



Universidad
Zaragoza

Trabajo Fin de Grado

Asistente virtual para la prevención del deterioro
cognitivo en personas mayores

Virtual assistant to prevent cognitive decline in
elderly people

Autora

Pilar Ester de Val

Directora

Sandra Baldassarri

Escuela de Ingeniería y Arquitectura
2023



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe remitirse a seceina@unizar.es dentro del plazo de depósito)

D./Dña. **Pilar Ester de Val**,

en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de Estudios de la titulación de

Grado en Ingeniería Informática



(Título del Trabajo)

Asistente virtual para la prevención del deterioro cognitivo en personas mayores.

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, **31/08/2023**

Fdo: **Pilar Ester de Val**

AGRADECIMIENTOS

A Sandra Baldassarri por dirigir y realizar el proyecto conmigo. Sin su ayuda este proyecto no habría sido posible.

A mis compañeros de clase, con los que he aprendido mucho, tanto de los buenos como de los malos momentos. Estos cuatro años han sido difíciles, pero junto a ellos se han hecho mucho más llevaderos.

A mi familia y amigos por su apoyo, cariño y motivación en todo momento; en especial a mis padres y hermanos por hacerme ver siempre el lado bueno de las cosas.

Asistente virtual para la prevención del deterioro cognitivo en personas mayores

RESUMEN

Aunque hoy en día existen muchas aplicaciones orientadas al entrenamiento de la memoria de las personas mayores, este grupo de personas no las suele utilizar debido a la dificultad de las personas de avanzada edad para entender la tecnología y manipular dispositivos electrónicos. Esto fomenta que la dificultad para que los mayores puedan entrenar la memoria por su cuenta aumente considerablemente.

Para mitigar este problema, en este trabajo de fin de grado se ha desarrollado una aplicación que permite a las personas mayores entrenar su memoria por medio de tests cognitivos periódicos y con el uso de tres minijuegos distintos, que permiten trabajar de distintas maneras (por voz, por voz con apoyo visual y con varios jugadores) y mediante los cuales obtienen puntos y recompensas. A diferencia de otras aplicaciones del mercado, en este caso se permite también que los cuidadores observen el progreso de los mayores mediante la visualización de los resultados de los tests y las puntuaciones conseguidas, entre otras cosas. Para hacer la experiencia lo más personalizada posible, los cuidadores también pueden configurar ciertos aspectos como las recompensas para el usuario, la periodicidad de los tests y algunos parámetros de accesibilidad de la pantalla del dispositivo.

Adicionalmente, en este trabajo se ha llevado a cabo una evaluación con un pequeño grupo de usuarios finales, tanto personas mayores como cuidadores, con el fin de verificar el funcionamiento de la aplicación, la adecuación de las tecnologías utilizadas y posibles puntos débiles y aspectos a extender o mejorar.

Virtual assistant to prevent cognitive decline in elderly people

ABSTRACT

Although nowadays there are many applications aimed at training the memory of elderly people, this group of individuals usually doesn't use them due to the difficulty that older people have in understanding technology and manipulating electronic devices. This promotes a significant increase in the difficulty for the elderly to train their memory on their own.

To mitigate this problem, in this undergraduate thesis, an application has been developed that allows elderly individuals to train their memory through periodic cognitive tests and the use of three different mini-games. These mini-games allow them to work in different ways (by voice, by voice with visual support, and with multiple players) and through which they earn points and rewards. Unlike other applications in the market, in this case, caregivers are also allowed to observe the progress of the elderly by visualizing the test results and achieved scores, among other things. To make the experience as personalized as possible, caregivers can also configure certain aspects such as user rewards, test frequency, and some device screen accessibility parameters.

Additionally, in this work, an evaluation has been carried out with a small group of users, both elderly individuals and caregivers, in order to verify the functionality of the application, the suitability of the technologies used, and possible weaknesses and aspects to expand or improve.

Índice

1	Introducción y objetivos	1
1.1	Contexto del proyecto	1
1.2	Motivación y problema abordado	1
1.3	Objetivos del proyecto	2
1.4	Estructura del documento	3
2	Análisis del sistema	5
2.1	Estudio de aplicaciones similares	5
2.1.1	Público objetivo	5
2.1.2	Resumen de las aplicaciones estudiadas	6
2.2	Requisitos de la aplicación	7
3	Diseño del sistema	11
3.1	Arquitectura software del sistema	11
3.2	Tecnologías utilizadas	13
3.3	Decisiones de diseño	14
4	Implementación del sistema	17
4.1	Componentes generales	17
4.1.1	Alexa Presentation Language: Interfaz visual del dispositivo	17
4.1.2	Tono de voz y sonidos del dispositivo Alexa en la aplicación	20
4.1.3	Extensión de la aplicación a distintos idiomas	21
4.2	Componentes de cuidador	21
4.2.1	Recordatorios para el usuario	21
4.2.2	Aspectos configurables de la aplicación	22
4.2.3	Visualización de datos	24
4.3	Componentes de usuario	25
5	Evaluación	29
5.1	Metodología	29
5.2	Resultados y análisis	30

6	Gestión del proyecto	35
6.1	Desarrollo del proyecto	35
6.2	Dedicación	36
7	Conclusiones y trabajo futuro	39
7.1	Conclusiones	39
7.2	Valoración desde el punto de vista personal	40
7.3	Trabajo futuro	40
	Glosario	42
	Bibliografía	45
	Bibliografía	45
	Lista de Figuras	48
	Lista de Tablas	49
	Anexos	50
A	Detalle de aplicaciones similares estudiadas	53
B	Estudio de la base de datos - Amazon S3 frente a DynamoDB	57
C	Detalle del módulo de sonidos especiales y de la librería de sonidos de Alexa	59
D	Extensión de la aplicación a distintos idiomas	61
E	Detalle de la configuración del modo y el tamaño de letra de la pantalla del dispositivo	65
F	Detalle del cálculo y establecimiento de la periodicidad de los tests	67
G	Inclusión de nuevos minijuegos en la aplicación	69
H	Detalle de los minijuegos implementados	71

Capítulo 1

Introducción y objetivos

En este capítulo se muestra el contexto de este trabajo de fin de grado, la motivación y problema a abordar, y los objetivos del mismo; así como el esquema general de la memoria.

1.1. Contexto del proyecto

La Organización Mundial de la Salud [16] advierte de que la proporción de envejecimiento de la sociedad está en continuo crecimiento y prevé que el número de personas mayores de 80 años se triplique en los próximos 27 años, hasta alcanzar la cifra de 426 millones. Este grupo de personas tiene necesidades distintas al resto por el deterioro que experimentan cuando van envejeciendo.

Las personas pertenecientes a este segmento de la población afrontan desafíos cognitivos que se intensifican conforme envejecen. Adicionalmente, muchas de ellas experimentan sentimientos de soledad y aislamiento que contribuyen a la pérdida de memoria.

Los juegos orientados a estimular la memoria suponen una estrategia efectiva para que las personas de avanzada edad puedan entrenar su mente en un entorno relajado, disfrutando al mismo tiempo de momentos agradables [21], [24].

1.2. Motivación y problema abordado

Las personas mayores disponen de aplicaciones en el mercado con las que jugar y a su vez entrenar la memoria. Pero en un mundo en el que la tecnología está en constante evolución, las personas de avanzada edad se están quedando rezagadas en la adaptación al mundo digital y tienen que lidiar diariamente con la barrera tecnológica que no son capaces de superar.

A pesar de que sí existen ciertos juegos que estimulan la memoria, la mayoría de ellos están desarrollados para tablet, ordenador o teléfono móvil, dispositivos con los que los

ancianos no están acostumbrados a interactuar y que, por lo tanto, son una herramienta difícil de utilizar para ellos. De esta manera, se pretende encontrar una solución para que este grupo de edad pueda entrenar su memoria de manera más asequible y fácil y se deduce que este problema podría reducirse si el mercado ofreciese aplicaciones cuya interacción con el usuario sea por voz, pues se trata de un tipo de relación que resulta más natural y sencilla para las personas.

Es por eso, que al igual que estas personas cuidaron de sus hijos y nietos algún día, hoy es el turno de las nuevas generaciones de cuidar a sus mayores, ayudándoles a fortalecer su memoria y facilitando su interacción con los dispositivos electrónicos.

Estos son los motivos que han impulsado el desarrollo de una aplicación con interacción por voz que permita a los mayores entrenar su memoria y al cuidador controlar el progreso del usuario gracias a la herramienta en cuestión.

La aplicación puede ser una herramienta útil en dos sentidos. El primero de ellos es que ayudará a disminuir la sensación de soledad con la que conviven día a día las personas dependientes que no pueden contar con supervisión las 24 horas. El segundo, que la población de avanzada edad podrá entrenar su memoria desde casa y de manera autónoma.

1.3. Objetivos del proyecto

El proyecto consiste en desarrollar una aplicación para prevenir el deterioro cognitivo en personas mayores utilizando la interacción por voz con apoyo visual. Para ello, la aplicación ofrece dos perfiles: uno para los usuarios, que serán mayormente las personas de edad avanzada; y otro para sus cuidadores. La aplicación debe permitir a los usuarios entrenar su memoria a través de minijuegos con los que podrán desbloquear recompensas personalizadas, realizar tests de manera periódica para evaluar su progreso cognitivo y recibir recordatorios sobre otros aspectos configurables, como cuándo deben tomarse determinada medicación o cuándo hacer videollamada con familiares. Por otra parte, desde el perfil del cuidador la aplicación debe permitir visualizar y evaluar el progreso cognitivo de los usuarios así como configurar ciertos aspectos del sistema para personalizar la experiencia.

A continuación, se enuncian los objetivos específicos a abordar en el proyecto.

- Buscar y estudiar aplicaciones similares existentes actualmente.
- Estudiar y seleccionar tests médicos para el deterioro cognitivo.
- Desarrollar una aplicación con minijuegos de memoria para el usuario final, basados en la interacción por voz y/o con soporte visual.

- Permitir visualizar los progresos del usuario por parte del cuidador.
- Permitir ajustar aspectos de configuración a la persona que cuida o está a cargo del usuario.
- Estudiar métodos de almacenamiento y recuperación de información del usuario y del cuidador para poder utilizarlos posteriormente.
- Implementar la aplicación para dos dispositivos distintos (Alexa Echo y Alexa Echo Show).
- Evaluar la aplicación desarrollada mediante técnicas de evaluación de la UX con usuarios finales (personas mayores) y con cuidadores.

1.4. Estructura del documento

Este documento se divide en dos partes. La primera contiene la memoria del proyecto, en la que se presenta, de forma resumida, el trabajo llevado a cabo. La segunda parte contiene los anexos que explican de manera más detallada ciertos aspectos específicos de la memoria.

La memoria cuenta con los siguientes capítulos:

- Capítulo 1 - Introducción y objetivos: explica el problema, la motivación y los objetivos a abordar en el proyecto.
- Capítulo 2 - Análisis del sistema: presenta las aplicaciones similares analizadas y el listado de requisitos funcionales y no funcionales.
- Capítulo 3 - Diseño del sistema: se compone por la arquitectura de la aplicación, las tecnologías y herramientas que se utilizan en su implementación y las decisiones de diseño establecidas.
- Capítulo 4 - Implementación del sistema: contiene los aspectos más importantes de la implementación de la aplicación.
- Capítulo 5 - Evaluación: se presenta el detalle de la evaluación de la aplicación con usuarios y cuidadores, los resultados que se obtienen y las posibles mejoras.
- Capítulo 6 - Gestión del proyecto: narra la planificación del proyecto, la metodología utilizada y el tiempo dedicado al proyecto.
- Capítulo 7 - Conclusiones y trabajo futuro: explica las conclusiones obtenidas del proyecto y posibles tareas a implementar en un futuro.

El apartado de los anexos consta de los siguientes subapartados:

- Anexo A - Detalle de aplicaciones similares estudiadas
- Anexo B - Estudio de la base de datos: Amazon S3 frente a DynamoDB
- Anexo C - Detalle del módulo de sonidos especiales y de la librería de sonidos de Alexa
- Anexo D - Inclusión de nuevos minijuegos en la aplicación
- Anexo E - Detalle de los minijuegos implementados
- Anexo F - Extensión de la aplicación a distintos idiomas
- Anexo G - Detalle de la configuración del modo y el tamaño de letra de la pantalla del dispositivo
- Anexo H - Detalle del cálculo y establecimiento de la periodicidad de los tests

Capítulo 2

Análisis del sistema

Este capítulo presenta las aplicaciones similares que se han analizado para identificar puntos fuertes y débiles a introducir y evitar en la aplicación respectivamente, seguido del análisis de requisitos que debe satisfacer la aplicación.

