene algunas frases usadas durante el desarrollo del juego, como la bienvenida o la despedida. Esto crea una mayor abstracción, separando este código del archivo principal (*index.js*), desde donde serán llamadas.

En el caso de *Google Assistant*, tan solo disponemos de un archivo desarrollado en el editor en línea. En él, se incluyen tanto los manejadores como la lógica de la aplicación.

Para entender mejor cómo funciona el código, necesitamos conocer las principales variables usadas.

```
1 // Habitación
2 let room = '';
3
4 // Inventario del jugador
5 let inventory = [];
6
7 // Jugador
8 let points = '';
9 let name = '';
10
11 // Boolean para saber si el jugador ha muerto
12 let dead;
13
14 // Boolean para saber si el juego ha finalizado
15 let finished;
```

Estas variables guardarán los datos del jugador mientras se desarrolla la partida. Además, la habitación también utiliza algunas variables importantes.

```
1 // Objetos en la habitación
2 let objectsRoom = [];
3
4 // Objetos en el suelo de la habitación
5 let floorObjectsRoom = [];
6
7 // Array que almacena los objetos con los que se ha
     interaccionado
8 let roomElements = [];
9
10 // Array que indica si un objeto puede ser cogido
11 let canTakeObjects = new Map();
12
13 // Variable que indica si hay suficiente luz para explorar
14 let light;
```

Cuando se empieza un nuevo juego, las variables son inicializadas. En la lógica en español sería:

```

1   light = false;
2   inventory = [];
3
4   // Jugador
5   points = 0;
6   name = '';
7   dead = false;
8
9   // Objetos en la habitación
10  objectsRoom = ['llave', 'martillo', 'comida de gato', 'silla'];
11
12  // Objetos en el suelo de la habitación
13  floorObjectsRoom = ['silla'];
14
15  // Array que indica si un objeto puede ser cogido
16  canTakeObjects.set('llave', false);
17  canTakeObjects.set('martillo', false);
18  canTakeObjects.set('comida de gato', false);
19  canTakeObjects.set('silla', true);
20
21  roomElements = [];
22  finished = false;
23  room = 'habitación';

```

Además, se actualiza la variable *name* con el nombre escogido por el jugador. Una vez iniciado el juego, cada *intent* usado activará su manejador correspondiente y, con ello, los valores de las variables serán modificados en consecuencia. Por ejemplo, si queremos mirar al norte de la habitación, se activará *Look Intent*. El código de la función para *Google Assistant* sería:

```

1 app.intent(LOOK, (conv) =>{
2   const orientation = conv.parameters[ORIENTATION_ENTITY].toLowerCase();
3   let speechText = 'No puedes mirar nada porque no estás en
4   ninguna habitación.';
5   if(light){
6     switch(orientation){
7       case 'norte':
8         if(room === 'habitación'){
9           speechText = 'Al norte está el interruptor
10          que has pulsado y la puerta de la
11          habitación.';
12       }
13       break;
14       case 'sur':
15         if(room === 'habitación'){
16           speechText = 'Al sur puedes ver un
17           ';
18       }
19     }
20   }
21 }
22 
```

```

14             escritorio y una ventana .';
15         }
16     break;
17 case 'este':
18     if(room === 'habitación'){
19         speechText = 'Al este se encuentra la cama.
20             En ella hay una almohada algo estropeada
21             .';
22     }
23     break;
24 case 'oeste':
25     if(room === 'habitación'){
26         speechText = 'Al oeste hay un armario con
27             estantes y un cuadro muy bonito .';
28     }
29     break;
30 case 'arriba':
31     if(room === 'habitación'){
32         speechText = 'Arriba está la lámpara que
33             ilumina la habitación .';
34     }
35     break;
36 case 'abajo':
37     if(room === 'habitación'){
38         if(floorObjectsRoom.length === 0){
39             speechText = 'Abajo solo está el suelo.
40                 No hay nada interesante .';
41         } else{
42             speechText = 'En el suelo encuentras: '
43                 + floorObjectsRoom;
44         }
45     }
46     conv.ask(speechText);
47 });

```

Como el comando era *mirar norte*, este será el valor del *slot* representado por la variable *orientation*. Así, si la habitación tiene luz, el asistente nos responderá la frase asociada a la opción 'norte':

*Al norte está el interruptor que has pulsado y la puerta de la habitación.*

Del mismo modo, cada *handler* gestionará la respuesta que debe dar el

asistente y cambiará el valor de las variables según las acciones realizadas por el jugador.

### 7.3.2. Entrega 1

La entrega 1 desarrolla una versión del juego más extendida. En ella, se amplía el prototipo inicial de la aventura de forma estática.

El juego pasa de tener una habitación a tener siete. Por tanto, el objetivo se convierte en escapar de la casa en lugar de salir de una única habitación. El mapa de la casa quedaría como se muestra en la imagen.

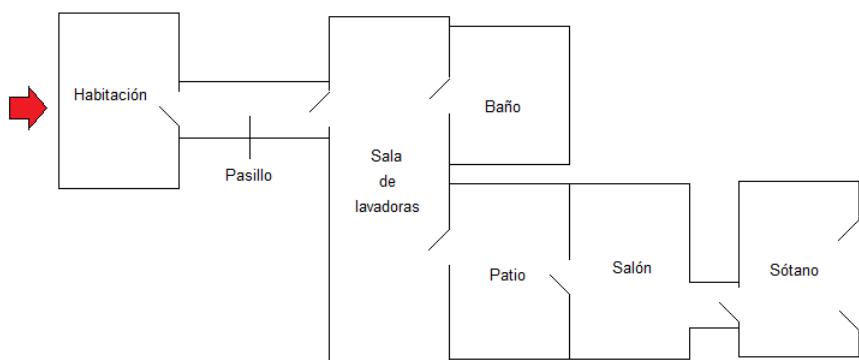


Figura 7.24: Mapa de la casa en la entrega 1.

En esta entrega, ya se incorpora la trama relacionada con los gatos. Además, suceden casi todos los eventos mencionados en el punto 7.2.2 asociados con las habitaciones añadidas.

Se trata de una versión funcional de la aventura en la que solo se incluye una historia principal. Dicha historia se basa en la resolución de retos. El jugador debe entregarle cuatro objetos al gato de forma ordenada para conseguir escapar.

Esta fase incorpora la creación de 4 nuevos *slots*.

- **Places:** son las distintas habitaciones de la casa.

Places		SAVE	⋮
<input checked="" type="checkbox"/> Define synonyms <small>?</small>	<input type="checkbox"/> Regexp entity <small>?</small>	<input type="checkbox"/> Allow automated expansion	
<input type="checkbox"/> Fuzzy matching <small>?</small>			
sótano	sótano, garaje, aparcamiento		
salón	salón, cuarto de estar		
patio	patio, patio interior		
baño	baño, cuarto de baño, aseo, servicio		
sala de lavadoras	sala de lavadoras, lavandería		
pasillo	pasillo, corredor		
habitación	habitación, cuarto, dormitorio		

Figura 7.25: Slot de tipo *Places*.

- **Symbol:** se trata de cuatro símbolos que ayudan a la resolución de uno de los retos.

Symbol		SAVE	⋮
<input checked="" type="checkbox"/> Define synonyms <small>?</small>	<input type="checkbox"/> Regexp entity <small>?</small>	<input type="checkbox"/> Allow automated expansion	
<input type="checkbox"/> Fuzzy matching <small>?</small>			
fuego	fuego		
aire	aire		
agua	agua		
tierra	tierra		
Click here to edit entry			

Figura 7.26: Slot de tipo *Symbol*.

- **Author:** se corresponde con varios escritores y poetas que son necesarios en uno de los desafíos.

**Author**

SAVE

Jorge Manrique	Jorge Manrique, Manrique
Antonio Machado	Antonio Machado, Machado
Calderón de la Barca	Calderón de la Barca
Lorca	Lorca, Federico García Lorca
Click here to edit entry	

Figura 7.27: Slot de tipo Author.

- **CatObjects:** son los objetos del gato. Estos objetos son esenciales cuando se llega al final del juego.

**CatObjects**

SAVE

comida de gato	comida de gato, comida
cuenco	cuenco
ovillo de lana	ovillo de lana, ovillo, lana
ratón de juguete	ratón de juguete, ratón, rata
Click here to edit entry	

Figura 7.28: Slot de tipo Cat Objects.

Además, se han añadido nuevos valores a los *slots* previos, como *Objects* o *Still Objects*. También se ha usado un *slot* preconstruido consistente en cuatro dígitos numéricos.

Con respecto a los *intents*, la adición de tantas habitaciones y retos nuevos ha supuesto su aumento de forma significativa. Algunos de los que se han incorporado en esta entrega son:

- **Go To:** permite que el usuario se desplace de una habitación a otra. Incluye el slot *Places*.

Intents / GoToIntent

Sample Utterances (19) [?](#)

Bulk Edit Export

What might a user say to invoke this intent? [+](#)

ve a la {place}	<a href="#">Delete</a>
desplázate a la {place}	<a href="#">Delete</a>
muévete a la {place}	<a href="#">Delete</a>
desplazarse a la {place}	<a href="#">Delete</a>
moverse a la {place}	<a href="#">Delete</a>

◀ 1 – 5 of 19 ▶ Show All



Figura 7.29: *Go To Intent*.

- **Read:** se trata de un reto de lectura. Utiliza el slot *Author*.

Intents / ReadIntent

Sample Utterances (4) [?](#)

Bulk Edit Export

What might a user say to invoke this intent? [+](#)

Leer a {author}	<a href="#">Delete</a>
Lee a {author}	<a href="#">Delete</a>
Lee {author}	<a href="#">Delete</a>
Leer {author}	<a href="#">Delete</a>

◀ 1 – 4 of 4 ▶



Figura 7.30: *Read Intent*.

- **Push Symbols:** hace referencia a un reto asociado con unos símbolos. Usa el slot *Symbol*.

The screenshot shows the configuration for the `PushSymbolsIntent`. It includes a header with "Intents / PushSymbolsIntent", a "Sample Utterances (2)" section, and a "Bulk Edit" and "Export" button. Below are two utterance examples:

- What might a user say to invoke this intent? (with a plus icon)
- Pulsa {symbol} {symboltwo} {symbolthree} {symbolfour} (with a trash icon)
- Pulsar {symbol} {symboltwo} {symbolthree} {symbolfour} (with a trash icon)

At the bottom right, there is a "1 – 2 of 2" indicator.

Figura 7.31: *Push Symbols Intent.*

- **Number Code:** se trata de un desafío en el que debe introducirse un código de cuatro dígitos. Usa un *slot* preconstruido.

The screenshot shows the configuration for the `NumberCodeIntent`. It includes a header with "Intents / NumberCodeIntent", a "Sample Utterances (2)" section, and a "Bulk Edit" and "Export" button. Below are two utterance examples:

- What might a user say to invoke this intent? (with a plus icon)
- Introducir {number} (with a trash icon)
- Introduce {number} (with a trash icon)

At the bottom right, there is a "1 – 2 of 2" indicator.