2.1. Estudio de aplicaciones similares

Antes de comenzar con el desarrollo de la aplicación, se lleva a cabo un proceso de búsqueda de aplicaciones similares preexistentes [13] para el público objetivo de este trabajo; previo al estudio de otras aplicaciones, se describen brevemente las características del público objetivo de la aplicación.

2.1.1. Público objetivo

Los *stakeholders* o público objetivo de este proyecto se dividen en dos grupos principales: personas mayores y cuidadores.

- Las personas mayores que quieren entrenar su memoria mediante minijuegos y tests cognitivos. Se corresponden con personas generalmente de 75 años o más, que experimentan o quieren prevenir el deterioro cognitivo. Muchas veces no tienen compañía las 24 horas del día, y debido a no recibir los cuidados, o a tener problemas físicos que dificultan su desplazamiento, no pueden ir al médico para evaluar su memoria de manera periódica. Además, este grupo de personas suele sufrir de problemas visuales y/o auditivos que dificultan el uso de dispositivos electrónicos más tradicionales, como los teléfonos móviles.
- Los cuidadores de las personas mayores. Estos individuos tienen generalmente una edad menor a la de los usuarios finales. Suelen ser familiares o personas contratadas específicamente para acompañar a los mayores en momentos

puntuales, no suelen acompañar a los mayores durante todo el día, por lo que necesitan poder almacenar de alguna manera los progresos cognitivos de sus ancianos para consultarlos posteriormente y poder tomar medidas en consecuencia.

2.1.2. Resumen de las aplicaciones estudiadas

Se lleva a cabo un proceso de búsqueda de aplicaciones similares para determinar el funcionamiento de las mismas y los aspectos que podría ser interesante adoptar en la aplicación o cuáles convendría evitar. En este proceso de búsqueda, se investiga sobre aplicaciones existentes para personas mayores que fomenten trabajar la memoria y se ejecuten en dispositivos Alexa o teléfonos móvil. En todos los casos, se estudian juegos para ancianos, lo que se muestra por la pantalla del dispositivo y, en el caso de incluir interfaz por voz, para determinar lo que realiza por voz.

La aplicación que se desarrolla en este proyecto debe contar con dos perfiles (de usuario y de cuidador), también debe contar con aspectos configurables para hacer la experiencia más personal, contar con interfaz de voz con apoyo visual y permitir al usuario realizar tests cognitivos y minijuegos para entrenar la memoria. En base a esto se lleva a cabo el estudio de aplicaciones similares. En la tabla A.1 se muestra un resumen de las aplicaciones más significativas encontradas con sus características principales. Durante el estudio de aplicaciones se estudiaron otras que no están presentes en la tabla, pero solo se han recogido las más relevantes.

En el Anexo A: Detalle de aplicaciones similares estudiadas, se presentan más detalles de las aplicaciones analizadas y los criterios utilizados para realizar la búsqueda de las mismas.

Nombre	Móvil o Alexa	Tipo de juegos	¿Guarda datos?	¿Es configurable?	¿Recompensa para usuario?
Mayores activos	Alexa	Preguntas	Sí	No	No
Reto memoria	Alexa	Repetir colores	No	No	No
Pirámide de la memoria	Alexa	Repetir palabras	No	No	No
Rehabilitación cognitiva	Móvil	Memoria	Sí	No	Sí
Fit Brain	Móvil	Memoria	Sí	No	Sí
Lumosity	Móvil	Memoria	Sí	No	Sí
Focus	Móvil	Mentales y motrices	Sí	No	Sí

Tabla 2.1: Características más relevantes de las aplicaciones estudiadas

En la tabla se observa que no existen muchas aplicaciones para Alexa que guarden datos sobre el perfil o el progreso del usuario y ninguna que sea configurable. Tras este proceso de búsqueda se llega a la conclusión de que la aplicación a implementar debe ser sencilla y contener minijuegos que sean fáciles de realizar mediante interacción por voz. Esta búsqueda también ayuda a comprender la importancia de que la aplicación sea configurable, ya que es la manera más fácil de conseguir una experiencia adaptada a cada usuario concreto y satisfacer sus necesidades.

No se ha encontrado ninguna aplicación para dispositivos Alexa que utilice la pantalla para dar apoyo al usuario, todas las aplicaciones orientadas a entrenar la memoria en estos dispositivos son por voz.

Además, se destaca que las aplicaciones estudiadas que dan recompensas a los usuarios tras conseguir superar minijuegos, son las mejor valoradas por los usuarios.

2.2. Requisitos de la aplicación

Como resultado del estudio previo y de los usuarios, se determina que se debe desarrollar una aplicación con interacción por voz, con apoyo visual, que permita a usuarios mayores realizar tests cognitivos y jugar a minijuegos, almacenando información sobre los mismos. Además, debe permitir configurar ciertos aspectos del sistema para personalizar la experiencia del usuario y permitir observar los resultados del entrenamiento del usuario.

La aplicación debe diferenciar dos perfiles distintos, de usuario y de cuidador; cada uno podrá realizar algunas de las tareas especificadas.

Este resumen de las funcionalidades de la aplicación se traduce en los siguientes requisitos.

- RF1 - El sistema debe permitir al usuario introducir sus datos de perfil.
- RF2 - El sistema debe almacenar los datos que el usuario introduzca para personalizar su experiencia.
 - RF2.1 - Los datos estáticos introducidos por el usuario que la aplicación debe almacenar son: su fecha de nacimiento, nombre, número de teléfono, nombre completo de su madre y nombre de su cuidador o cuidadora.
 - RF2.2 - Los datos dinámicos que la aplicación debe almacenar se corresponden con resultados y estadísticas sobre los tests y los minijuegos realizados.

- RF3 - El sistema debe permitir al usuario realizar tests periódicos para poder evaluar su memoria.
- RF4 - El sistema debe permitir al usuario realizar tres minijuegos de memoria distintos.
- RF5 - El sistema debe guardar los puntos conseguidos en los minijuegos y las veces que el usuario juega a cada uno.
- RF6 - El sistema debe dar al usuario una puntuación al finalizar un minijuego que será acumulable.
- RF7 - El sistema debe permitir al usuario recibir una recompensa al alcanzar un número de puntos determinado.
- RF8 - El sistema debe permitir al cuidador establecer recordatorios de distintos tipos.
- RF9 - El sistema debe permitir al cuidador consultar datos del entrenamiento del usuario.
 - RF9.1 - El sistema debe permitir al cuidador visualizar los puntos que el usuario ha acumulado hasta el momento.
 - RF9.2 - El sistema debe permitir al cuidador visualizar el número de veces que el usuario ha jugado a cada minijuego en el mes actual y resultados de los juegos.
- RF10 - El sistema debe permitir al cuidador consultar el resultado de los últimos tests realizados y compararlos entre sí.
- RF11 - El sistema debe permitir al cuidador configurar el modo de pantalla (claro/oscuro) y el tamaño de letra de la pantalla.
- RF12 - El sistema debe permitir al cuidador configurar la periodicidad con que el usuario realizará los tests de memoria.
- RF13 - El sistema debe permitir al cuidador configurar recompensas personalizadas para el usuario.
- RF14 - El sistema debe permitir al cuidador configurar la periodicidad con que el usuario recibirá las recompensas.

- RF15 - El sistema debe permitir al cuidador configurar los puntos que el usuario debe alcanzar para conseguir la recompensa establecida.
- RNF1 - El sistema debe utilizar un dispositivo Alexa Echo o Alexa Echo Show.
- RNF2 - El sistema debe ser utilizado con conexión a Internet.
- RNF3 - El sistema debe almacenar los datos en la nube en una base de datos Amazon S3.

La Figura 2.1 muestra la interacción del usuario y del cuidador con elementos de la aplicación para facilitar la comprensión de los componentes que la forman.

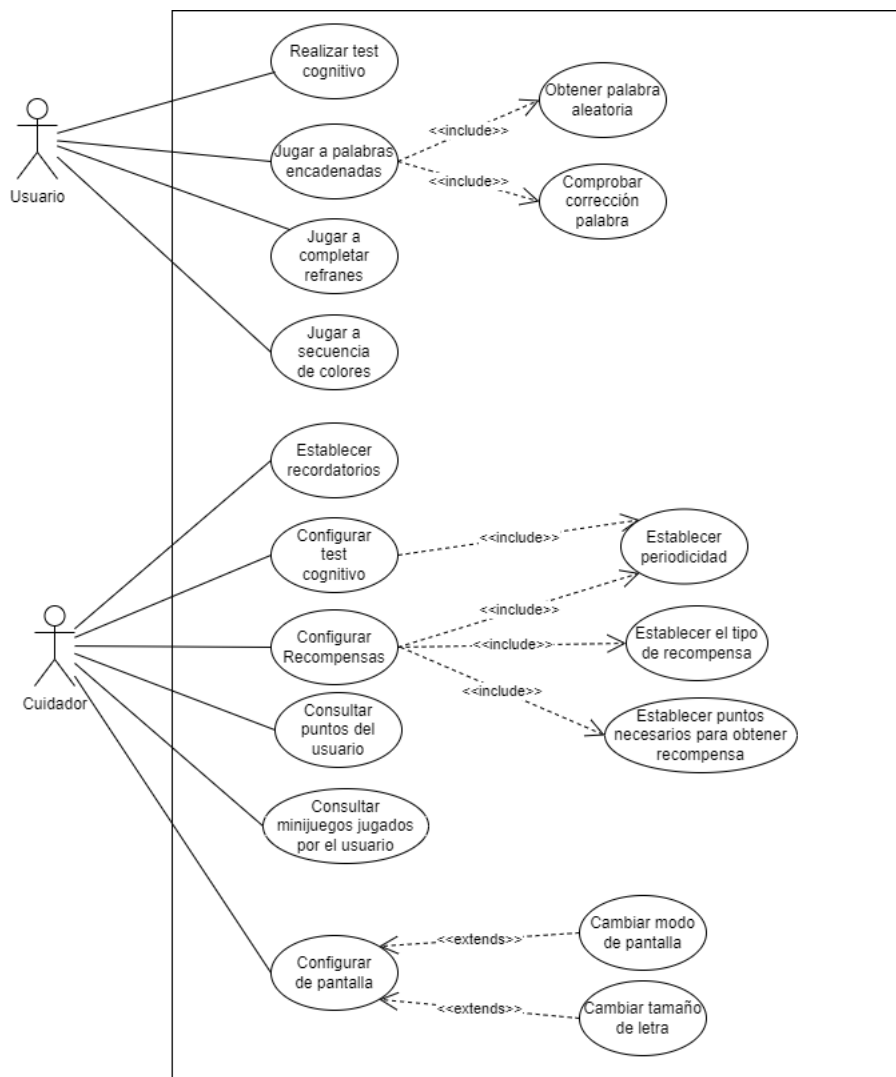


Figura 2.1: Diagrama de casos de uso del sistema.

Capítulo 3

Diseño del sistema

En este capítulo se presenta la arquitectura software del sistema desarrollado, las tecnologías y herramientas utilizadas en su implementación y las decisiones de diseño adoptadas.

3.1. Arquitectura software del sistema

En esta sección se explican los componentes desde el punto de vista del diseño. En el capítulo 4, se explicará más en detalle la implementación de los mismos. En la figura 3.1 se muestra el proceso general de interacción del usuario con la aplicación. La interacción con el usuario se realiza por medio de la voz con soporte visual a través del dispositivo Alexa Echo Show que cuenta con una pantalla que muestra información de interés al usuario [19]. Aunque la aplicación también ha sido diseñada para poder utilizarse en un dispositivo Alexa Echo sin pantalla.

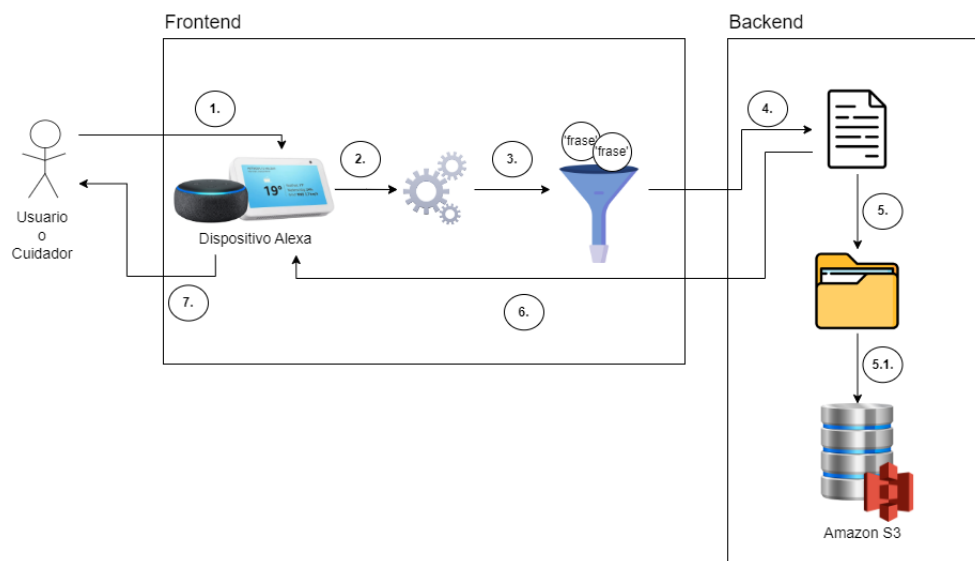


Figura 3.1: Interacción de la aplicación para Alexa Show con el usuario.

Este proceso es el mismo para el usuario y el cuidador, ya que la forma de interactuar con el dispositivo es la misma, aunque las acciones que realiza cada uno son diferentes. A continuación se explica el proceso paso a paso. Para poder comprender este proceso, el documento cuenta con un glosario de términos que se nombran a lo largo de la memoria.

- Paso 1: El usuario realiza una petición mediante la interacción por voz con el dispositivo Alexa.
- Paso 2: El dispositivo realiza un proceso de reconocimiento de voz y traduce la palabra del usuario a texto.
- Paso 3: El texto se interpreta como una sentencia que tiene un significado en la aplicación; esta reconoce la sentencia (porque se corresponde con una *utterance* asociada a un *intent*) y la asocia a un *intent* específico.
- Paso 4: Cada *intent* tiene asociado en el *backend* un *handler*, que es el encargado de recibir los datos introducidos por el usuario y construir la respuesta de la aplicación.
- Paso 5: Desde el *backend* se almacenan los datos necesarios que han sido introducidos por el usuario. Dichos datos se almacenan en los *atributos de sesión* para ser utilizados en la misma.
- Paso 5.1: Los datos también se guardan en la base de datos si el usuario decide cerrar la sesión, para poder utilizarlos en interacciones futuras.
- Paso 6: Cuando se genera la respuesta, esta se envía al *frontend*.
- paso 7: El dispositivo Alexa reinterpreta el texto y lo convierte en habla para comunicar los resultados al usuario por voz y/o de manera visual.

En la Figura 3.2 se observa el esquema con los componentes principales de la aplicación.

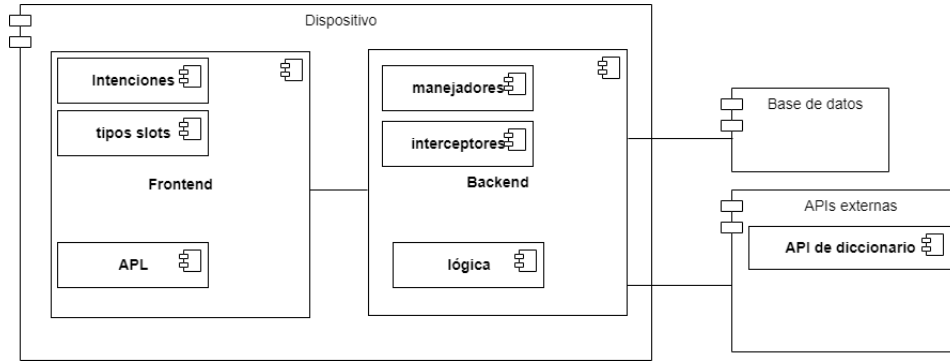


Figura 3.2: Arquitectura del sistema con componentes.

Este se constituye por tres componentes principales, las APIs externas que se utilizan para completar algunas funcionalidades, la base de datos que se corresponde con Amazon S3 y el dispositivo Alexa, que cuenta con un *frontend*, correspondiente con los *intents* o acciones que el usuario quiere ejecutar, y los tipos de datos que el usuario proporciona a la aplicación mediante el diálogo. El *frontend* también cuenta con un apoyo visual mediante la pantalla del dispositivo (si este cuenta con una) en la que se muestran frases e iconos que ayudan al diálogo del usuario y se presentan mediante el llamado *Alexa Presentation Language* (APL). El dispositivo cuenta también con un *backend*, que se compone por los *handlers* de los *intents* del usuario, los *interceptors* que se encargan de guardar los datos en los atributos de sesión o en la base de datos si la sesión va a finalizar, y la lógica de la aplicación, que se corresponde con aquellas funciones en las que se apoyan los *handlers* para dar respuestas al usuario.

3.2. Tecnologías utilizadas

Para la elección de las tecnologías y herramientas a utilizar en el proyecto, se ha investigado sobre las opciones más cómodas y comunes disponibles para el desarrollo de skills de Alexa. Los dos lenguajes por excelencia utilizados para el desarrollo de aplicaciones de Alexa son *node.js* y *python*, se decide utilizar *node.js* para la implementación de la aplicación por la experiencia previa en este lenguaje. Además, se lleva a cabo en el entorno en línea de *Alexa Developer Console* [5]. Este entorno permite llevar a cabo en el mismo la mayor parte del desarrollo, incluyendo pruebas y distribución. En cuanto al almacenamiento en base de datos, finalmente se ha decidido utilizar la base de datos *Amazon S3* frente a *DynamoDB* que fue la primera planteada, tal y como se detalla en la siguiente sección.

3.3. Decisiones de diseño

Esta sección aborda las decisiones de diseño de mayor relevancia en el contexto del proyecto. Todos los aspectos explicados en esta sección han sido estudiados también profundamente para poder llegar a las conclusiones establecidas. Este estudio será de utilidad para aquellas funcionalidades que se decidan llevar a cabo en trabajos futuros. Para empezar, se decide usar la base de datos *Amazon S3* frente a la planteada inicialmente, *DynamoDB*. Esta decisión se basa en el estudio de las características de ambas bases de datos, que se resumen a continuación. El detalle del análisis de las dos bases de datos planteadas se presenta en el Anexo B: Estudio de la base de datos - Amazon S3 frente a DynamoDB.

Se observa que *DynamoDB* ofrece escalabilidad automática, rendimiento constante, un modelo de datos flexible, consultas rápidas y soporte para transacciones. Pero tiene las desventajas de contar con costos variables y necesita un esquema de datos bien estructurado para evitar consultas ineficientes.

Por su parte, *Amazon S3* proporciona escalabilidad, almacenamiento de objetos, costos flexibles e integración con otros servicios. Aunque también cuenta con inconvenientes, pues no es adecuado para consultas complejas y tiene latencia variable.

Tras este estudio se concluye que la base de datos NoSQL *DynamoDB*, al guardar datos estructurados, es una solución sobredimensionada y es suficiente con utilizar *Amazon S3*, ya que las características y capacidades de *Amazon S3* satisfacen los requisitos de la aplicación y permiten manejar correctamente los datos almacenados por los usuarios gracias a la gran durabilidad de los datos y su eficiencia.

Por otra parte, existen diversas funcionalidades que se plantearon inicialmente y que, tras una investigación exhaustiva de las perspectivas de desarrollo de la aplicación, se ha constatado que no pueden ser implementadas dentro del plazo definido para el proyecto. A continuación, se presenta la descripción de estas funcionalidades.

- Utilización de un personaje virtual: Sería interesante incluir un personaje virtual 3D que interactúe con el usuario, simulando una conversación con el mismo. Se investigaron las posibilidades de cómo introducir dicho personaje en la aplicación, pero el tiempo que requiere el proceso de creación de un avatar virtual para la aplicación sobrepasa los límites del proyecto, por lo que se decide no llevar a cabo esta tarea, pero no se descarta para posibles trabajos futuros. Como alternativa al personaje virtual se utilizan imágenes y animaciones que ayudan a personalizar la experiencia del usuario.
- Enviar correo electrónico o mensaje de texto (SMS): Se considera que la opción

de establecer comunicación con el cuidador a través de dispositivos móviles constituiría una estrategia altamente efectiva para posibilitar la consulta de los resultados del entrenamiento del usuario. No obstante, se ha determinado que esta funcionalidad no posee carácter prioritario en el contexto de la aplicación, y por consiguiente, no ha sido incluida debido a la necesidad de una inversión temporal que excede los recursos actualmente asignados a esta labor.

Como sustituto al envío de ciertos datos del usuario mediante correo electrónico o mensajes de texto, se ha decidido implementar diversas características que facultarán al cuidador para realizar consultas directas a la aplicación en relación a estos resultados. El sistema presentará esta información en pantalla y también la proporcionará a través de salidas de voz. A pesar de esta alternativa implementada, se mantiene la posibilidad de abordar esta funcionalidad como parte de trabajos futuros.

A continuación se exponen otras funcionalidades que se plantearon inicialmente para su implementación en el proyecto con el propósito de enriquecer la aplicación, pero que ha sido imposible desarrollar debido a la ausencia de autorizaciones por parte de Amazon o por el coste desmesurado que supondría su implementación.

- Uso de la cámara del dispositivo Alexa: Inicialmente se contempló la posibilidad de permitir al cuidador capturar imágenes de los medicamentos del usuario, con el propósito de recordarle su consumo, presentándole las imágenes como referencia visual. Asimismo, se tenía la intención de emplear la cámara para tomar una fotografía diaria del usuario, permitiendo así la observación de sus emociones a través de su expresión facial. Adicionalmente, se planeaba aprovechar la cámara del dispositivo para establecer comunicaciones de vídeo regularmente con los familiares del usuario, con el fin de mitigar su sensación de soledad.

No obstante, estas tres funcionalidades no son viables para su implementación debido a las restricciones impuestas por Amazon, que prohíben el uso de la cámara en aplicaciones o *skills* para Alexa de terceros [6].

- Carga de imágenes y/o vídeos: Se deseaba permitir al cuidador grabar y subir vídeos a la aplicación para felicitar al usuario por sus logros y animarle a seguir entrenando. El inconveniente que se manifiesta en esta funcionalidad reside en la imposibilidad para efectuar el almacenamiento de un archivo de vídeo a través de comandos de voz. Para ello se necesitaría implementar una aplicación móvil desde la cual el cuidador pueda subir y almacenar el vídeo en la base de datos que corresponda y desde la aplicación por voz acceder a la base de datos para

recuperar el vídeo. Esta implementación se sale del rango abaricable del proyecto por lo que no se lleva a cabo, pero se tendrá en cuenta para trabajos futuros.

- Detectar la identidad del usuario por voz: La aplicación de Alexa [17] para teléfono móvil permite a los usuarios crear perfiles de voz que reconoce e identifica posteriormente según un identificador de voz generado en la creación de dichos perfiles. Se estudió la posibilidad de utilizar dicho identificador para reconocer la voz del usuario y prescindir de la necesidad de realizar una consulta de identificación al momento de iniciar la aplicación, pero Amazon no permite el uso de este identificador para los perfiles de voz en aplicaciones o *skills* de Alexa de terceros [11], por lo que esta tarea no se ha podido llevar a cabo.

Capítulo 4

Implementación del sistema

En este capítulo se describen de manera más detallada los componentes software más destacables implementados. Se tratan los aspectos de implementación ordenados en tres categorías. En primer lugar se detallan los componentes que son genéricos, e independientes del tipo de usuario; en segundo lugar los componentes relacionados con el cuidador y por último los relacionados con el usuario final.

4.1. Componentes generales

Como se ha explicado previamente en la sección 3.1, las aplicaciones de Alexa o *skills* de Alexa [15] constan de dos partes fundamentales; el *backend* y el *frontend*.