Figura 7.32: *Number Code Intent.*

- **Combine Objects:** permite combinar dos objetos del tipo *Objects*.

The screenshot shows the configuration for the `CombineObjectsIntent`. It includes a header with "Intents / CombineObjectsIntent", a "Sample Utterances (3)" section, and a "Bulk Edit" and "Export" button. Below are three utterance examples:

- What might a user say to invoke this intent? (with a plus icon)
- Combinar {item} y {itemsegundo} (with a trash icon)
- Usar {item} con {itemsegundo} (with a trash icon)
- Usar {item} en {itemsegundo} (with a trash icon)

At the bottom right, there is a "1 – 3 of 3" indicator.

Figura 7.33: *Combine Objects Intent.*

- **Use Object:** se trata de un *intent* asociado al uso de un objeto del tipo *Object*.

[Intents](#) / UseObjectIntent

Sample Utterances (4) 

 Bulk Edit  Export

What might a user say to invoke this intent?	
Utilizar {object}	
Utiliza {object}	
Usa {object}	
Usar {object}	

◀ 1 – 4 of 4 ▶

Figura 7.34: Use Object Intent.

- **Give Cats:** está asociado con el final del juego, en el que deben dársele varios objetos al gato. Usa *Cat Objects*.

[Intents](#) / GiveCatIntent

Sample Utterances (1) 

 Bulk Edit  Export

What might a user say to invoke this intent?	
Dar {elementone} {elementtwo} {elementthree} {elementfour}	

◀ 1 – 1 of 1 ▶

Figura 7.35: Give Cats Intent.

En referencia al código, se han añadido nuevos *handlers*. Además, cada habitación incorpora sus propias variables, de forma similar a las variables de la habitación inicial en la entrega 0. Por ejemplo, en el caso del pasillo:

```

1 // Objetos en el pasillo
2 let objectsCorridor = [];
3
4 // Objetos en el suelo del pasillo
5 let floorObjectsCorridor = [];
6
7 // Array con interacciones de los elementos del pasillo
8 let corridorElements = [];
9
10 // Array que indica si un objeto se puede coger
11 let canTakeObjects;
12
13 // Variable que indica si hay suficiente luz para explorar
14 let light;
```

### 7.3.3. Entrega 2

Esta entrega incorpora una funcionalidad algo más dinámica. Para ello, se ha organizado el código asociado al juego de forma que cada habitación está en un archivo diferente. Esto supone un mayor nivel de abstracción, haciendo más fácil añadir o quitar alguna estancia a la hora de modificar la aventura. De este modo, hay un archivo principal en el que se gestiona la lógica del juego, a partir del cual se accede a cada una de las habitaciones.

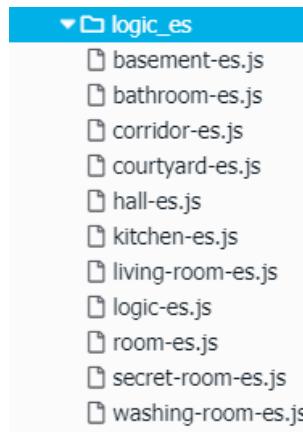


Figura 7.36: Archivos de las habitaciones en *Alexa*.

Además, se ha añadido una trama alternativa a la historia principal. Así, las acciones del jugador determinarán la aventura a la que se enfrenta. Por último, se han incorporado más elementos de toma de decisiones. Esto hace que el usuario tenga que escoger un camino, como ocurre en los librojuegos.

## Historia principal

Los elementos de toma de decisiones se basan en los objetos relacionados con el gato. Cada vez que encontramos uno de ellos, se nos ofrecen dos alternativas. A continuación se detallan dichos elementos dentro de la trama principal:

1. **Comida de gato:** podemos cogerla o comérnosla.



Figura 7.37: Comida de gato.

2. **Ovillo de lana:** podemos guardarla o jugar con él.



Figura 7.38: Ovillo de lana.

3. **Ratón de juguete:** podemos cogerlo o apretarlo.



Figura 7.39: Ratón de juguete.

4. **Cuenco:** podemos guardarlo o lanzarlo.

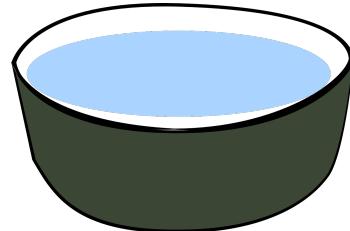


Figura 7.40: Cuenco.

En todos los casos, coger el objeto nos da diez puntos, mientras que escoger la otra alternativa hace que el jugador pierda diez puntos. Las modificaciones que permiten esta toma de decisiones se ven reflejadas en el *slot Choose* y el *intent Choose Object*.

Otro de los cambios afecta al final del juego. Ya no es necesario entregar cuatro objetos al gato en un cierto orden. Pueden entregársele menos objetos provocando distintos resultados, tal y como se muestra en el siguiente diagrama. Esto ha supuesto que el *intent Cat Objects* sea modificado.

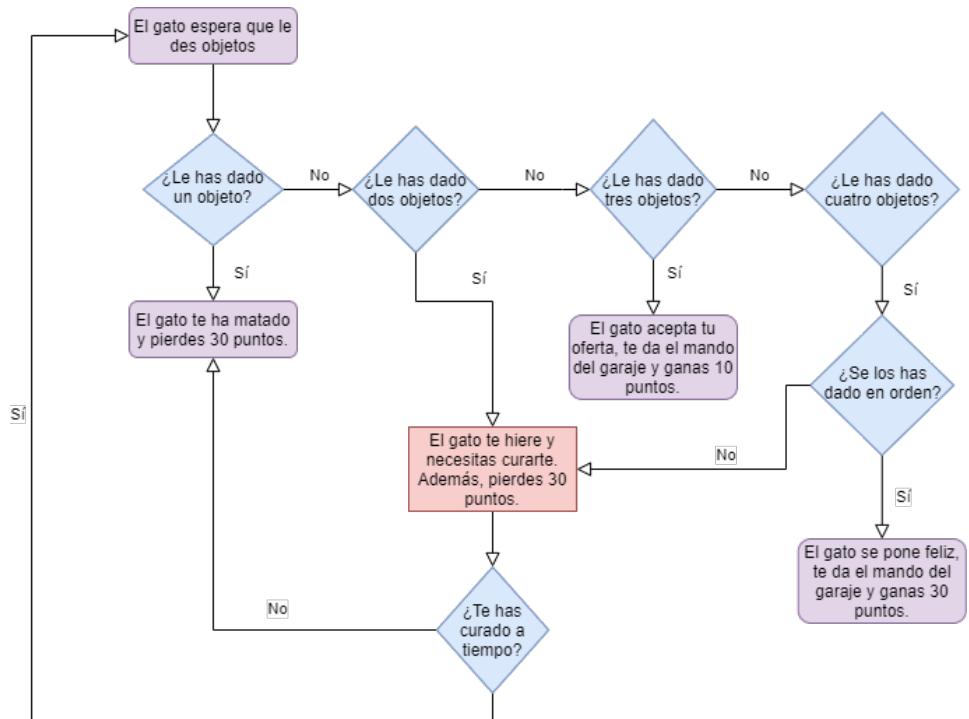


Figura 7.41: Diagrama de flujo al darle elementos al gato.

Este diagrama se basa en la suposición de que los objetos que se le entregan al gato están en el inventario del jugador y no se repiten.

### Historia alternativa

La historia alternativa lleva al jugador a las habitaciones que no aparecen en la entrega 1. En el caso de que decida examinar una pared cuando se encuentra en la sala de lavadoras, aparecerá en la cocina y accederá a esta trama. Al contrario que la historia principal, donde la resolución de retos era lo principal, esta historia se basa más en la narrativa.

El final de esta trama también tiene relación con un gato. No obstante, en este caso se trata de un gato fantasma.

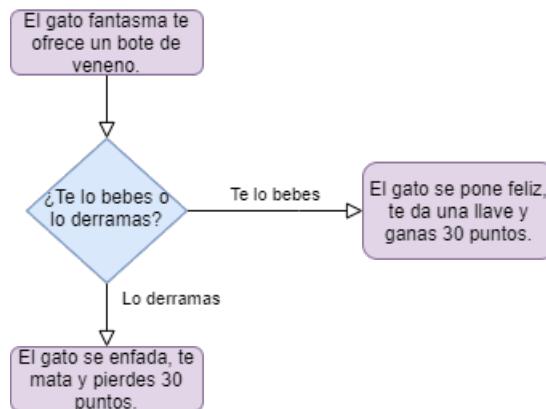


Figura 7.42: Diagrama de flujo al encontrarte con el gato fantasma.

La decisión de beber o derramar el veneno también se incluye en el *intent Choose Object*. Además, forma parte de la toma de decisiones mencionada anteriormente.

**Veneno:** podemos bebérnoslo o derramarlo.



Figura 7.43: Veneno.

#### 7.3.4. Entrega 3

Esta entrega incluye los sonidos del juego. En el caso de *Alexa*, la mayoría de ellos han sido sacados de *Alexa Skills Kit Sound Library*. Para *Google Assistant*, se ha utilizado su biblioteca de sonidos, *Sound Library*.

Además, también se han añadido imágenes al juego para su versión en *Alexa*. Esto ha sido posible mediante el uso de *Alexa Presentation Language*. Dado que el dispositivo usado para desarrollar el proyecto es el *Amazon Echo Show 5*, hemos podido probar la componente visual mientras se implementaba.

Vamos a ver la funcionalidad incorporada a través de las imágenes. Una vez que se abre el juego, aparece una pantalla inicial con la imagen de una mansión.



Figura 7.44: Pantalla de introducción.

Si se inicia un nuevo juego, aparece la imagen de la habitación en la que despierta el personaje.



Figura 7.45: Pantalla de la habitación.

Este formato de imagen se repite con cada una de las habitaciones del juego. Conforme nos desplazamos por la casa, cambia la imagen de fondo y el título, indicándonos el nombre de la habitación. Además, hay dos botones que nos permiten ver el inventario u obtener una pista.

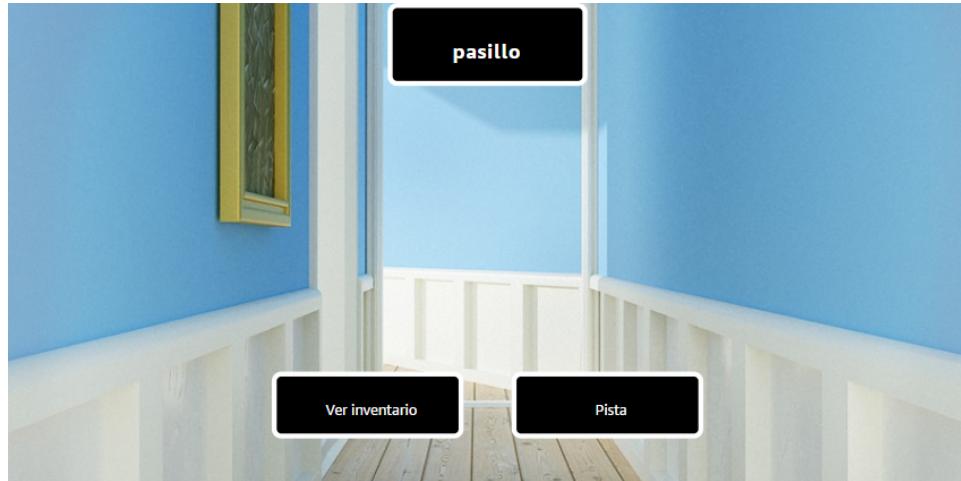


Figura 7.46: Pantalla del pasillo.