El *backend* de la aplicación se encarga de procesar las solicitudes enviadas por Alexa y generar respuestas apropiadas para cada interacción con el usuario; es decir, es donde se encuentra la lógica de la aplicación.

El *frontend* se corresponde con el [Voice User Interface](#), este engloba el modelo de interacción, que hace referencia a cómo los usuarios interactúan con la aplicación, y cómo esta responde a las solicitudes del usuario. Es una descripción del diálogo entre el usuario y la aplicación, permitiendo al dispositivo Alexa interpretar las solicitudes y proporcionar respuestas a las mismas.

A continuación se explican los elementos de la aplicación que son genéricos y cuyo uso se establece para todos los usuarios: Alexa Presentation Language, tono de voz y sonidos del dispositivo y extensión de la aplicación a distintos idiomas.

4.1.1. Alexa Presentation Language: Interfaz visual del dispositivo

Para mostrar información en la pantalla de los dispositivos Alexa se utiliza el framework APL, de las siglas Alexa Presentation Language en inglés. Este framework permite utilizar distintos tipos de elementos: GIFs, vídeos, textos y personajes virtuales.

En la aplicación finalmente se ha decidido incluir imágenes en los minijuegos y animaciones en formato GIF cuando el usuario finaliza un minijuego, para felicitarle. Los textos y las imágenes se utilizan principalmente para apoyar al usuario y facilitar la comunicación con el dispositivo. Por ejemplo, para que el usuario elija a qué minijuego jugar, se muestra en la pantalla una lista horizontal deslizable de los minijuegos disponibles, dicha lista contiene elementos con una imagen y un texto con el nombre de cada minijuego como se muestra en la Figura 4.1.



Figura 4.1: Pantalla inicial del perfil de usuario.

Para poder mostrar estos elementos por la pantalla se necesita crear un archivo JSON cuya estructura básica se explica a continuación y se muestra en la Figura 4.2.

```

{
  "type": "APL",
  "version": "2023.1",
  "theme": "light",
  "import": [
    {
      "name": "alexa-layouts",
      "version": "1.7.0"
    }
  ],
  "resources": [
    {
      "description": "Stock color for the theme",
      "colors": {
        "colorTextPrimary": "#151920"
      }
    }
  ],
  "layouts": {}
  "mainTemplate": {
    "items": [
      {
        "type": "Text",
        "text": "Text"
      }
    ]
  }
}

```

Figura 4.2: Estructura de un documento APL.

En el documento cuya estructura se acaba de mostrar, se importan los paquetes que se necesitan más adelante y seguidamente se establecen los recursos que se pueden referenciar en toda la interfaz. En el ejemplo de la Figura 4.2 se establece el color que se asociará posteriormente al fondo de pantalla. A continuación se define la sección de *layouts*, donde se establece cómo se estructuran y organizan los elementos en la interfaz. En el caso de la pantalla de inicio del usuario, en la sección de *layouts* se define una estructura correspondiente con una lista de elementos; es la lista de juegos que se van a mostrar por la pantalla. Por último, el apartado de *mainTemplate* define la estructura principal de la interfaz visual que se mostrará en el dispositivo, en él se organizan y colocan los elementos visuales utilizando los *layouts* establecidos previamente y se introducen los datos que se envían desde el *backend*.

```

if (util.supportsAPL(handlerInput)) {
  handlerInput.responseBuilder.addDirective({
    type: 'Alexa.Presentation.APL.RenderDocument',
    version: '2023.1',
    document: constants.APLDocs.launchLight,
    datasources: {
      launchData: {
        type: 'object',
        properties: {
          headerTitle: requestAttributes.t('LAUNCH_HEADER_MSG'),
          mainText: requestAttributes.t('CONTACT_DATA_TEXT_MSG'),
          hintString: requestAttributes.t('CONTACT_DATA_HINT_MSG'),
          textSizeBody: fontSize,
          textSizePrimary: fontSize + 10,
          textSizeSecondary: fontSize + 5,
          textSizeDetails: fontSize,
          textSizeSecondaryHint: fontSize
        },
        transformers: [{
          inputPath: 'hintString',
          transformer: 'textToHint',
        }]
      },
    },
  });
}

```

Figura 4.3: Estructura de un documento APL.

A través de la estructura mostrada en la Figura 4.3 se envían los datos concretos (como el contenido de los elementos de la lista de minijuegos) desde el *backend* al documento APL. Los elementos dentro de *properties* son aquellos datos que se envían desde el *backend* al documento APL.

4.1.2. Tono de voz y sonidos del dispositivo Alexa en la aplicación

Los dispositivos Alexa interactúan principalmente mediante la voz. Es por eso, que para conseguir una experiencia más personal y real con los usuarios, se decide utilizar dos módulos que permiten modificar el tono de la respuesta del dispositivo y añadir sonidos a las respuestas: el módulo de sonidos especiales y la librería de sonidos de Alexa.

Los *speechcons* [12] son palabras y frases especiales y típicas de los distintos idiomas que Alexa pronuncia de una manera especial.

La librería de sonidos *Alexa Sound Library* [1] permite reproducir sonidos de muy distintos ámbitos para acompañar a las frases que reproduce el dispositivo.

Para no recargar la memoria principal, el detalle de la implementación y uso de estos dos módulos se presenta en el Anexo C.

4.1.3. Extensión de la aplicación a distintos idiomas

Cabe destacar que todas las respuestas de voz del dispositivo Alexa a través de la aplicación tienen que ser configuradas manualmente. Esto conlleva la dificultad de conseguir que la aplicación pueda ser traducida automáticamente a distintos idiomas, pues si el código del *backend* está repleto de frases literales en un idioma concreto, su sustitución podría resultar una tarea tediosa.

En el *frontend* se establecen las posibles frases que el usuario va a decir para activar o acceder a los diferentes [intents](#). Estas frases están escritas también en un idioma concreto y si el usuario habla en un idioma distinto, la aplicación no comprenderá la petición y no se activará ningún [intent](#).

Para solucionar este problema, en el *backend* hay que crear una estructura que contiene las frases para los distintos idiomas, cada frase en forma de variable repetida para cada idioma disponible, permitiendo de esta manera utilizar en el código las variables de las respuestas del dispositivo, sin necesidad de cambiar la frase dependiendo del idioma utilizado por el usuario, ya que esto se realiza de manera automática.

En cuanto al *frontend* la solución es similar, se añaden idiomas para la aplicación en el entorno de desarrollo, de esta manera se permite añadir [intents](#) en otros idiomas en documentos separados para cada idioma.

La forma de permitir que la selección del idioma se realice de forma automática cuando el usuario entra en la aplicación se explica de manera detallada en el Anexo [D](#): Detalle del módulo de sonidos especiales y de la librería de sonidos de Alexa.

4.2. Componentes de cuidador

El papel del cuidador cobra una gran importancia en la aplicación, siendo el responsable de configurar ciertos aspectos del sistema y de controlar el progreso del usuario. En esta sección se detallan los componentes que están destinados al uso por parte del cuidador. A continuación, se explican los elementos más importantes como son el establecimiento de recordatorios, los aspectos configurables de la aplicación y los datos que el cuidador puede visualizar en el dispositivo.

4.2.1. Recordatorios para el usuario

Un recordatorio es una característica del dispositivo Alexa que permite a los cuidadores configurar y gestionar avisos para eventos o tareas específicas. Los recordatorios son programados por el cuidador y posteriormente serán notificados al usuario por la aplicación en el momento especificado.

La aplicación permite establecer los recordatorios con dos tipos distintos de tópicos: videollamada con familiares o la toma de la medicación.

El sistema pide al cuidador la acción que quiere establecer como recordatorio y la fecha en la que se establece. La aplicación recoge estos datos para definir el mensaje de la alerta y el momento en que debe comunicarla. Después, mediante la función *serviceClientFactory.getReminderManagementServiceClient()* la aplicación proporciona un cliente que interactúa con la API de administración de recordatorios de Alexa. A través de este cliente, el sistema puede crear, modificar y eliminar recordatorios en nombre del cuidador; primero, se revisa si hay algún recordatorio establecido anteriormente y se elimina. Tras esto, se invoca a una función que se encarga de crear un objeto con la información necesaria para configurar el nuevo recordatorio (mensaje y fecha del mismo). Dicho objeto se utiliza para crear el recordatorio para el usuario.

4.2.2. Aspectos configurables de la aplicación

Una característica destacable de la aplicación desarrollada es la gran cantidad de aspectos configurables que ofrece. Se pretende que la experiencia de usuario sea lo más personalizada posible, y sabiendo que los usuarios son personas mayores que no tienen un manejo tan avanzado de la tecnología en la mayoría de las ocasiones, es el cuidador el que puede establecer los valores de ciertos parámetros para conseguir que el usuario se sienta lo más cómodo posible utilizando la aplicación y que esta se ajuste lo máximo posible a las necesidades del usuario.

En este apartado se recogen de manera resumida los principales aspectos de configuración de la aplicación: los ajustes de la pantalla, la periodicidad de los tests, la periodicidad y el tópico de los recordatorios y las recompensas de los minijuegos.

a) Ajustes de pantalla: modo y tamaño de letra

Aprovechando la disponibilidad de una infraestructura visual a través del Alexa Presentation Language (APL), es necesario que dicha infraestructura sea adaptativa para el usuario, con el objetivo de garantizar su comodidad y proporcionar asistencia efectiva. Para ello, la aplicación permite al cuidador modificar el tamaño de la letra mostrada y si la pantalla se muestra en modo claro u oscuro. Ya se ha explicado en el apartado de APL la forma de utilizar el apoyo visual de los dispositivos Alexa Show, pero para poder modificar de manera dinámica el modo y el tamaño de la letra se ha llevado a cabo el siguiente proceso. El cuidador puede modificar el modo de la pantalla y el tamaño de letra introduciendo el nuevo valor del parámetro. Este se guarda en la base de datos de manera persistente y se recupera en cada sesión para poder mostrar

la pantalla con los parámetros ajustados. En el Anexo E: Detalle de la configuración del modo y el tamaño de letra de la pantalla del dispositivo, se explica en detalle el proceso específico para llevar a cabo estas dos tareas.

b) Periodicidad de los tests

La aplicación permite que el cuidador determine cada cuánto tiempo el usuario deberá llevar a cabo los tests cognitivos: de forma diaria, semanal o mensual.

Mediante el análisis de la respuesta proporcionada por el cuidador y considerando la fecha en la que se llevó a cabo el último test del usuario, se efectúa un cálculo para determinar la nueva fecha para la realización de la próxima prueba. Así, cuando el usuario accede a la aplicación, se calcula si la fecha actual es igual o posterior a la fecha del próximo test establecido. Si es así, antes de habilitar el acceso a los minijuegos, se requiere que primero complete el test correspondiente. Cuando el usuario termina el test, se calcula la nueva fecha. En el Anexo F: Detalle del cálculo y establecimiento de la periodicidad de los tests, se explica con más detalle el cálculo y establecimiento de la periodicidad.

c) Periodicidad y tópico de los recordatorios

La aplicación permite al cuidador decidir la periodicidad de los recordatorios y el tópico de los mismos. Para ello establece avisos para alertar al usuario de la acción que debe realizar.

Desde el *frontend* la aplicación pregunta al usuario por la acción que quiere recordar y cada cuánto quiere que se realice el recordatorio. Cuando el usuario ha introducido estos datos, se envían al *backend*, donde se guardan en los atributos de sesión y posteriormente en la base de datos. Si el cuidador desea modificar dichos valores, puede hacerlo diciendo “establecer recordatorio”.

El proceso para establecer la periodicidad es igual al explicado en el apartado anterior “Periodicidad de los test”.

d) Recompensas de minijuegos

El cuidador es quien determina la cantidad de puntos que debe alcanzar el usuario para lograr la recompensa, el tiempo que tiene para conseguir los puntos y la recompensa que recibe el usuario cuando alcanza los puntos necesarios. La bonificación es totalmente configurable, permitiendo al cuidador definir la acción según sus preferencias. Por ejemplo, el cuidador puede establecer como recompensa ir a pasear, tomar un chocolate, hacer una videollamada o cualquier otra acción que considere oportuna. Cuando el usuario alcanza la puntuación necesaria para conseguir

la recompensa, el sistema comunica cuál es el premio y le indica al usuario que hable con su cuidador para recibirla.

El proceso para establecer la periodicidad es el mismo al explicado en el apartado "Periodicidad de los tests".