Figura 7.47: Pantalla de la sala de lavadoras.



Figura 7.48: Pantalla del baño.



Figura 7.49: Pantalla del patio.



Figura 7.50: Pantalla del salón.

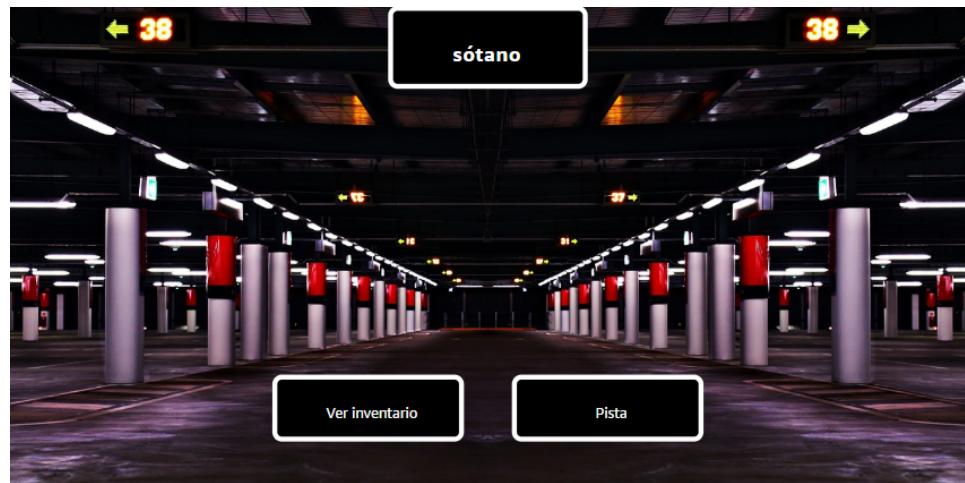


Figura 7.51: Pantalla del sótano.



Figura 7.52: Pantalla de la cocina.



Figura 7.53: Pantalla de la habitación secreta.

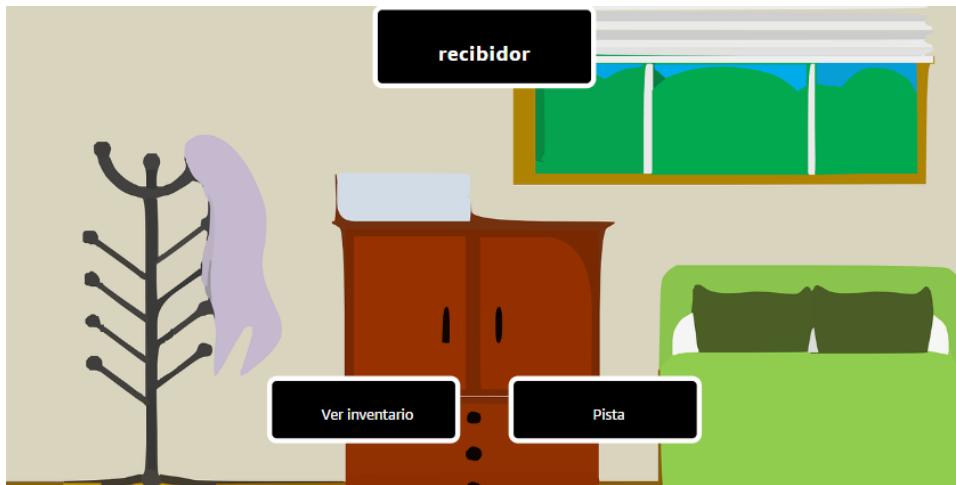


Figura 7.54: Pantalla del recibidor.

En las dos primeras habitaciones también se ha implementado otra funcionalidad que podría ser una futura mejora extensible a toda la aventura. Se trata de añadir distintas imágenes en función de la dirección en la que quiera mirar el jugador. De este modo, podrá ver qué objetos hay en cada orientación.

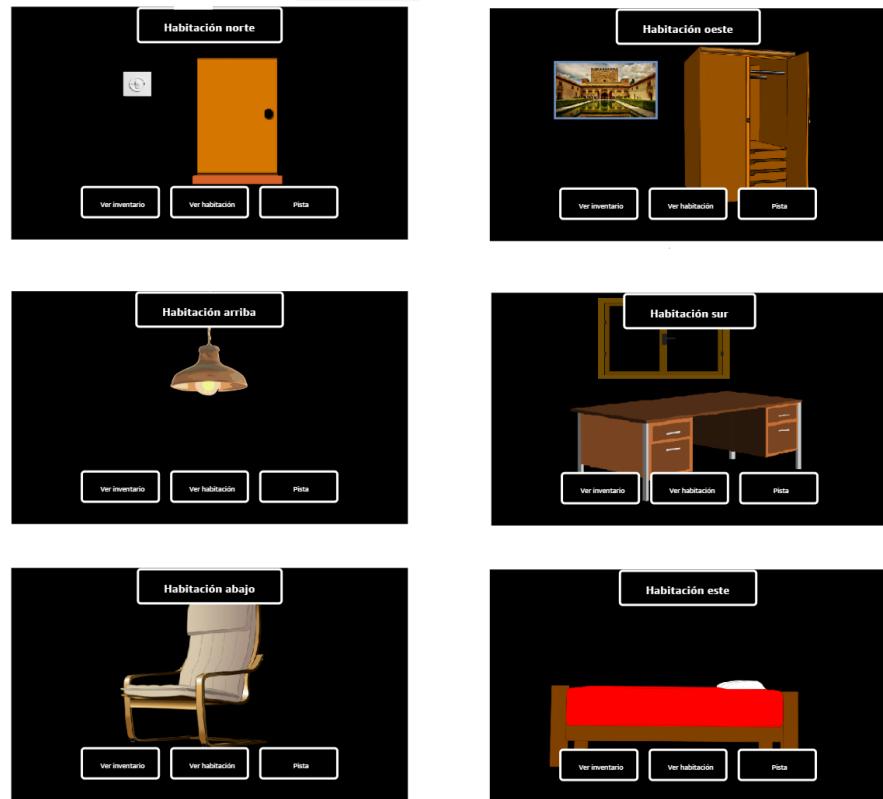


Figura 7.55: Imágenes en cada dirección de la habitación.

Si decidimos ver el inventario, se nos muestra una lista de los elementos que llevamos. Desde esta pantalla, se puede volver a la imagen de la habitación o recargar el inventario si se ha cogido o soltado algo nuevo.



Figura 7.56: Pantalla del inventario.

Por último, hay dos pantallas que aparecen cuando se acaba el juego. En el caso de que se gane, se muestra la siguiente imagen.

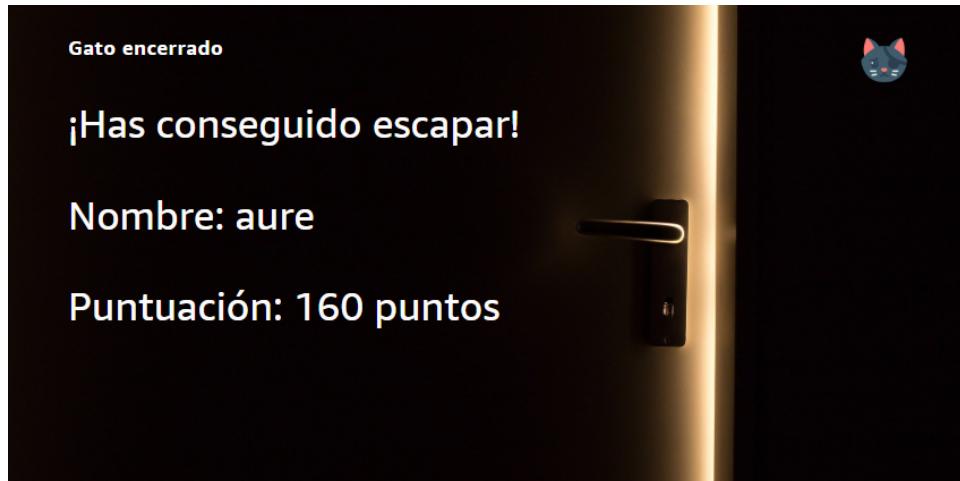


Figura 7.57: Pantalla del juego ganado.

Por otro lado, si se pierde aparece esta imagen.

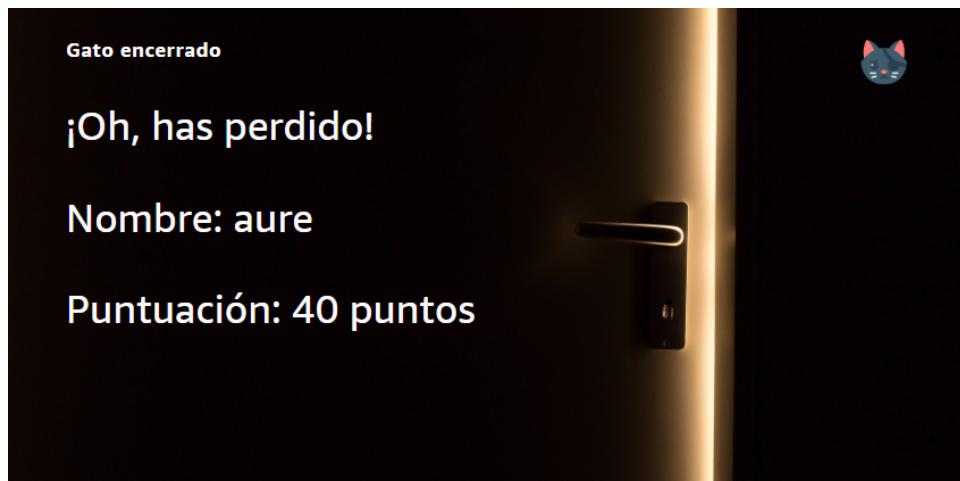


Figura 7.58: Pantalla del juego perdido.

### 7.3.5. Entrega 4

La entrega 4 añade la función de guardar la partida. Es una opción muy útil, ya que permite que el usuario pueda parar de jugar cuando quiera y continúe en otro momento por donde dejó su aventura.

Los datos se van guardando de forma automática. Dado que *Alexa* y *Google Assistant* cierran el juego cuando no reciben una interacción continua por parte del jugador, se trata de algo realmente necesario. De este modo, si el usuario está resolviendo un reto y tarda demasiado en responder, puede volver a abrir el juego y seguir avanzando sin tener que empezar de nuevo su partida.

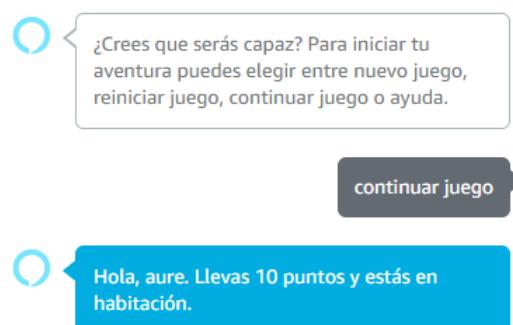


Figura 7.59: Continuar juego en *Alexa*.

Para lograr implementar esta función, hemos añadido variables que almacenan la información de la partida en una base de datos. Así, cuando el jugador opta por continuar la partida, los datos se recuperan. Esto se ha llevado a cabo mediante el *intent Continue Game*.

The screenshot shows the "Intents / ContinueGameIntent" section. It includes a "Sample Utterances (4)" list:

- continuar partida anterior
- seguir
- continuar
- continuar juego

At the bottom right, there are "Bulk Edit" and "Export" buttons, and a page indicator "1 - 4 of 4".

Figura 7.60: *Intent Continue Game*.

El código asociado a este *intent* en *Alexa* sería:

```
1 const ContinueGameIntentHandler = {
2     canHandle(handlerInput) {
```

```
3     return Alexa.getRequestType(handlerInput.requestEnvelope
4         ) === 'IntentRequest'
5     && Alexa.getIntentName(handlerInput.requestEnvelope)
6         === 'ContinueGameIntent';
7 },
8 handle(handlerInput) {
9     const { attributesManager } = handlerInput;
10    const requestAttributes = attributesManager.
11        getRequestAttributes();
12    const reprompt = requestAttributes.t('REPROMPT_MSG');
13    const title1 = requestAttributes.t('BUTTON_INVENTORY');
14    const title2 = requestAttributes.t('BUTTON_CLUE');
15
16    const sessionAttributes = attributesManager.
17        getSessionAttributes();
18    const language = handlerInput.requestEnvelope.request.
19        locale;
20
21    let name;
22    let points;
23    let room;
24    let speakOutput = '';
25    let end;
26    let url = '';
27
28    if (language === 'es-ES') {
29        name = sessionAttributes['nameEs'];
30        points = sessionAttributes['pointsEs'];
31        room = sessionAttributes['roomEs'];
32        end = sessionAttributes['deadEs'] ||
33            sessionAttributes['finishedEs'];
34    } else if (language === 'en-US') {
35        name = sessionAttributes['nameEn'];
36        points = sessionAttributes['pointsEn'];
37        room = sessionAttributes['roomEn'];
38        end = sessionAttributes['deadEn'] ||
39            sessionAttributes['finishedEn'];
40    }
41
42    if (end) {
43        speakOutput = requestAttributes.t('ENDED_GAME');
44    } else {
45        if (room && name) {
46            speakOutput = requestAttributes.t('CONTINUE_MSG'
47                , name, points, room);
48            if (language === 'es-ES') {
49                url = logic_es.continueGame(
50                    sessionAttributes);
51            } else if (language === 'en-US') {
52                url = logic_en.continueGame(
53                    sessionAttributes);
54            }
55            apl.getRoom(handlerInput, room, url, title1,
56                title2);
57        }
58    }
59}
```

```

46         } else {
47             speakOutput = requestAttributes.t('NO_GAME_SAVED
48                         ');
49         }
50     }
51     return handlerInput.responseBuilder
52         .speak(speakOutput)
53         .reprompt(reprompt)
54         .getResponse();
55     }
56 };

```

Los valores almacenados se recuperan de la base de datos y se asocian a *Session Attributes*. De este modo, se obtienen las principales variables relacionadas con el jugador. Esto permite saber si la partida anterior fue finalizada o si no hay ninguna partida empezada. En el caso de que sí se pueda continuar la partida, se llama a *continueGame*. Esta función guarda el resto de datos en variables locales para proseguir con la aventura. Además, en función de la habitación en la que se encuentra el usuario se actualiza la imagen de la pantalla.

En el caso de *Google Assistant*, se ha procedido de forma análoga. Se ha utilizado el almacenamiento proporcionado por *userStorage*.

### 7.3.6. Entrega 5

La entrega 5 añade una versión completa del juego en inglés, tanto para *Alexa* como para *Google Assistant*. Esto amplía el público al que puede llegar.

El código es básicamente idéntico al de la versión en español. No obstante, todos los textos y el nombre de los objetos y las habitaciones han sido traducidos y adaptados al inglés.

*Alexa* incluye ambos idiomas dentro del mismo proyecto de *Alexa Developer Console*. En el caso de *Google Assistant*, ha resultado más cómodo realizar dos proyectos individuales para cada uno de los idiomas en *Dialogflow* y *Google Actions Console*.

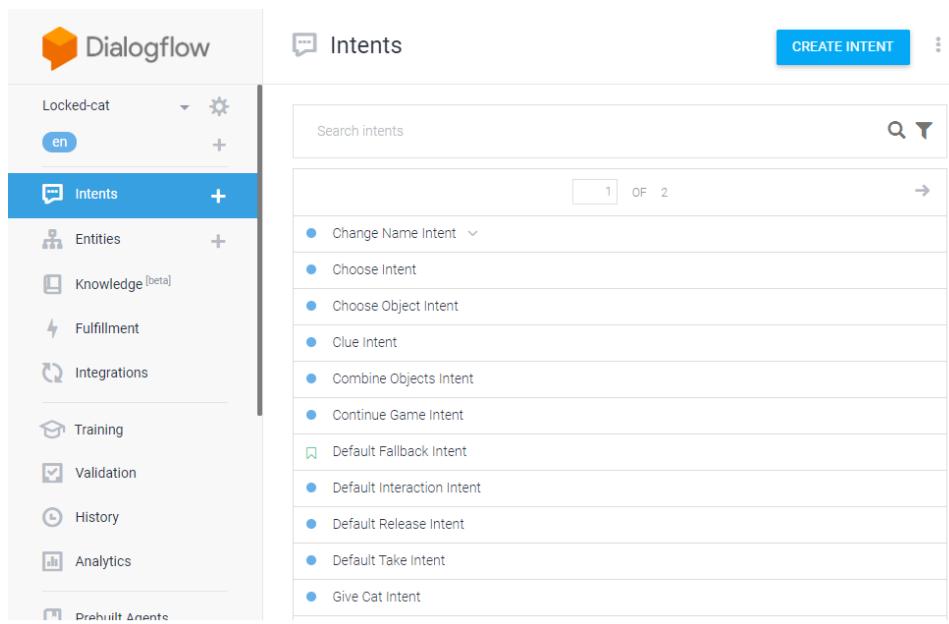


Figura 7.61: *Locked cat* en *Dialogflow*.

Además, en cada plataforma se invoca de manera diferente. En el caso de *Alexa*, el juego se llama *locked cat*.

*Alexa, open locked cat.*

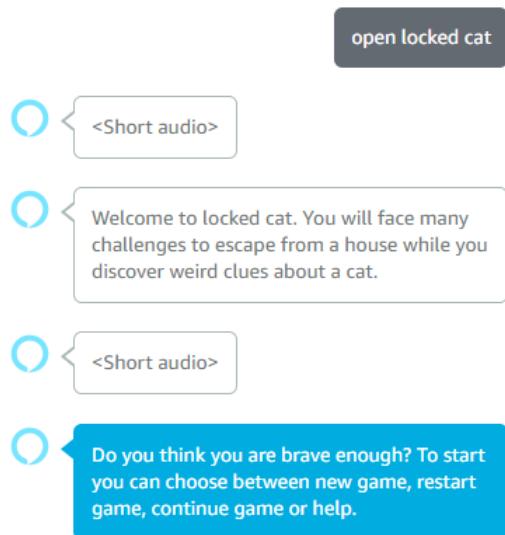


Figura 7.62: Juego en inglés en *Alexa*.

Sin embargo, en el caso de *Google Assistant*, su nombre es *trapped cat*.

*Ok, Google, talk to trapped cat.*

## 7.4. Análisis de usabilidad

Para evaluar los prototipos, se han probado los cambios implementados conforme se iban añadiendo. Además, se ha realizado un análisis de usabilidad con potenciales usuarios para conocer sus opiniones acerca del juego.

### 7.4.1. Encuesta a usuarios que no han probado el juego

Por un lado, hemos creado una encuesta que trata de recolectar la impresión de los usuarios que no han jugado al juego. Para ello, hemos añadido dos vídeos de demostración de la aventura. Una vez visionados, los usuarios pueden responder unas preguntas sobre su experiencia previa con asistentes virtuales y su opinión sobre la trama y los controles del juego.

Con respecto a esta encuesta, hemos obtenido 18 respuestas. El rango de edades de los participantes ha sido variado, siendo 22 la edad mínima y 69 la máxima.

Hemos empezado poniendo preguntas sobre la experiencia de los usuarios con los asistentes virtuales. En este caso, algo más de la mitad había usado algún asistente alguna vez.

¿Has usado alguna vez algún asistente virtual? (Asistente de Google, Alexa, Bixby, Siri...).

18 respuestas

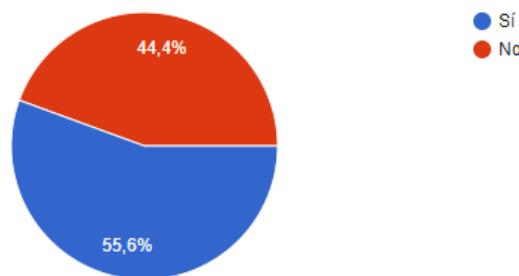


Figura 7.64: Pregunta 1 de la encuesta 1 de usabilidad.

En el caso de los usuarios que habían utilizado alguna vez un asistente

## TFG - Juego para un asistente conversacional

¡Hola! Estoy desarrollando un juego como trabajo fin de grado. Se trata de una aventura que combina la resolución de puzzles, la toma de decisiones y la narrativa. La interacción es principalmente oral. Estoy desarrollándolo para Alexa y el asistente de Google. En el caso de Alexa, también incorpora algunas imágenes si el dispositivo dispone de pantalla.

El juego se llama "Gato encerrado". Esta es la temática:

Despiertas en una habitación que se asemeja a la tuya. Sin embargo, comienzas a explorarla y descubres que las cosas no te cuadran. El objetivo es escapar. Para ello, debes interactuar con los objetos que vas encontrando y resolver retos que te permitan avanzar por la casa. Además, conforme avanzas en tu aventura van apareciendo elementos relacionados con los gatos, pero tú nunca has tenido uno...