4.2.3. Visualización de datos

Adicionalmente a establecer parámetros de configuración, el cuidador también puede visualizar y escuchar datos que han sido almacenados mientras el usuario utiliza la aplicación.

Por un lado, como ya se ha mencionado anteriormente, el usuario realiza tests cognitivos de manera periódica. Tras estos, la aplicación comprueba con los datos almacenados en la base de datos si las respuestas han sido o no correctas, almacenando el número de respuestas correctas y la diferencia de respuestas correctas con respecto a la anterior vez que el usuario realizó el test. El cuidador puede acceder a estos datos para observar si el usuario progresa adecuadamente y podrá decidir en función a la evolución del usuario la periodicidad de los tests, y modificarla si lo cree oportuno.

Además, la aplicación almacena en la base de datos la cuenta de las veces que el usuario juega a cada minijuego en el mes actual, dato al cual también puede acceder el cuidador. Para implementar esta funcionalidad se definieron tres variables, una para llevar la cuenta del uso de cada minijuego y se guardan en los atributos de sesión durante toda la sesión del usuario y según este va realizando minijuegos, las variables se incrementan en uno. Cuando el usuario abandona la sesión (es decir, cierra la aplicación), las variables se hacen persistentes en base de datos. Se almacena también el mes actual de las cuentas, y si el último mes guardado no coincide con el mes actual, se ponen las variables a cero y se actualiza el mes; de esta forma cada mes se reinician las cuentas de cuántas veces ha jugado el usuario a cada minijuego.

Por último, la aplicación almacena también la puntuación que el usuario recibe en los distintos juegos que realiza (que aumenta conforme acierta más rondas del juego). Estos datos también están disponibles para el cuidador y se reinician periódicamente dependiendo de la frecuencia establecida por el cuidador para recibir las recompensas por la puntuación. La Figura 4.4 muestra la visualización de la puntuación del usuario.



Figura 4.4: Visualización de los puntos del usuario.

El objetivo principal de que el cuidador pueda acceder a toda esta información es que pueda decidir modificar ciertos parámetros, como la periodicidad de las recompensas o de los tests, el tipo de recompensas para motivar más al usuario, etc. , en función de la evolución del uso de los juegos y los tests por parte de la persona mayor.

4.3. Componentes de usuario

En esta sección se explican los minijuegos con los que cuenta la aplicación para que el usuario pueda entrenar la memoria.

Como se ha mencionado previamente, la aplicación permite al usuario participar en una serie de minijuegos para entrenar la memoria de una forma entretenida. En un principio se implementan tres juegos distintos, pero la aplicación está preparada para soportar la incorporación de más minijuegos de manera rápida y sencilla sin tener que aplicar grandes modificaciones a su estructura. En el Anexo [G](#): Inclusión de nuevos minijuegos en la aplicación, se presenta de forma detallada cómo añadir nuevos minijuegos.

La lógica de los juegos implementados se explica en el Anexo [H](#): Detalle de los minijuegos implementados. Sin embargo, en esta sección se procede a explicar un elemento externo utilizado para la implementación del juego de palabras encadenadas. Se explica únicamente este elemento para no recargar la memoria; separando la lógica de los minijuegos explicada en el anexo mencionado, de la implementación y uso del elemento externo que se explica a continuación.

El juego de palabras encadenadas es un juego multijugador en el cual el dispositivo dice al usuario una palabra, captura posteriormente la palabra que el usuario responde y si esta comienza por la misma letra que la palabra anteriormente dicha acaba, el usuario ha acertado.

Para generar una palabra aleatoria y comprobar que las palabras que el usuario dice son gramaticalmente correctas, se hace uso de una API externa llamada Datamuse API [3], que es un motor de búsqueda de palabras para desarrolladores. Esta API tiene una amplia gama de funcionalidades, entre las cuales existe la posibilidad de generar palabras aleatoriamente y comprobar la corrección de palabras. La implementación de estas funcionalidades se lleva a cabo mediante el uso de dos funciones, una para generar una palabra aleatoria (Figura 4.5) y otra para comprobar la corrección gramatical de una palabra (Figura 4.6). Si se observa la función de la Figura 4.5, se puede comprobar que se lleva a cabo una petición de tipo GET a la base de datos de Datamuse, especificando los parámetros del idioma y el número de palabras a recuperar; idioma español y una palabra en este caso.

```
const getRandomSpanishWord = async () => {
  try {
    const response = await axios.get(DATAMUSE_API_BASE_URL, {
      params: {
        sp: 'es', // Especifica el idioma español
        max: 1, // Solicita solo una palabra aleatoria
      },
    });
    return response.data[0].word;
  } catch (error) {
    console.error('Error fetching random Spanish word:', error);
    return null;
  }
};
```

Figura 4.5: Obtención de palabra aleatoria con la API de Datamuse.

En cuanto a la función mostrada en la Figura 4.6, se puede observar que se realiza una petición de tipo GET de una palabra concreta del idioma español (esto se consigue añadiendo 'es' delante de la palabra, para que la API sepa de qué idioma viene la misma). Si la petición devuelve la palabra de manera exitosa, quiere decir que la palabra introducida (que es la que el usuario ha proporcionado) es correcta y existe en el idioma establecido; si por el contrario la búsqueda no encuentra la palabra, el juego no admitirá la respuesta del usuario aunque comience por la misma letra por la que acaba la palabra anterior nombrada, pues no se permite al usuario utilizar palabras inventadas, permitiendo que el juego sea más competitivo y complejo.

```

const doesSpanishWordExist = async (word) => {
  try {
    const response = await axios.get(DATAMUSE_API_BASE_URL, {
      params: {
        sp: `es:${word}`, // Agregamos 'es:' al inicio de la palabra
      },
    });
    return response.data.length > 0;
  } catch (error) {
    console.error('Error checking word existence:', error);
    return false;
  }
};

```

Figura 4.6: Corrección gramatical con la API de Datamuse.

Capítulo 5

Evaluación

En este apartado se detallan las pruebas con usuarios que se han llevado a cabo tras la implementación de la aplicación. En primer lugar, se presenta la metodología seguida en las pruebas, luego se explican los resultados obtenidos y el análisis de los mismos.

5.1. Metodología

El objetivo de la evaluación consiste en comprobar el correcto funcionamiento de la aplicación, y su utilización por parte de los usuarios.

Para ello, se ha decidido realizar la evaluación con dos pares de cuidador - persona mayor.

Para que los usuarios objeto de la evaluación pudieran comprender el objetivo de la aplicación, se les ha explicado lo siguiente “La aplicación sirve para que las personas mayores entrenen su memoria en casa de manera sencilla y relajada, para ello se les permite jugar a tres minijuegos diferentes y hacer tests cognitivos de manera periódica. Además, la aplicación permite al cuidador de la persona mayor configurar aspectos de la pantalla, de los juegos y de los tests, así como consultar resultados de los mismos.” La evaluación consta de tres partes principales.

- La primera parte se corresponde con un cuestionario previo a la utilización de la aplicación para recoger el perfil y las características específicas del usuario que pueden influir e interferir en el uso de la aplicación.
- La siguiente parte se corresponde con el uso de la aplicación en la que se encomienda a los usuarios diferentes tareas, según sean usuarios finales o cuidadores. Al usuario mayor se le pide jugar a los tres minijuegos disponibles en la aplicación. A dos de ellos juega él solo, y al juego de las palabras encadenadas, al ser multijugador, juega también el cuidador. El cuidador también debe completar

tareas del perfil del cuidador, más concretamente debe cambiar el modo de la pantalla a modo oscuro y consultar el resultado del último test que realizó el usuario. En esta parte se aplica el método *thinking aloud* [18] que consiste en que el/los usuarios van expresando sus opiniones al interactuar con la aplicación y el evaluador observa y registra cualquier eventualidad.

- La última parte consiste en una entrevista semiestructurada individual a cada participante. La entrevista se lleva a cabo una vez terminada la parte de interacción con la aplicación. En esta entrevista se realizan preguntas sobre la dificultad a la hora de completar las tareas y se les pide sus opiniones sobre posibles mejoras, sugerencias, etc. De esta manera se podrán sacar conclusiones bastante completas de la evaluación.

5.2. Resultados y análisis

En esta sección se recogen los resultados obtenidos después de que utilizaran la aplicación dos pares cuidador-usuario, presentando el perfil de los usuarios encuestados, las respuestas a las preguntas realizadas en la entrevista y las conclusiones obtenidas tras analizar las respuestas.

Los perfiles de los usuarios encuestados se recogen en la tabla 5.1.

Identificador	Nombre	Edad	Rol
Usuario 1	M ^a Pilar	90	Persona Mayor
Cuidador 1	Reyes	53	Cuidador
Usuario 2	José	75	Persona Mayor
Cuidador 2	María	26	Cuidador

Tabla 5.1: Usuarios finales partícipes de la evaluación

Las preguntas del cuestionario previo están orientadas a determinar el grado de uso y aceptación de las tecnologías, especialmente de las personas mayores. Se presentan en las Figuras 5.1 y 5.2 junto con las respuestas de los usuarios finales y de los cuidadores respectivamente. Cabe destacar que las preguntas son con opciones para facilitar la tarea de clasificación.

A la vista de las respuestas obtenidas por los usuarios en este primer cuestionario, se puede observar que principalmente las personas mayores no hacen un gran uso de la tecnología y no se encuentran del todo cómodos con el uso de dispositivos electrónicos. Además, algunas de ellas sufren problemas en vista y oídos, por lo que en el estudio se presta especial atención a las dificultades que puedan surgir con la interacción del usuario y el dispositivo.

ID de la pregunta	Pregunta	Respuesta
Q1	¿Padece de problemas visuales, auditivos o de habla?	Visuales: 2 Auditivos: 1 De habla: 0
Q2	¿Cuánto manejo de las tecnologías tiene?	Alto: 0 Medio: 1 Bajo: 0 Nulo: 1
Q3	¿Le tiene miedo a las tecnologías?	Alto: 0 Medio: 0 Bajo: 1 Nulo: 1
Q4	¿Se maneja con el teléfono móvil?	Alto: 0 Medio: 0 Bajo: 1 Nulo: 1
Q5	¿Para qué utiliza más el teléfono móvil? ¿WhatsApp, solo llamadas, jugar a juegos?	WhatsApp: 0 Llamadas: 2 Jugar: 0
Q6	¿Tiene algún otro dispositivo? ¿Cual?	Sin otro dispositivo: 1 Tablet: 0 Ordenador: 1 Alexa: 0

Figura 5.1: Cuestionario previo al estudio realizado a los usuarios.

ID de la pregunta	Pregunta	Respuesta
Q1	¿Padece de problemas visuales, auditivos o de habla?	Visuales: 1 Auditivos: 0 De habla: 0
Q2	¿Cuánto manejo de las tecnologías tiene?	Alto: 1 Medio: 1 Bajo: 0 Nulo: 0
Q3	¿Le tiene miedo a las tecnologías?	Alto: 0 Medio: 0 Bajo: 1 Nulo: 1
Q4	¿Se maneja con el teléfono móvil?	Alto: 1 Medio: 1 Bajo: 0 Nulo: 0
Q5	¿Para qué utiliza más el teléfono móvil? ¿WhatsApp, solo llamadas, jugar a juegos?	WhatsApp: 2 Llamadas: 1 Jugar: 0
Q6	¿Tiene algún otro dispositivo? ¿Cual?	Sin otro dispositivo: 0 Tablet: 0 Ordenador: 2 Alexa: 2

Figura 5.2: Cuestionario previo al estudio realizado a los cuidadores.

Las preguntas realizadas a los cuidadores y sus respuestas de la entrevista realizada tras la interacción con la aplicación están recogidas en la Figura 5.3. Por otro lado, las preguntas realizadas a los usuarios mayores y sus respuestas se recogen en la Figura

5.4.

Además de las preguntas presentes en las tablas, se les preguntó a los usuarios si tenían otras ideas o sugerencias.

ID de la pregunta	Pregunta	Respuesta
Q7	¿La aplicación te resulta fácil de usar?	Muy fácil: 0 Fácil: 2 Media: 0 Difícil: 0 Muy difícil: 0
Q8	¿La aplicación te parece fácil de usar para los mayores?	Muy fácil: 0 Fácil: 1 Media: 1 Difícil: 0 Muy difícil: 0
Q9	¿La aplicación ayuda a tener control de evaluación cognitiva del usuario?	Alto: 2 Medio: 0 Bajo: 0 Nulo: 0
Q10	¿Las recompensas ayudan al usuario?	Alto: 1 Medio: 1 Bajo: 0 Nulo: 0
Q11	¿Tienes alguna recomendación sobre el apoyo visual?	Cuidador 1: mensajes más cortos para que el usuario no tenga que leer mucho. Cuidador 2: mostrar mensajes más claros para que el usuario sepa bien cómo interactuar.

Figura 5.3: Encuesta realizada a los cuidadores y sus respuestas.

ID de la pregunta	Pregunta	Respuesta
Q12	¿Qué juego te resulta más fácil de usar?	Secuencia de colores: 0 Palabras encadenadas: 2 Completar refranes: 0
Q13	¿Qué juego te resulta más difícil de usar?	Secuencia de colores: 0 Palabras encadenadas: 0 Completar refranes: 2
Q14	¿El apoyo visual ayuda o daba problemas?	Muy bien: 0 Bien: 1 Medio: 1 Mal: 0 Muy mal: 0
Q15	¿Alexa te entendía cada vez que hablabas?	Muy bien: 0 Bien: 1 Medio: 1 Mal: 0 Muy mal: 0
Q16	¿Entiendes las instrucciones de Alexa?	Muy bien: 0 Bien: 1 Medio: 1 Mal: 0 Muy mal: 0

Figura 5.4: Encuesta realizada a los usuarios y sus respuestas.

Observando la Figura 5.3 se puede concluir que generalmente los cuidadores se sienten cómodos utilizando la aplicación (Q7) y sienten que les podría servir de ayuda para controlar el progreso cognitivo del usuario (Q9); pero presienten que en algún momento la aplicación podría resultar complicada para el usuario (Q8).

Los cuidadores recomiendan principalmente añadir recordatorios o algún elemento similar para las recompensas, para avisarles cuando los usuarios alcanzan el número de puntos con los cuales deberían obtener una recompensa. También hacen hincapié en la falta de claridad de los mensajes mostrados por la pantalla, explicando la necesidad de que sean más claros y concisos para el usuario.

Observando la Figura 5.4 se concluye que las personas mayores responden mejor ante juegos con apoyo visual (Q12 y Q13) y ante el juego multijugador (Q12 y Q13). Esto, según los mismos usuarios han contado durante la evaluación, se debe principalmente a que a veces no saben del todo cómo responder a lo que el dispositivo les pide y además se sienten más seguros y relajados si tienen al lado a una persona que maneja las tecnologías.

Un usuario mayor sugiere introducir un minijuego en el que se pueda interactuar con la pantalla de forma táctil, para reforzar el apoyo visual y no depender solo de la interacción por voz si se quedan atascados en una tarea.

Por tanto, siguiendo las recomendaciones de los usuarios, se ha revisado y modificado los mensajes mostrados por pantalla y su formato, para que el apoyo visual sea lo más útil posible para los usuarios.

Capítulo 6

Gestión del proyecto

En este capítulo se explica la planificación llevada a cabo para el desarrollo del trabajo, así como la metodología utilizada y la dedicación al proyecto.

6.1. Desarrollo del proyecto

El trabajo se divide en diferentes tareas.

La primera tarea que se realiza se corresponde con la investigación sobre las aplicaciones existentes similares a la que se implementa y sobre el desarrollo de aplicaciones por voz.

Tras realizar esta tarea, comienza la definición y acotación de las funcionalidades y requisitos del proyecto.

Una vez están claras las funcionalidades a incluir en la aplicación, se procede a implementar las mismas. Primero se implementa la interacción por voz y la lógica de las funcionalidades y, posteriormente, el apoyo visual para los dispositivos con pantalla. Durante el transcurso de la implementación se realizan pruebas unitarias para comprobar el correcto funcionamiento de las distintas funcionalidades. Aunque cabe destacar que el proceso de investigación sobre la implementación de aplicaciones por voz continúa durante esta etapa.

Por último, una vez la aplicación está terminada, se realizan las pruebas con usuarios. Para mostrar de manera visual cómo transcurren las distintas tareas en el tiempo, se incluye un diagrama de Gantt en la Figura [6.1](#).

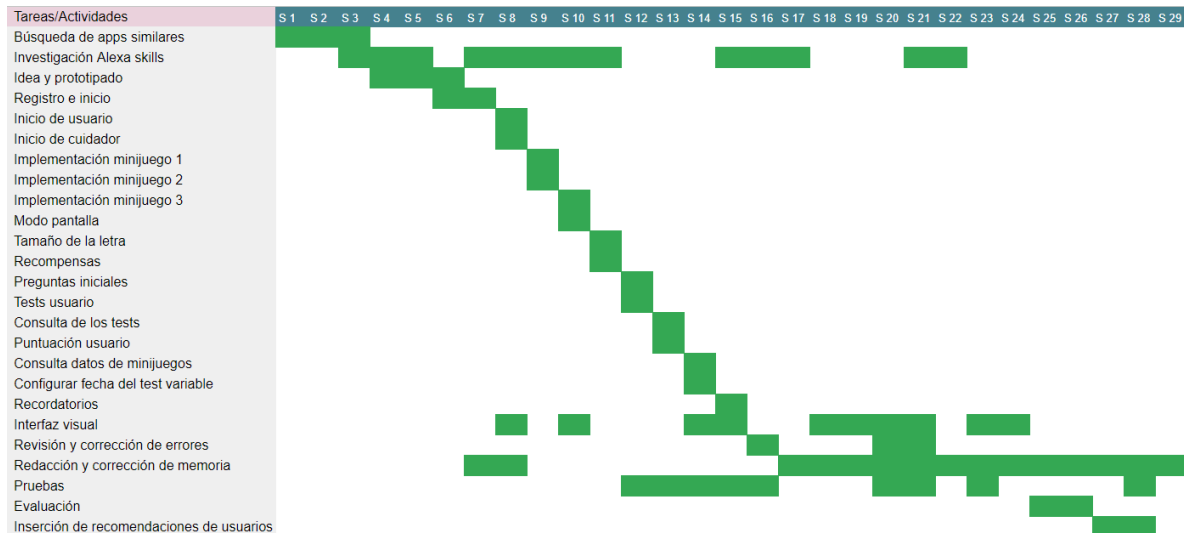


Figura 6.1: Diagrama de Gantt del desarrollo del proyecto.

Para la realización del proyecto se ha seguido una metodología ágil con UX, organizado las tareas mediante *sprints*. Así, cada tarea se realiza en un corto período (de unos 2 o 3 días) y se van terminando las tareas completas. De esta manera, se puede hacer pruebas de las distintas funcionalidades sin necesidad de tener la aplicación completa terminada.

6.2. Dedicación

La dedicación en tiempo al proyecto se ha recogido en un documento Excel, donde se presentan las horas invertidas en el proyecto separadas por tipos de tareas y con una descripción específica de la tarea concreta realizada cada día en el horario especificado (Figura 6.2).

Semana_3	3/03/2023	13:00:00	13:30:00	Reuniones	Decisión de búsqueda de apps similares
Semana_4	8/03/2023	10:00:00	12:00:00	Búsqueda de información	Búsqueda de aplicaciones similares
Semana_4	12/03/2023	12:30:00	13:30:00	Búsqueda de información	Prueba de skills de Alexa similares

Figura 6.2: Ejemplo del registro y seguimiento de las horas y tareas realizadas.

Además, dicho documento contiene la suma de horas totales dedicadas al proyecto y las horas dedicadas a cada tipo de tarea, como se muestra en la Figura 6.3.

<i>Tipo</i>		Pilar Ester
✓ Pruebas		15:54:00
📅 Reuniones		6:20:00
💻 Back-end		95:25:00
📄 Memoria		111:50:00
📱 Front-end		37:51:00
🔍 Búsqueda de información		36:15:00
💡 Idea y prototipado		3:00:00
Suma total		306:35:00

Figura 6.3: Horas totales y por tareas dedicadas al proyecto.

Se puede observar que se han dedicado un total de 306 horas y 35 minutos al desarrollo del trabajo de fin de grado.

Capítulo 7

Conclusiones y trabajo futuro

En este último capítulo se explican las conclusiones extraídas del proyecto realizado, la valoración personal y las posibles tareas para continuar con su desarrollo en el futuro.

7.1. Conclusiones

Se han cumplido e implementado los objetivos establecidos en el primer capítulo del proyecto.

- Se han estudiado aplicaciones similares a la implementada para poder comprender las necesidades del usuario y averiguar qué existe en el mercado.
- Se han estudiado las posibles tecnologías a utilizar para el desarrollo de la aplicación; eligiendo las herramientas y tecnologías más apropiadas y cómodas para el desarrollador para llevar a cabo la implementación del sistema.
- A partir de los estudios anteriores, se han definido los requisitos de la aplicación y la arquitectura de la misma para establecer límites claros del trabajo, facilitando el posterior desarrollo de la aplicación.
- Se ha desarrollado una aplicación por voz con apoyo visual para dispositivos Alexa Echo y Alexa Echo Show que permite al usuario entrenar su memoria realizando minijuegos y tests cognitivos periódicos. También permite al cuidador del usuario configurar aspectos de la aplicación y obtener resultados sobre el entrenamiento del usuario.
- Se ha llevado a cabo una evaluación con un grupo reducido de usuarios que ha permitido verificar el desempeño de la aplicación y detectar puntos de mejora y posibles líneas de trabajo futuro. La aplicación ha probado ser usable y contar con un funcionamiento bastante satisfactorio para el usuario, aunque se puedan realizar ciertas mejoras principalmente en la parte del apoyo visual.

7.2. Valoración desde el punto de vista personal

Este trabajo me ha servido para investigar sobre las aplicaciones con interacción por voz, sus limitaciones y sus ventajas ante otras aplicaciones de interfaz visual. Sin embargo, hay que aclarar que el desarrollo de este proyecto me ha implicado un coste añadido del aprendizaje de las tecnologías y métodos para implementar aplicaciones por voz para dispositivos Alexa [22], [10], [23], un campo que todavía no había estudiado hasta el momento y por lo que me supuso un gran esfuerzo de investigación y pruebas iniciales para familiarizarme con el entorno.

Pese a esta dificultad, he aprendido mucho sobre este tema y el resultado de la aplicación desarrollada es bastante satisfactorio.

Además, el trabajo me ha ayudado a comprender las limitaciones de las personas mayores y cómo solventarlas, sobre todo tras la evaluación con usuarios finales, viendo las respuestas obtenidas por los usuarios y los aspectos que habría que mejorar de la aplicación. También me ha permitido observar la importancia de adaptar las aplicaciones al usuario final, en este caso personas mayores, y la importancia de establecer aspectos configurables en la aplicación para que la experiencia sea lo más personal posible.

En general, se puede concluir que se han conseguido cumplir los objetivos del proyecto de manera adecuada y se espera que esta aplicación pueda ayudar a la población mayor a entrenar su memoria sin encontrar dificultades en la interacción con las tecnologías.

7.3. Trabajo futuro

Tras finalizar este trabajo se observa que de cara a futuras implementaciones de la aplicación, sería conveniente mejorar algunos aspectos que los usuarios de la aplicación han comentado en la evaluación, como incluir mensajes de ayuda más representativos para que los usuarios sepan como activar las acciones correctas y mejorar el *feedback* con el usuario. Para ello sería conveniente realizar pruebas con un mayor número de usuarios finales y cuidadores para obtener datos más significativos.

Otra posible mejora consistiría en permitir añadir más usuarios para que utilicen la aplicación. Por el momento solo se plantea la aplicación para un usuario y un cuidador, pero se podría adaptar la aplicación para permitir configurar más usuarios y cuidadores en un mismo dispositivo.

Por otro lado, sería interesante añadir más minijuegos para aumentar la diversidad y

usabilidad de la aplicación. Esta mejora no supondría demasiado trabajo de desarrollo, ya que se ha dejado preparada la aplicación, tal y como se explica en el Anexo [G](#): Inclusión de nuevos minijuegos en la aplicación.

Por otro lado, sería interesante permitir al cuidador elegir qué test debe realizar el usuario; hasta el momento solo se plantea un test posible (como ejemplo de test cognitivo) pero hay otras opciones de tests para utilizar que el usuario podría realizar. También se estudia la posibilidad de añadir personajes virtuales y enlazar la aplicación con Spotify para poner música al usuario.