Aquí puedes ver un par de vídeos que muestran un ejemplo de la funcionalidad del juego:

<https://drive.google.com/file/d/10sRB5NNywXkwjnsdkl-nqWn7FyZ0Er-P/view?usp=sharing>  
[https://drive.google.com/file/d/1\\_rJMyXRPIQNhQAmBrDTCxjNDHo\\_kRUj-/view?usp=sharing](https://drive.google.com/file/d/1_rJMyXRPIQNhQAmBrDTCxjNDHo_kRUj-/view?usp=sharing)

El juego completo es más largo e incluye varias tramas posibles.

Estoy buscando la opinión de los posibles usuarios. Para ello, puedes responder a las siguientes preguntas.

¡Gracias por tu colaboración! :)

Figura 7.63: Encuesta 1 de usabilidad dirigida a los usuarios que no han probado el juego.

virtual, todos ellos decían haber tenido una experiencia positiva.

Si has respondido que sí, ¿te ha gustado la experiencia?

11 respuestas

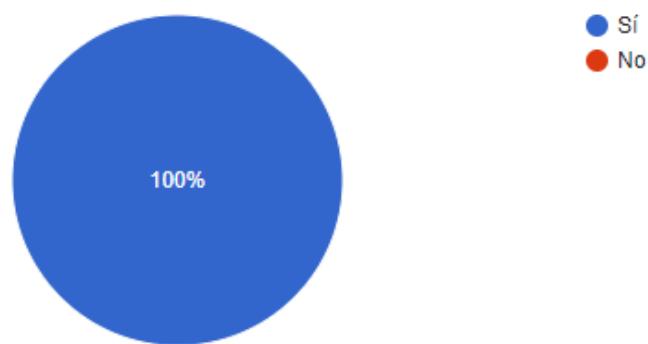


Figura 7.65: Pregunta 2 de la encuesta 1 de usabilidad.

Aquellos que no lo habían usado, lo consideraban interesante y querían probarlo alguna vez.

Si no has usado nunca un asistente virtual, ¿te gustaría probarlo alguna vez? ¿Lo consideras interesante?

12 respuestas

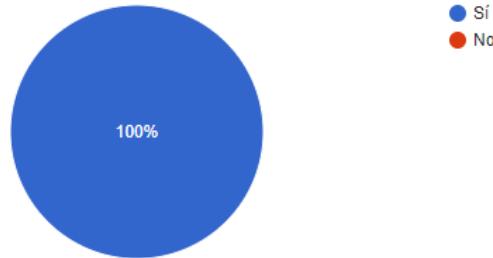


Figura 7.66: Pregunta 3 de la encuesta 1 de usabilidad.

Una vez vista la opinión y la práctica de los usuarios con los asistentes, hemos entrado en el ámbito de los juegos. Un 77.8% de los encuestados no había jugado nunca a ningún juego con un asistente.

¿Has jugado a algún juego con estos asistentes?

18 respuestas

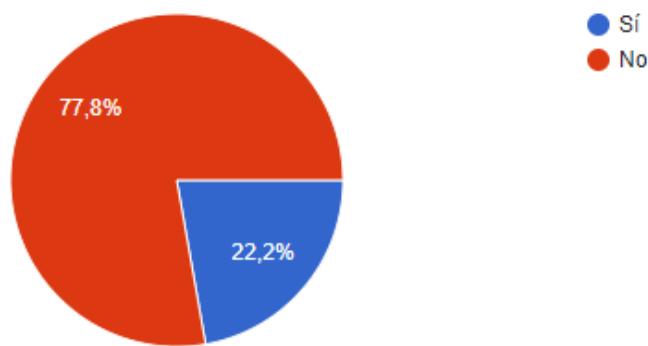


Figura 7.67: Pregunta 4 de la encuesta 1 de usabilidad.

De aquellos que sí habían probado algún juego, el 80 % respondía que le había gustado la experiencia.

Si has jugado, ¿te gustó?

5 respuestas

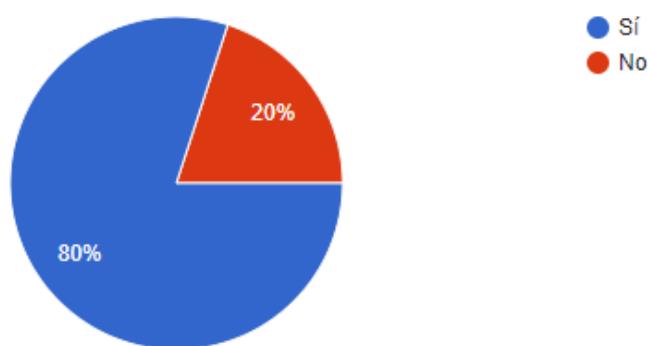


Figura 7.68: Pregunta 5 de la encuesta 1 de usabilidad.

Hemos continuado la encuesta con preguntas sobre el juego y los vídeos visionados. Todos los usuarios alegaban que la temática del juego les parecía interesante.

¿Te parecen interesantes la temática y la idea de este juego?

18 respuestas

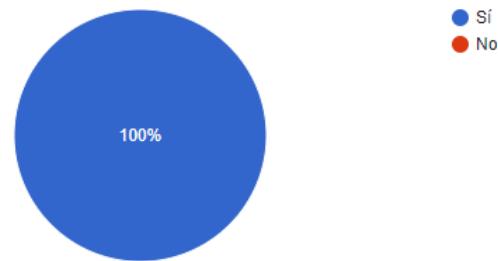


Figura 7.69: Pregunta 6 de la encuesta 1 de usabilidad.

Si no te interesan, ¿por qué?

0 respuestas

Aún no hay respuestas para esta pregunta.

Figura 7.70: Pregunta 7 de la encuesta 1 de usabilidad.

Con respecto a los controles, el 88.9 % opinaba que son fáciles de usar.

¿Crees que los controles del juego son fáciles de usar?

18 respuestas

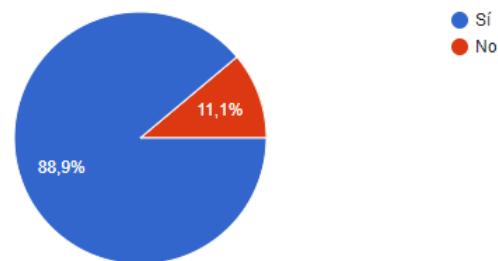


Figura 7.71: Pregunta 8 de la encuesta 1 de usabilidad.

Aquellos que contestaron que los controles no eran fáciles, argumentaban lo siguiente:

Si no te resultan fáciles los controles del juego, ¿por qué y qué cambiarías?

2 respuestas

El tema es que unx puede decir las cosas de muchas maneras entonces a lo mejor te vuelves un poco locx intentando encontrar las palabras que sirven. Es una idea.

Muchas instrucciones seguidas en la pantalla de Nuevo Juego, me costaría recordarlas todas aunque pidiera repetirlas porque son muchas seguidas. Pondría una opción para repetir partes concretas de las instrucciones o una imagen o imágenes en pantalla que las resuma, por si el jugador se despista.

Figura 7.72: Pregunta 9 de la encuesta 1 de usabilidad.

Para solventar este problema, se han incorporado al juego dos tipos de ayudas: una corta y una larga. De este modo, los usuarios pueden solicitar la que deseen. La ayuda corta resulta más fácil cuando se inicia el juego, dado que permite conocer los controles más básicos. Por otro lado, la larga aporta más información, y puede ser necesaria cuando el jugador ya se ha familiarizado un poco con la aventura.

La siguiente pregunta hacía referencia a la predisposición de los encuestados a probar el juego.

¿Sería un juego al que te parecería interesante jugar?

18 respuestas

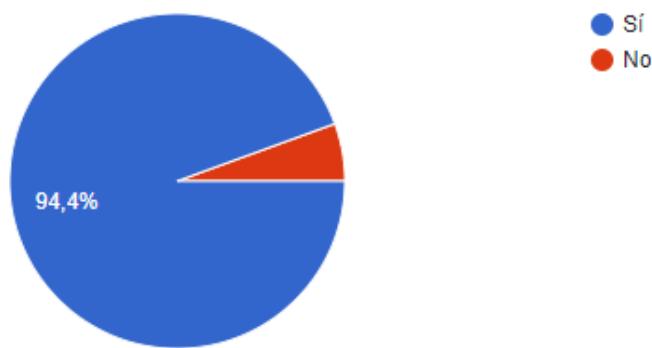


Figura 7.73: Pregunta 10 de la encuesta 1 de usabilidad.

Tan solo un usuario respondía que no, alegando que no le interesaba probar el juego.

Si tu respuesta es no, ¿por qué? ¿Se trata de un tema personal o tiene que ver con la funcionalidad o la temática del juego?

1 respuesta

El juego

Figura 7.74: Pregunta 11 de la encuesta 1 de usabilidad.

En último lugar, hemos preguntado a los usuarios qué añadirían al juego. Varias respuestas coinciden en un modo multijugador. Sin embargo, dadas las características de los asistentes virtuales, no sería algo posible por el momento. Otras réplicas sugieren incorporar más intrigas, un mapa de las habitaciones, animaciones o más imágenes según la dirección en la que mira el jugador.

¿Habria algo que te gustaría que el juego incluyera?

18 respuestas

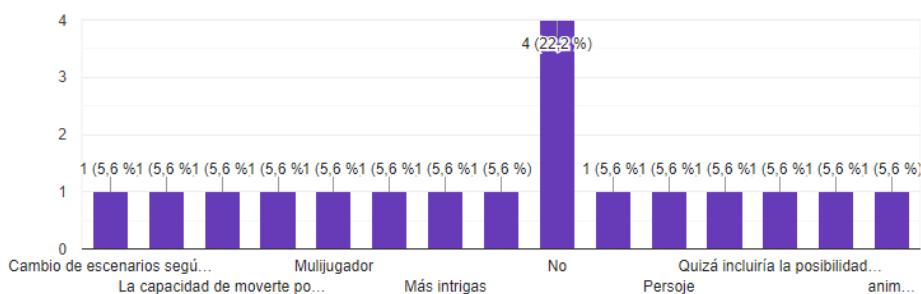


Figura 7.75: Pregunta 12 de la encuesta 1 de usabilidad. Sugerencias realizadas por los usuarios acerca de lo que añadirían al juego.

#### 7.4.2. Encuesta a usuarios que han probado el juego

Por otra parte, algunos usuarios han probado una versión casi completa del juego y han llenado otra encuesta en la que se ha valorado su experiencia con la aventura.

Las respuestas de este cuestionario son más reducidas, ya que solo 4 personas han probado el juego. Algunos de ellos lo han jugado en el asistente de *Google* disponible en los teléfonos móviles, mientras que otros lo han usado en *Alexa*, con el dispositivo *Echo Show 5*. De este modo, aquellos que han jugado en *Alexa* han podido probar la funcionalidad de las imágenes que aparecen por pantalla.

## TFG - Gato encerrado

Cuestionario que evalúa la experiencia del usuario una vez que ha probado el juego "Gato encerrado". Se trata de una aventura de interacción oral en la que debes escapar de una casa enfrentándote a numerosos retos y tomando decisiones por el camino.

¡Gracias por tu participación!

Figura 7.76: Encuesta 2 de usabilidad dirigida a los usuarios que han probado el juego.

La edad de los usuarios que han llenado esta encuesta está entre los 23 y los 29 años. Todos ellos coinciden en que el juego les ha gustado y los controles les han resultado fáciles de usar.

¿Te ha gustado el juego?

4 respuestas

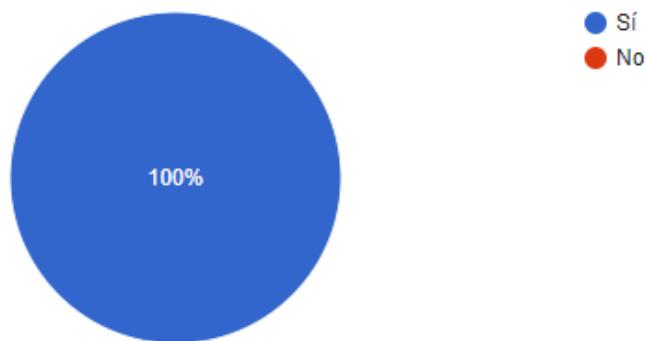


Figura 7.77: Pregunta 1 de la encuesta 2 de usabilidad.

¿Crees que los controles son fáciles de manejar? (En referencia a la interacción dentro del juego).

4 respuestas



Figura 7.78: Pregunta 2 de la encuesta 2 de usabilidad.

Ninguno cambiaría nada de los controles, pero uno de los usuarios hace referencia a que el asistente le resulta repetitivo. Esto se debe principalmente a que los asistentes requieren una interacción rápida y continua. De este modo, al no recibir una respuesta de forma inmediata pregunta al usuario constantemente.

¿Cambiarias algo con respecto a los controles?

4 respuestas

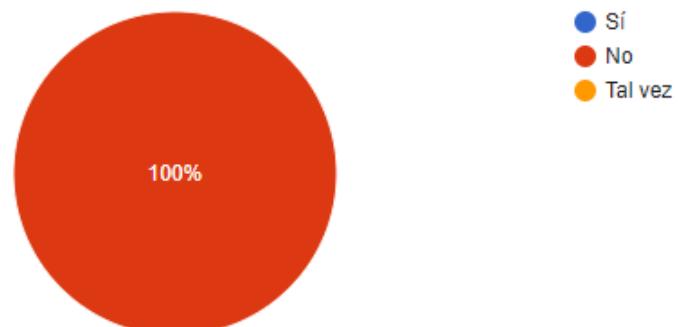


Figura 7.79: Pregunta 3 de la encuesta 2 de usabilidad.

Si es así, ¿qué cambiarías?

1 respuesta

A veces resulta repetitiva Alexa, pero entiendo que es la forma de hacerlo.

Figura 7.80: Pregunta 4 de la encuesta 2 de usabilidad.

Además, les ha parecido interesante la trama y han considerado adecuado el nivel de dificultad de los retos.

¿Qué te ha parecido el nivel de complejidad de los retos?

4 respuestas

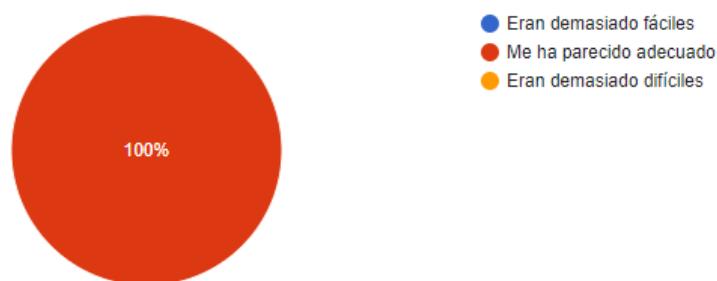


Figura 7.81: Pregunta 5 de la encuesta 2 de usabilidad.

¿Qué te ha parecido el nivel de complejidad de los retos?

4 respuestas

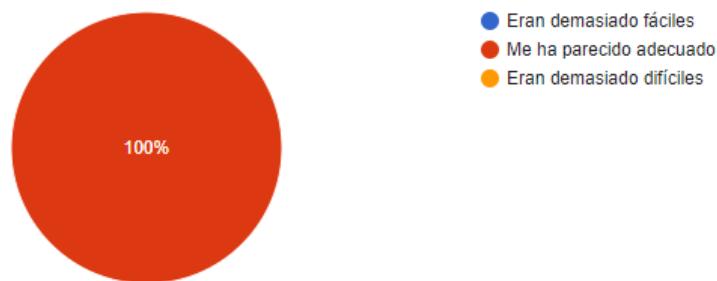


Figura 7.82: Pregunta 6 de la encuesta 2 de usabilidad.

Algunas sugerencias con respecto a los retos han sido las siguientes:

¿Qué podría mejorar con respecto a los retos que tienes que ir resolviendo para avanzar?

4 respuestas

Me han gustado mucho. Añadiría una versión infantil para que a los niños les fuese más fácil jugar.

Los acertijos me parecen adecuados. La existencia de pistas ayuda en caso de dificultad.

Creo que están bien tal y como están.

Los retos han estado bien teniendo en cuenta que es un juego por asistente de voz. No sé las limitaciones que tiene pero podría hacerse una especie de Simón dice con sonido, que te diga una clave para cada sonido y luego tener que repetir una secuencia.

También me gustaría que hubiera más objetos para combinar del tipo linterna y pilas. O ir recogiendo fragmentos, de una fotografía o carta, y cuando se tengan todos poder juntarnos y leer lo que pone.

Figura 7.83: Pregunta 7 de la encuesta 2 de usabilidad.

Con respecto a lo que cambiarían del juego, se han obtenido respuestas variadas.

¿Cambiarias algo? ¿El qué? (De forma general en el juego).

4 respuestas

Añadiría una versión principiante o una prueba piloto antes de comenzar.

Recomendaría, quizás, añadir una pequeña introducción explicando el objetivo general del juego. A lo mejor algo con una música del estilo de la Casa Encantada de Disney y una voz contando una especie de leyenda podría quedar bien.

Añadiría un final más en el caso de darle al gato los cuatro objetos en orden para que eso incentive el completar todo el juego.

En general, no cambiaría nada.

Figura 7.84: Pregunta 8 de la encuesta 2 de usabilidad.

Algunos usuarios han sugerido la adición de nuevos retos, una versión infantil del juego, un final alternativo en el caso de que el jugador le entregue al gato todos los objetos en orden o una introducción más interesante.

En referencia a los problemas surgidos durante el juego, sucedieron ciertos errores mientras los usuarios lo probaban. Algunos de ellos se relacionaban con la implementación de la aventura y pudieron solucionarse. Sin embargo, otros fallos hacían referencia a lo repetitivos que resultaban los asistentes o a que a veces no entendían bien ciertas órdenes, con lo que no eran defectos vinculados con el juego. Por tanto, todos los errores relativos al desarrollo se solventaron de forma satisfactoria.

¿Has tenido algún problema mientras usabas el juego? Si es así, ¿cuál? (Puede ser que no te resultara intuitiva la manera de interactuar, que el juego fallara en algún punto...).

4 respuestas

A veces el asistente virtual se salía del juego.

La forma de interactuar es muy sencilla y no genera problemas. Alexa es a veces un poco exasperante cuando no se entera de algo (pero es lo que hay). El juego se borró una de las veces.

El asistente de voz a veces falla, pero no creo que sea por cosa del juego.

Si, alguna vez no me reconocía las órdenes correctamente o no me dejaba realizar órdenes que sí se podían en un momento.

Figura 7.85: Pregunta 9 de la encuesta 2 de usabilidad.

La mayoría de encuestados encuentra este tipo de juegos interesante, con lo que su opinión es muy importante al tratarse de potenciales usuarios.

¿Te parecen interesantes este tipo de juegos en asistentes de voz?

4 respuestas

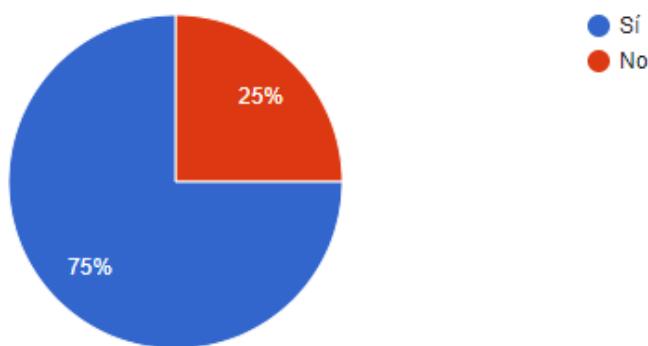


Figura 7.86: Pregunta 10 de la encuesta 2 de usabilidad.

Los últimos apartados de la encuesta aluden a lo que les ha gustado más y menos a los usuarios y a sus sugerencias sobre la aventura.

¿Qué es lo que más te ha gustado?

4 respuestas

La trama y los retos

El diseño del juego, las múltiples opciones, la temática y la implementación de una sala de escape mediante un asistente virtual.

El que tenga un camino alternativo y diversos finales.

Me ha gustado el juego en general: tanto la trama, como retos y las historias alternativas que hacen que el juego tenga diferentes finales. Las pistas que te da para seguir avanzando también son buenas. También me ha gustado la consulta de los poetas y que sea un reto.

Figura 7.87: Pregunta 11 de la encuesta 2 de usabilidad.

¿Qué es lo que menos?

4 respuestas

Que el asistente virtual no entendiera todas las indicaciones.

Alexa es a veces repetitiva y algún problema al haberse reseteado solo el juego.

El tener que entregar los objetos en orden al gato sin tener todas las pistas en el inventario.

Por poner una pega, que no se repitan algunas cosas como usar la silla tanto para coger las pilas como la radiografía. Pero entiendo las limitaciones que tiene en cuanto a la inclusión de nuevas soluciones. No he llegado a entender muy bien el sistema de puntos y su finalidad, aunque he entendido que es una medida de lo bien que se ha superado el juego

Figura 7.88: Pregunta 12 de la encuesta 2 de usabilidad.

Sugerencias, comentarios, opinión...

4 respuestas

Me ha gustado mucho y tiene muchísimo trabajo. ¡Enhorabuena!

Añadir una intro. El juego en general está muy bien y resulta muy entretenido. El hecho de que se pueda guardar también está muy bien.

Por mi parte propondría añadir un final más como he añadido anteriormente. También me parecería mejor para poder dar los cuatro objetos en orden al gato en el final del juego que las pistas estuvieran en el inventario, en vez de tener dos ahí y otras dos fuera de este y en distintas habitaciones, por lo que necesitas revisitarlas.

El juego está muy bien y es muy entretenido. Esperando la segunda parte jeje. Así como ideas locas que no sé si pueden hacer o no pues podría ser introducir diferentes voces durante el juego o incluir un sonido ambiente por habitación, lo que también ayudaría su situarse.