Por último, otras extensiones futuras de esta aplicación podrían consistir en implementar una aplicación móvil que permitiese al cuidador subir fotos y vídeos, para poder mostrárselas posteriormente al usuario; por ejemplo, foto de las medicinas que tiene que tomar y mostrarlas con el recordatorio de tomar medicinas, vídeo de familiares felicitándole y mostrarlo cuando termine un minijuego, etc. Esta aplicación podría incluir la capacidad de enviar mensajes SMS o por correo electrónico al cuidador.

Glosario

atributos de sesión son datos o información que una Skill específica puede almacenar y utilizar durante una interacción de usuario en una sesión activa. Una sesión se inicia cuando un usuario interactúa con una Skill de Alexa y continúa hasta que la Skill completa su tarea o se cierra.. [12](#), [72](#)

attributes manager expone atributos que se pueden recuperar y actualizar en los *handlers*. [63](#)

handler procesan eventos que suceden en el backend. Cada *handler* se encarga de manejar un evento concreto y tienen dos funciones: *canHandle* y *handle*. [12](#), [13](#), [69](#), [72](#)

handler Input es un elemento que contiene información importante sobre la solicitud recibida y proporciona acceso a elementos importantes de la aplicación.. [63](#)

intent representan las acciones que cumplen con las solicitudes demandadas por el usuario. Las intenciones se precisan en una estructura de JSON llamada esquema de interacción [\[9\]](#).. [12](#), [13](#), [21](#), [61](#), [69](#), [71](#), [72](#)

interceptor es una función que se ejecuta entre medio de un *request* o de un *response* para hacer algo. Más adelante veremos algún ejemplo de dónde se utiliza.. [13](#), [62](#)

slot contienen datos que el usuario proporciona en la interacción con el dispositivo y que pueden variar de valor. Hay distintos tipos de *slots* proporcionados por Alexa y también el propio programador puede crear sus propios tipos de *slots* [\[9\]](#).. [72](#)

utterance son las posibles frases que el usuario puede decir, que están asociadas a las intenciones. Incluye tantas frases representativas como sea posible. [\[9\]](#). [12](#), [61](#), [69](#)

Voice User Interface son las interfaces que permiten interactuar con el usuario a través de comandos de voz. Estas interfaces no requieren contacto visual ni manual. [\[20\]](#). [17](#)

Bibliografía

- [1] Alexa skills kit sounds library. URL: <https://developer.amazon.com/en-US/docs/alexa/custom-skills/ask-soundlibrary.html>.
- [2] Chatgpt. URL: <https://chat.openai.com/>.
- [3] Datamuse api. URL: <https://www.datamuse.com/api/>.
- [4] Las 12 mejores apps para que los mayores ejerciten su mente. *Familiados*, 2020. URL: <https://blog.familiados.com/las-12-mejores-apps-para-ejercitar-la-mente-de-nuestros-mayores/>.
- [5] Amazon. Alexa developer console. URL: <https://developer.amazon.com/alexa/console/ask>.
- [6] Amazon. Alexa developer documentation. URL: <https://developer.amazon.com/en-US/docs/alexa/documentation-home.html>.
- [7] Amazon. Características de amazon dynamodb. *AWS Amazon*. URL: <https://aws.amazon.com/es/dynamodb/features/>.
- [8] Amazon. Características de amazon s3. *AWS Amazon*. URL: <https://aws.amazon.com/es/s3/features/>.
- [9] Amazon. Create the interaction model for your skill. URL: <https://developer.amazon.com/en-US/docs/alexa/custom-skills/create-the-interaction-model-for-your-skill.html>.
- [10] Amazon. Curso de alexa skills: Diseño de voz. URL: <https://build.amazonalexadev.com/curso-diseno-de-voz.html>.
- [11] Amazon. ¿qué es la opción personalizar skills del id de voz de alexa? URL: <https://www.amazon.es/gp/help/customer/display.html?nodeId=GF4LMSZG6SUFHQUZ>.

- [12] Amazon. Speechcon reference (interjections): Spanish (es), 2023. URL: <https://developer.amazon.com/en-US/docs/alexa/custom-skills/speechcon-reference-interjections-spanish.html>.
- [13] Laura Bonzano Claudio Vassallo Mario Alberto Battaglia Gianluigi Mancardi Marco Bove Giampaolo Brichetto Andrea Tacchino¹, Ludovico Pedullà. A new app for at-home cognitive training: Description and pilot testing on patients with multiple sclerosis. *JMIR MHEALTH AND UHEALTH*, 2015.
- [14] E. Paul Cherniack. Not just fun and games: applications of virtual reality in the identification and rehabilitation of cognitive disorders of the elderly, disability and rehabilitation: Assistive technology. 2011. URL: <https://doi.org/10.3109/17483107.2010.542570>.
- [15] Dustin A. Coates. *Voice Applications for Alexa and Google Assistant*. Manning Publications Co, Shelter Island, NYC, 2019.
- [16] Organización Mundial de la Salud. Envejecimiento y salud. *Organización Mundial de la Salud*, 2022. URL: <https://www.who.int/es/news-room/fact-sheets/detail/ageing-and-health#:~:text=Todos%20los%20pa%C3%ADses%20del%20mundo,en%202020%20a%201400%20millones>.
- [17] Pilar Frías. Experiencia de usuario en interfaces de usuario por voz. *Eunoia Digital*. URL: <https://play.google.com/store/apps/details?id=com.amazon.dee.app&hl=es&gl=US&pli=1>.
- [18] Toni Granollers. Pensando en voz alta (thinking aloud). URL: <https://mpiua.invid.udl.cat/pensando-en-voz-alta-thinking-aloud/>.
- [19] Ji Youn Shin Jina Huh-Yoo. Designing everyday conversational agents for managing health and wellness. *14th EAI International Conference on Pervasive Computing Technologies for Healthcare*, 2020.
- [20] Amazon Mobile LLC. Amazon alexa. *Playstore*. URL: <https://www.eunoia.es/blog/experiencia-usuario-en-interfaces-de-usuario-por-voz>.
- [21] Elena Sánchez Vega María del Carmen Rodríguez-Martínez, Emilio Ortega Fernández. Aplicación de nuevas tecnologías en personas mayores con trastorno cognitivo leve-moderado desde la terapia ocupacional. *INNOEDUCA. INTERNATIONAL JOURNAL OF TECHNOLOGY AND EDUCATIONAL INNOVATION*, 2017.

- [22] German Viscuso. Curso oficial para desarrollo de skills de alexa. *KeepCoding*. URL: <https://plataforma.keepcoding.io/p/curso-desarrollo-alexa-skills>.
- [23] German Viscuso. De cero a héroe. *Amazon*. URL: <https://developer.amazon.com/es-ES/blogs/alexa/alexa-skills-kit/2020/03/el-curso-de-desarrollo-de-skills-de-cero-a-heroe-se-expande-con-nuevos-videos>.
- [24] Minxia Du Lanshu Zhou Xuan Zhou. Use of mobile applications in post-stroke rehabilitation: a systematic review. *Topics in Stroke Rehabilitation*, 2018.

Lista de Figuras

2.1	Diagrama de casos de uso del sistema.	9
3.1	Interacción de la aplicación para Alexa Show con el usuario.	11
3.2	Arquitectura del sistema con componentes.	13
4.1	Pantalla inicial del perfil de usuario.	18
4.2	Estructura de un documento APL.	19
4.3	Estructura de un documento APL.	20
4.4	Visualización de los puntos del usuario.	25
4.5	Obtención de palabra aleatoria con la API de Datamuse.	26
4.6	Corrección gramatical con la API de Datamuse.	27
5.1	Cuestionario previo al estudio realizado a los usuarios.	31
5.2	Cuestionario previo al estudio realizado a los cuidadores.	31
5.3	Encuesta realizada a los cuidadores y sus respuestas.	32
5.4	Encuesta realizada a los usuarios y sus respuestas.	32
6.1	Diagrama de Gantt del desarrollo del proyecto.	36
6.2	Ejemplo del registro y seguimiento de las horas y tareas realizadas. . .	36
6.3	Horas totales y por tareas dedicadas al proyecto.	37
D.1	Estructura para manejar distintos idiomas.	62
D.2	Interceptor para detectar el idioma.	62
E.1	Configuración del tamaño de la letra.	66
E.2	Configuración de tamaño de letra en archivos JSON.	66
F.1	Configuración de la periodicidad de las acciones.	67
H.1	Generación de un número aleatorio	71

Lista de Tablas

2.1	Características más relevantes de las aplicaciones estudiadas	6
5.1	Usuarios finales partícipes de la evaluación	30
A.1	Aplicaciones similares estudiadas	55

Anexos

Anexo A

Detalle de aplicaciones similares estudiadas

Para buscar aplicaciones que pueda resultar interesante estudiar previamente a la implementación de la aplicación, se siguen varios criterios de búsqueda.

Por una parte, se buscan aplicaciones para los dispositivos Alexa [14] (con y sin pantalla), para poder ver ideas de cómo es la interacción con el dispositivo, qué uso se hace de la pantalla y de la voz y qué puntos débiles tienen la mayoría de aplicaciones de este dispositivo, para intentar evitarlos.

Por otro lado, se realiza una búsqueda sobre aplicaciones con intenciones similares a la aplicación del proyecto; aplicaciones con minijuegos de memoria de distintos tipos; se observa cuáles pueden resultar más fáciles para ser realizados por voz, cuáles requieren de un soporte visual y qué juegos son los más entretenidos para usuarios, teniendo en cuenta que sean juegos distintos entre sí. En este Anexo se explica cuáles son las aplicaciones más significativas e interesantes que se investigan y prueban durante este proceso.

- *Mayores Activos*: aplicación de Alexa Echo que hace preguntas al usuario para ver si las respuestas del mismo son consistentes en las distintas ocasiones que pregunta. Además de hacer preguntas personales, también hace preguntas sobre secuencias (meses del año, estaciones, etc.), de la actualidad (día de hoy, estación, año), con palabras (por ejemplo: animal que empiece por la letra H), gimnasia mental con números (sumas y restas), preguntas de cultura general, etc. Esta es la aplicación ya existente que más se asemeja a la planteada en el proyecto, con la diferencia de que Mayores Activos no es una aplicación para dispositivos Alexa con pantalla o Alexa Echo Show.
- *Reto memoria*: aplicación de Alexa Echo para ejercitar la memoria. Esta aplicación consiste en 10 niveles en los cuales el asistente dice una lista de colores que aumenta con cada nivel y el usuario debe repetir la lista en el mismo orden.

- *Pirámide de la memoria*: aplicación de Alexa Echo similar a la anterior pero en vez de ser una lista de colores, el asistente dice en cada nivel una lista de palabras (creciente conforme pasa de nivel) que el usuario repite en orden para superar el nivel.
- *Aplicación para rehabilitación cognitiva en casa*: esta aplicación consta de varios ejercicios para trabajar distintos campos de la memoria como la memoria visual, memoria matemática con ejercicios de sumas, números mayores o menores, etc. Una vez el usuario realiza estos ejercicios, se procesan los resultados que este ha obtenido. Con algoritmos adaptativos de la carga de trabajo se definen los parámetros de los ejercicios adaptados según las capacidades del usuario para el siguiente intento. En cada intento se procesan los resultados para poder ir adaptando la dificultad de los ejercicios.
- *Fit Brain Trainers*: es una aplicación para dispositivos móviles desarrollada por un equipo de neurocientíficos para mejorar y entrenar el rendimiento del cerebro. La aplicación cuenta con ejercicios de memoria para distintos ámbitos como el *memory*, recordar una secuencia de colores, recordar qué figuras estaban antes y cuales no, etc.
- *Lumosity*: al iniciar la aplicación de móvil pregunta por la edad del usuario y este tiene que realizar una prueba de nivel para ver si está en la media de las respuestas del resto de usuarios en su rango de edad. Una vez realizada dicha prueba, se puede acceder a diversos juegos de atención, memoria, velocidad, resolución de problemas, etc. Conforme el usuario va realizando juegos, va desbloqueando nuevos juegos, evitando un entrenamiento repetitivo.
- *Focus*: es una aplicación para teléfono móvil muy completa que cuenta con juegos de memoria, de razonamiento, de atención, de coordinación y visual-espaciales que ayudan a reforzar tanto la memoria mental como la motriz.