Figura 7.89: Pregunta 13 de la encuesta 2 de usabilidad.

Se han valorado los retos de forma positiva, así como la existencia de una trama alternativa. Por otra parte, los fallos durante el juego, la repetición en el uso de ciertos objetos o el final de la historia principal les han gustado menos a los jugadores.

Finalmente, ha habido sugerencias relativas a la modificación del final, la creación de una segunda parte o la adición de una introducción.

A partir de las opiniones recibidas, se han mejorado ciertos aspectos del juego y se han modificado algunas cosas. Por ejemplo, se han cambiado diálogos que eran malinterpretados, se ha añadido una introducción a la aventura, se han incorporado sonidos que te sumergen en la temática del juego desde el primer momento, se ha incluido un nuevo final... Además, se han adaptado las pistas a la información que se ha podido ver que los usuarios necesitaban en cada momento.

## 7.5. Publicación

Tras finalizar las entregas previstas, se ha procedido a la publicación del juego en *Alexa* y *Google Assistant*. Ambos procedimientos tardan un tiempo, dado que la aplicación desarrollada debe pasar un proceso de revisión. Por este motivo, aún no están disponibles en el mercado.



Figura 7.90: Vista previa del juego en *Google Assistant*.

Una vez que la *skill* se publique, puede descargarse desde *Amazon*. Para ello, simplemente hay que iniciar sesión y buscar en la barra de navegación el nombre de la *skill*.

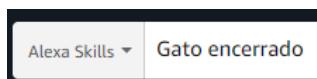


Figura 7.91: Búsqueda de la *skill* publicada en *Amazon*.

Después de descargarla, estará disponible en nuestros dispositivos con *Alexa*.

En el caso de *Google Assistant*, podemos acceder a la página web del asistente y explorar las *actions*. De forma análoga a lo explicado con *Alexa*, debemos registrarnos y descargar la *action*. Después estará disponible para los dispositivos en los que dispongamos de *Google Assistant*.

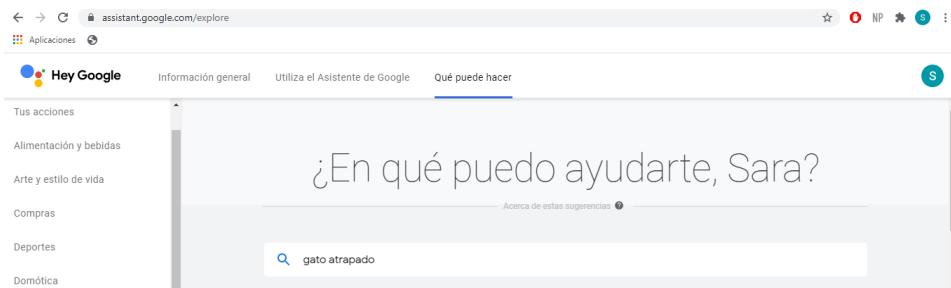


Figura 7.92: Búsqueda de la *action* publicada para *Google Assistant*.

## 7.6. Conclusiones

Hemos tratado de diseñar y desarrollar un juego atractivo para el público. Para ello, hemos procurado que la interacción resulte sencilla, y que la trama y los retos constituyan un elemento fundamental que animen al usuario a jugar.

La implementación de este proyecto se ha basado en la creación de sucesivos prototipos, siguiendo el modelo del desarrollo ágil. De este modo, se han ido incorporando sucesivos cambios y mejoras conforme el juego lo requería. Además, la opinión de los usuarios ha sido un punto clave a la hora de evaluar el proyecto. Se han realizado dos tipos de encuestas a potenciales usuarios. Por una parte, se ha tratado de obtener la opinión de un grupo de personas que no había probado el juego, con el propósito de evaluar su interés y primeras impresiones. Por otro lado, se ha encuestado a algunos usuarios después de que probaran la aventura. Esto ha permitido incorporar modificaciones en función de los deseos de los jugadores, conociendo de primera mano su experiencia al interactuar con los asistentes.

El resultado del desarrollo y las diversas entregas ha resultado, pues, satisfactorio. Se ha obtenido un prototipo del juego que, si bien podría ser mejorado pasando por una fase de diseño y marketing, es bastante completo y se ajusta a lo esperado.

## Capítulo 8

# Conclusiones y trabajo futuro

Este capítulo trata las conclusiones finales del proyecto. En primer lugar, se discute el cumplimiento de los objetivos propuestos inicialmente. A continuación, se realiza una comparativa de las plataformas de desarrollo utilizadas. Finalmente, se concretan posibles trabajos futuros relacionados con el juego implementado.

### 8.1. Cumplimiento de los objetivos

El desarrollo de este proyecto ha permitido la consecución de los objetivos especificados en la introducción de la memoria. En primer lugar, se ha cumplido el objetivo principal de desarrollar un juego de tipo aventura conversacional para asistentes virtuales.

Con respecto a los subobjetivos mencionados en el capítulo 1:

1. El juego está disponible para varias plataformas, ya que puede usarse tanto en *Alexa* como en *Google Assistant*.
2. La aventura combina elementos narrativos (juegos de rol), interacción mediante el uso de distintos comandos (aventuras conversacionales), resolución de retos (*escape room*) y toma de decisiones que influye en el desarrollo del juego (librojuegos).
3. La existencia de varios finales, y la variación del juego en función de las decisiones que tomes lo dota de un carácter dinámico. Además, el

código se ha dividido en ficheros que separan las habitaciones, de modo que sea más sencillo añadir o quitar una nueva.

4. Tanto *Google Assistant* como *Alexa* incluyen sonidos a lo largo del recorrido. Además, si poseemos un dispositivo *Alexa* con pantalla, también podremos ver imágenes.
5. La partida puede guardarse. Esto permite que el usuario pueda dejar la partida y retomarla por el punto por el que iba. Además, se trata de una funcionalidad muy útil, ya que tanto *Alexa* como *Google Assistant* cierran el juego si no se producen interacciones de forma continuada. Así, no se pierde el progreso.
6. El juego está disponible en inglés y español.

## 8.2. Comparativa de las plataformas

Durante el desarrollo de este proyecto se han utilizado principalmente dos plataformas: *Alexa Developer Console* y *Dialogflow*.

*Alexa Developer Console* es la plataforma para desarrollar *skills* para *Alexa*. Se trata de un entorno muy cómodo. Ha sido posible implementar todo el juego desde la propia plataforma de forma fácil.

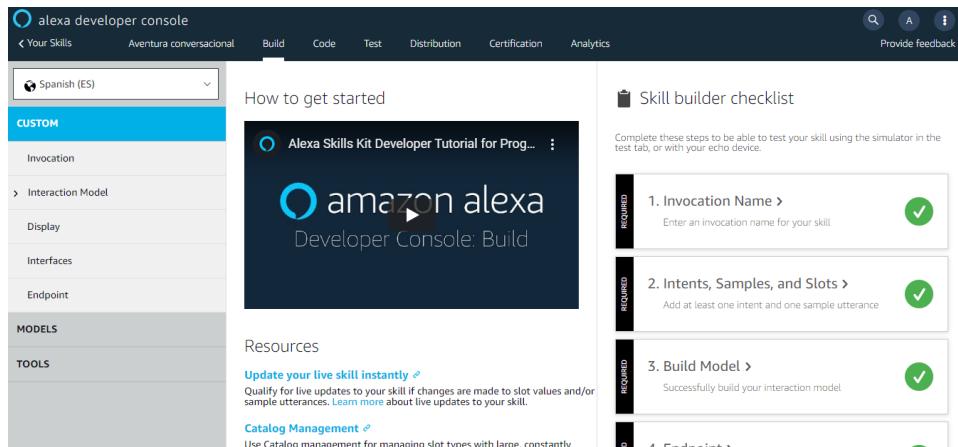


Figura 8.1: *Alexa Developer Console*.

Por un lado hemos añadido la funcionalidad asociada a los *intents* y las *utterances*. Por otra parte, hemos incluido todo el código asociado al juego, dividiéndolo en archivos y carpetas y sin necesidad de alojarlo de forma externa.

Además, la librería de sonidos, *Alexa Kit Sound Library*, es muy extensa y nos ha permitido darle mayor profundidad a la aventura. El *Alexa Presentation Language* también nos ha resultado muy útil e interesante para incorporar imágenes e, incluso, añadir botones con los que el usuario puede interaccionar desde su dispositivo *Alexa* con pantalla.

Por último, hemos podido usar el servicio de almacenamiento *S3* de *Amazon Web Services*, lo que nos ha permitido disponer de todas las imágenes y algunos de los sonidos que he añadido a la aventura accediendo a ellos mediante su *url*.

Con respecto a *Dialogflow*, es la plataforma que hemos utilizado para desarrollar una *action* para *Google Assistant*. A través de *Google Actions Console*, hemos usado *Dialogflow* para crear una aplicación más personalizada e integrarla al asistente de *Google*.

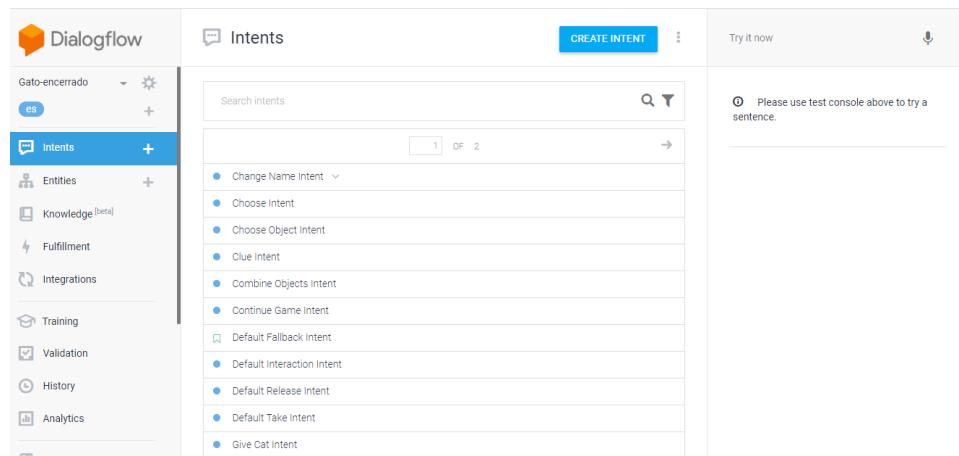


Figura 8.2: *Dialogflow*.

Esta plataforma ha sido menos cómoda que la de *Alexa*. Dadas las limitaciones a la hora de alojar el código en el editor en línea, hemos usado *Firebase* para conectar *Dialogflow* con el código almacenado en el ordenador.

En el ámbito de los *intents* y *utterances* no ha habido mayor problema que con *Alexa Developer Console*. La interfaz varía un poco de una plataforma a otra, pero en ambas son fáciles de utilizar.

Por otra parte, la librería de sonidos de *Google Sound Library*, es bastante más limitada que la de *Alexa*. Debido a que los sonidos de la librería de *Alexa* no son funcionales para el asistente de *Google*, hemos tenido que modificarlos y, con ello, hemos perdido mucha variedad que aportaba un aspecto importante de la narrativa al juego.

Por último, la voz del asistente de *Google* en español es muy robótica y desagradable. Aunque el reconocimiento parece ser algo mejor que el de *Alexa*, jugar escuchando ese tono mecánico hace que la aventura pierda interés.

### 8.3. Conclusiones

Una vez finalizado el desarrollo, se ha obtenido un producto que cumple satisfactoriamente con todos los objetivos propuestos de forma inicial. Además, dada la metodología ágil seguida, el juego ha ido mejorando su funcionalidad y características de forma progresiva en cada iteración.

El enfoque centrado en los usuarios también ha constituido un punto clave que ha permitido conocer de primera mano las opiniones y deseos de los potenciales jugadores. Esto ha sido posible mediante la realización de dos encuestas de evaluación. La primera encuesta trataba de obtener las impresiones de un grupo de usuarios que no había probado la aventura. La idea de este análisis de usabilidad era conocer los intereses de un público variado, modificando los aspectos del juego que pudieran resultar menos atractivos para conseguir un producto deseable a la vez que funcional. Por otro lado, la segunda encuesta constaba de dos partes. En primer lugar, los usuarios jugaban a la aventura. A continuación, tras haber tenido dicha experiencia, respondían a una encuesta que evaluaba su satisfacción y opinión con el prototipo testado. Así, se ha podido observar la interacción de los jugadores con los asistentes, los problemas surgidos y las opiniones recibidas después de la simulación. También se han llevado a cabo cambios en función de las sugerencias y de las respuestas recibidas en la encuesta. Todo ello ha permitido que el proyecto haya seguido un proceso iterativo con un esquema similar al de la producción de videojuegos.

El prototipo que se ha obtenido dispone de una gran versatilidad. Debido a la separación de las habitaciones en archivos, la ampliación o reducción del juego es sencilla. Esto aporta flexibilidad y permite tomar el proyecto como base para producir nuevas ideas o historias. Además, a partir del análisis de los juegos de aventura en asistentes, se han observado los aspectos menos explotados por otras aplicaciones y se han tratado de mejorar las funcionalidades existentes. Este es el motivo por el que se ha desarrollado un proyecto en el que se han combinado elementos de distintas categorías de juegos de aventura. El entretenimiento asociado a la resolución de retos, la existencia de un personaje que interactúa con los objetos, la toma de decisiones, la narrativa o la variedad de tramas son elementos que pretenden involucrar al usuario y hacer que tenga una experiencia entretenida. La incorporación de sonidos e imágenes también tratan de crear una atmósfera

más envolvente para el jugador.

Con respecto a los controles, se ha buscado una interacción que resultara fácil para el usuario pero que a la vez le permitiera disfrutar de cierta flexibilidad para poder elegir lo que hacer. Debido a que existe una amplia variedad de comandos, se ha dividido la ayuda del juego en ayuda corta y larga. Esto permite que el jugador empiece por conocer lo básico del juego y, una vez que sepa cómo funciona la interacción, pueda consultar la ayuda larga y explorar todas las posibilidades.

Por último, se ha considerado de gran utilidad añadir una función de guardado. Ninguno de los juegos analizados en el capítulo 3 incluía dicha función. Esto podía resultar tedioso, dado que la interacción con los asistentes ha ser constante y, en el caso de cerrarse el juego, podía perderse todo el progreso. En este caso, debido a que la aventura es larga y tiene varios posibles desarrollos de la trama, sería especialmente molesto tener que volver a empezar si la aplicación se cerrara. Así, guardar la partida resulta muy rentable y beneficioso para el jugador.

## **8.4. Trabajos futuros**

A la hora de desarrollar un trabajo futuro, una de las posibilidades sería incorporar algunos de los cambios sugeridos por los usuarios. Aunque ya se han implementado las principales sugerencias, que mejoran ciertos aspectos del juego, podría ser interesante considerar más posibilidades. Una de ellas sería producir el juego para otras plataformas en las que pudiera integrarse un modo multijugador. Además, podrían añadirse niveles con menor y mayor dificultad, incluir nuevos retos y funcionalidades, personalizar los sonidos y las voces para crear una experiencia más interactiva e inmersiva...

Dada la versatilidad del juego desarrollado y su potencial, la adición de este tipo de características, así como la ampliación de la aventura, tendrían un punto de partida idóneo que permitiría gran flexibilidad.

También sería atractivo llevar a cabo una etapa de posproducción en mayor profundidad. En ella, podría elaborarse una campaña publicitaria para distribuir el juego y promoverlo. Incluso podría realizarse un estudio de mercado para evaluar a los potenciales usuarios y enfocar el juego al público dispuesto a consumirlo.

Con respecto al diseño, podrían crearse imágenes especialmente planteadas para el proyecto. Así, cada habitación tendría su imagen personalizada y, al mirar en cada dirección, podrían verse todos los objetos que se encuentran en ese lugar.

Por último, podrían incluirse ciertas características especiales en el juego que solo estuvieran disponibles si se compra la aplicación. Algunas de las posibilidades serían habitaciones especiales, tramas secretas, más niveles... La aventura seguiría estando disponible de forma gratuita, pero esto permitiría monetizar ciertos contenidos extra para recibir beneficios por parte de los usuarios más interesados.

# Bibliografía

Amazon Developer. (Sin fecha). *Create the Interaction Model for Your Skill.* Recuperado de <https://developer.amazon.com/es-ES/docs/alexa/custom-skills/create-the-interaction-model-for-your-skill.html>.

Amazon Developer. (Sin fecha). *Host a Custom Skill as an AWS Lambda Function.* Recuperado de <https://developer.amazon.com/en-US/docs/alexa/custom-skills/host-a-custom-skill-as-an-aws-lambda-function.html#about-lambda-functions-and-custom-skills>.

Davies, C. (2019, Febrero). *Por dónde empezar con el desarrollo de Skills de Alexa.* Planeta Chatbot. Recuperado de <https://planetachatbot.com/desarrollo-de-skills-de-alexa-developer-console-f5efedda02cf/>

De La Cruz, B. (2019, Agosto). *Cómo solucionar problemas en un skill de Amazon Echo [Debug Alexa Skill].* Recuperado de <https://www.benjamindelacruz.com/problemas-debug-alexa-skill/>

Digital Trends. (2020, Febrero). *Algunas preguntas divertidas para el Asistente de Google.* Recuperado de <https://es.digitaltrends.com/entretenimiento/preguntas-divertidas-asistente-google/>

Digital Trends. (2020, Junio). *Preguntas divertidas para Alexa que alegrarán tu día.* Recuperado de <https://es.digitaltrends.com/inteligente/preguntas-divertidas-para-alexa/>

García, S. (2020, Enero). *Asistentes virtuales y videojuegos.* Planeta Chatbot. Recuperado de <https://planetachatbot.com/asistentes-virtuales-videojuegos-b98443e7f8d7>

Guevara Benites, A. (2016, Septiembre). *Frontend y Backend.* DevCode. Recuperado de <https://devcode.la/blog/frontend-y-backend/>

Hassan-Montero, Y.; Ortega-Santamaría, S. (2009). Diseño Centrado en el Usuario (DCU). *Informe APEI sobre Usabilidad.* Gijón: Asociación Profesional de Especialistas en Usabilidad e Interacción.

sional de Especialistas en Información, 2009, 73pp. ISBN: 978-84-692-3782-3.

Junior, A. (2018, Abril). *Qué es el Desarrollo ágil de software*. Recuperado de <https://adictec.com/que-es-desarrollo-agil-de-software/>.

Kikel, C. (Julio, 2016). *A Brief History of Voice Recognition Technology*. Total Voice Technologies. Recuperado de <https://www.totalvoicetech.com/a-brief-history-of-voice-recognition-technology/>.

Manrubia Pereira, A.M. (2014, Marzo). El proceso productivo del videojuego: fases de producción. *Historia y Comunicación Social*. Volumen (19), 791-805.

Menocal, O. (2019, Septiembre). *Tutorial parte 1: Desarrolla aplicaciones de voz multiplataforma con VOXA*. Planeta Chatbot. Recuperado de <https://planetachatbot.com/tutorial-parte-1-desarrolla-aplicaciones-de-voz-multiplataforma-con-voxa-cdee3f427f2a?source=false-----2>.

Ordóñez, C. [@SoftwareChido]. (2019, Junio). *Construyendo la computadora de Navegación Espacial*. [Archivo de vídeo].. Recuperado de <https://dex-microsites-prod.s3.amazonaws.com/video/mx/2019-06-26-12-51-47.mp4>.

Sánchez, J. (2011, Septiembre). En busca del Diseño Centrado en el Usuario (DCU): definiciones, técnicas y una propuesta. *No Solo Usabilidad*. Recuperado de <http://www.nosolousabilidad.com/articulos/dcuc.htm>.

Tolliday, T. (2019, Diciembre). *Creating Voice Skills For Google Assistant And Amazon Alexa*. Smashing Magazine. Recuperado de <https://www.smashingmagazine.com/2019/12/voice-skills-google-assistant-amazon-alexa/>.

Viscuso, G. (Sin fecha). *Curso oficial para desarrollo de Skills de Alexa*. Plataforma Keepcoding. Recuperado de <https://plataforma.keepcoding.io/p/curso-desarrollo-alexa-skills>.

Voxa. (Sin fecha). *voxa* Recuperado de <http://voxa.ai/>.

Wikipedia, La enciclopedia libre. (2020, Enero). *Dialogue System*. Recuperado de [https://en.wikipedia.org/wiki/Dialogue\\_system](https://en.wikipedia.org/wiki/Dialogue_system).

Wikipedia, La enciclopedia libre. (2020, Abril). *Software Agent*. Recuperado de [https://en.wikipedia.org/wiki/Software\\_agent](https://en.wikipedia.org/wiki/Software_agent).

Wikipedia, La enciclopedia libre. (2020, Mayo). *CALO*. Recuperado de <https://en.wikipedia.org/wiki/CALO>.

Wikipedia, La enciclopedia libre. (2020, Junio). *Asistente Virtual*. Recuperado de [https://es.wikipedia.org/w/index.php?title=Asistente\\_virtual&oldid=127320869](https://es.wikipedia.org/w/index.php?title=Asistente_virtual&oldid=127320869).

Wikipedia, La enciclopedia libre. (2020, Junio). *Flowchart*. Recuperado de <https://en.wikipedia.org/wiki/Flowchart>.

Wikipedia, La enciclopedia libre. (2020, Junio). *Virtual assistant*. Recuperado de [https://en.wikipedia.org/wiki/Virtual\\_assistant](https://en.wikipedia.org/wiki/Virtual_assistant).

Wikipedia, La enciclopedia libre. (2020, Julio). *Agile software development*. Recuperado de [https://en.wikipedia.org/wiki/Agile\\_software\\_development](https://en.wikipedia.org/wiki/Agile_software_development).