Se muestra la tabla ?? con la comparativa de las aplicaciones estudiadas.

En el estudio realizado se estudian más aplicaciones, como las aplicaciones presentes en el blog de Familiados [4] pero solo se reflejan en la memoria las más representativas y relevantes. Como conclusión del estudio se destaca la escasez de aplicaciones configurables en el mercado y la importancia de permitir al usuario ajustar ciertos componentes para hacer su experiencia más personal; así como la falta de aplicaciones con interacción por voz y apoyo visual conjuntamente. Adicionalmente, la mayoría de las aplicaciones solo guardan una pequeña cantidad de datos.

Nombre	Móvil o Alexa	Tipo de juegos	¿Guarda datos?	¿Es configurable?	¿Recompensa para usuario?
Mayores activos	Alexa	Preguntas	Sí	No	No
Reto memoria	Alexa	Repetir colores	No	No	No
Pirámide de la memoria	Alexa	Repetir palabras	No	No	No
Rehabilitación cognitiva	Móvil	Memoria	Sí	No	Sí
Fit Brain	Móvil	Memoria	Sí	No	Sí
Lumosity	Móvil	Memoria	Sí	No	Sí
Focus	Móvil	Mentales y motrices	Sí	No	Sí

Tabla A.1: Aplicaciones similares estudiadas

Anexo B

Estudio de la base de datos - Amazon S3 frente a DynamoDB

A continuación se presentan las ventajas e inconvenientes de las dos bases de datos estudiadas. Para la redacción de este capítulo se ha contado con la ayuda de los sitios web [8], [7] y [2]

Amazon S3 La base de datos Amazon S3 es la finalmente escogida para el desarrollo del proyecto, a continuación se detallan los aspectos más destacables de la misma. Ventajas:

- Escalabilidad: Es muy escalable y puede manejar grandes cantidades de datos.
- Almacenamiento de objetos: Permite almacenar y recuperar archivos multimedia (imágenes, vídeos y documentos).
- Durabilidad: Ofrece alta durabilidad de los datos con opción de replicación automática.
- Costos flexibles: Ofrece almacenamiento de coste variable; solo pagar por la cantidad de almacenamiento utilizado.

Inconvenientes:

- NoSQL limitado: S3 no es una base de datos NoSQL completa y no es adecuado para consultas complejas.
- Latencia variable: La latencia puede ser variable, por lo que no es la mejor opción para aplicaciones que requieren tiempos de respuesta muy bajos.
- No transaccional: S3 no proporciona soporte para transacciones y operaciones atómicas en los objetos.

DynamoDB A continuación se presentan las características de la base de datos DynamoDB. Ventajas:

- Escalabilidad automática: Conveniente para aplicaciones con demanda variable.
- Rendimiento constante: Rendimiento predecible y bajo latencia incluso en cargas de trabajo intensivas.
- Modelo de datos flexible: Admite diversos tipos de datos y esquemas flexibles.
- Consultas rápidas: Permite crear índices para consultas eficientes.
- Soporte para transacciones: Adecuado para aplicaciones que requieren operaciones atómicas.

Inconvenientes:

- Costos variables: DynamoDB puede volverse costoso en aplicaciones con alta carga de escritura o lectura.
- Esquema de datos cuidadoso: Hay que diseñar bien la estructura de los datos para evitar consultas ineficientes.
- No es óptimo para almacenamiento de objetos grandes: No es la mejor opción para el almacenamiento masivo de objetos grandes como lo es S3.

Anexo C

Detalle del módulo de sonidos especiales y de la librería de sonidos de Alexa

En este capítulo se detalla en qué consisten el módulo de sonidos especiales y la librería de sonidos de Alexa y se explica su utilización en la aplicación.

Módulo de sonidos especiales o *Speechcon*

Como ya se ha explicado, los *speechcons* son palabras y frases especiales y típicas de los distintos idiomas que Alexa pronuncia de una manera especial. Los *speechcon* son específicos para una serie de palabras determinadas, no se puede hacer que Alexa diga cualquier palabra con la entonación que se desea.

Para utilizarlos hay que acceder a la página de los mismos y buscar en la lista de audios aquella palabra que se quiera introducir en la aplicación.

Una vez localizada la frase que se quiere introducir, se declara una variable utilizando el elemento de marcado *say-as interpret-as* para indicar que el contenido dentro de él (en este caso la palabra 'crack') debe interpretarse como una interjección. De esta forma, se puede introducir esta variable en el output que se reproduce en respuesta al usuario y el dispositivo dirá la palabra con una entonación específica. En el caso de este trabajo, se utilizan palabras de ánimo cuando un usuario acierta en un minijuego.

Librería de sonidos de Alexa

Por su parte, la librería de sonidos *Alexa Sound Library* permite reproducir sonidos de muy distintos ámbitos para acompañar a las frases que reproduce el dispositivo. La forma de utilizar estos sonidos es utilizando la etiqueta */jaudio src/¿* e indicando en la misma el recurso (la url) del audio. Además de poder utilizar la gran variedad de sonidos disponibles en esta librería, se puede utilizar cualquier otro recurso de audio cuya url sea accesible desde la aplicación.

Para utilizarlos, se declara una variable con la etiqueta de audio src y se introduce en el output junto con la frase que debe decir el dispositivo. Así, el sonido se reproduce justo antes o justo después de la frase dependiendo de si se añade al principio o al final de la variable que contiene el texto.

En el proyecto se utilizan sonidos de la *Alexa Sound Library* para indicar que el usuario ha fallado o acertado en su respuesta a los minijuegos.

Anexo D

Extensión de la aplicación a distintos idiomas

En este Anexo se explica de manera detallada el método para extender la aplicación a distintos idiomas. La explicación es separada para el *backend* y el *frontend*.

Frontend

Trabajando con la plataforma *Alexa Developer Console*, el idioma en el que se desarrolla la aplicación en el *frontend* se configura en la creación de la misma, pero existe la posibilidad de añadir nuevos idiomas una vez la aplicación o *skill* de Alexa ha sido creada.

Para ello, en el apartado de *Build*, apartado donde está indicado el idioma que se ha configurado, se pueden añadir distintos idiomas. De esta manera, se pueden configurar las *utterances* de los *intents* en los idiomas seleccionados de manera independiente (separados uno y otro idioma). Así, el usuario podrá interactuar con la aplicación en cualquiera de los idiomas configurados.

A pesar de que esta es una solución correcta (y hasta el momento la única solución disponible para manejar distintos idiomas), es un poco costosa ya que hay que traducir e introducir manualmente en el esquema JSON del idioma añadido todas las frases que el usuario pueda decir en este nuevo idioma, lo que supone repetir este trabajo en cada idioma introducido.

Backend

Para el *backend* lo que se hace es detectar el idioma del dispositivo, y dependiendo de cuál sea, proveer un *strings* u otro (en un idioma u otro). Para simplificar esta tarea se hace uso de la librería *i18next* que permite proveer distintos *strings* dependiendo de las variables del idioma. Para usar esta librería hay que asegurar que la dependencia se encuentra en las dependencias del archivo *package.json*. Una vez hecho esto hay que

conseguir que *i18next* mapee a un idioma u otro los mensajes según el idioma local.

Además, hay que crear una estructura que contiene las frases en todos los idiomas que se quieren establecer como se muestra en la Figura D.1. La estructura cuenta con una subdivisión específica destinada a cada uno de los idiomas autorizados en el sistema. En la actual implementación de este proyecto, únicamente se encuentra habilitado el uso del idioma español ('es'). No obstante, gracias a este enfoque, se facilita de manera considerable la incorporación de nuevos idiomas de forma eficiente y sin complicaciones. Con este método perdura el problema de tener que reescribir las frases para cada idioma que se desee introducir, pero se realiza de una manera más controlada y acotada a un único fichero, lo cual resulta más cómodo y agiliza el proceso.

```
module.exports = {  
  es: {  
    translation: {  
      WELCOME_MSG: 'Bienvenido a cerebro en forma, donde podrás entrenar tu memoria.  
      HELLO_AGAIN_MSG: 'Bienvenido de nuevo a Cerebro en Forma. ¿Eres un cuidador? ¿  
      HELLO_CAREER_MSG: 'Bienvenido %s. Puedes modificar los ajustes de idioma "Configuración"
```

Figura D.1: Estructura para manejar distintos idiomas.

A continuación entran en juego los [interceptors](#), que permiten detectar el idioma local antes de empezar a trabajar y proveer un tipo de *strings* u otro dependiendo del idioma detectado. Para esto, se crea un [interceptor](#) de tipo *request* que se muestra en la Figura D.2.

```
LocalizationRequestInterceptor: {  
  process(handlerInput) {  
    const localizationClient = i18n.use(sprintf).init({  
      lng: handlerInput.requestEnvelope.request.locale,  
      fallbackLng: 'es',  
      overloadTranslationOptionHandler: sprintf.overloadTranslationOptionHandler,  
      resources: require('./localisation'),  
      returnObjects: true  
    });  
    const attributes = handlerInput.attributesManager.getRequestAttributes();  
    attributes.t = function (...args) {  
      return localizationClient.t(...args);  
    }  
  }  
},
```

Figura D.2: Interceptor para detectar el idioma.

La función correspondiente al [interceptor](#) del idioma, llamada *LocalizationRequestInterceptor*, inicializa una instancia de *i18next* con unos parámetros específicos.

- *lng*: el lenguaje; desde el [handler Input](#) se detecta el idioma local.
- *fallback*: idioma por defecto a utilizar si hay algún problema en la detección del idioma local.
- *overloadTranslationOptionHandler*: detecta si el *string* introducido tiene alguna variable (en forma de %s) y lo sustituye por el valor indicado.
- *resources*: de donde se obtienen las frases, en este caso se obtienen del fichero *./localisation*; donde se ubica la estructura que contiene las frases de los distintos idiomas.

Tras la inicialización del cliente *i18next* hay que obtener el valor de los *strings* dependiendo del idioma local. Primero se extraen los atributos de la petición de entrada a través del [attributes manager](#).

A los atributos obtenidos se les añade una función *t* que aplica el cliente instanciado para que seleccione, de acuerdo a los parámetros especificados en la instancia, el *string* correspondiente. De esta forma, ya se puede en cualquier momento llamar a la función *t* con el mensaje que se quiera de los definidos en la estructura de *strings* de idiomas, y se obtendrá el *string* en el idioma correspondiente.

Anexo E

Detalle de la configuración del modo y el tamaño de letra de la pantalla del dispositivo

A continuación se explica el proceso específico para el modo de pantalla y el tamaño de letra respectivamente.

- Modo de pantalla: Se implementa un proceso en el *backend* previo al envío de los datos a través del JSON. Durante este proceso, se verifica el valor del parámetro relacionado con el modo de pantalla, el cual puede ser ‘claro’ u ‘oscuro’. En función de esta determinación, se genera y envía el JSON correspondiente que incluye los ajustes de fondo de pantalla, color de letra y otros elementos pertinentes, acorde con el modo especificado.
- Tamaño de letra: la aplicación permite al cuidador establecer un tamaño pequeño, mediano o grande de letra. Según el valor indicado se aplican adaptaciones de tamaño a todos los tipos de texto disponibles, y posteriormente se incorporan como parámetros del JSON junto con el resto de parámetros, siguiendo el enfoque ilustrado en la Figura [E.1](#).

En la Figura [E.1](#), la variable *fontSize* contiene el valor del tamaño de la letra; 10, 15 o 20 dependiendo de si es pequeña, mediana o grande. Según el tipo de texto con el que se corresponden las distintas variables (texto principal, secundario, etc.) se le suma a ese valor un número u otro para ajustar sus tamaños. En el archivo JSON, los valores de las variables de tamaño de letra se establecen como los valores que se envían desde el *backend* como se muestra en la Figura [E.2](#).

```

handlerInput.responseBuilder.addDirective({
  type: 'Alexa.Presentation.APL.RenderDocument',
  version: '2023.1',
  document: constants.APLDocs.launchLight,
  datasources: {
    launchData: {
      type: 'object',
      properties: {
        headerTitle: requestAttributes.t('LAUNCH_HEADER_MSG'),
        mainText: name ? requestAttributes.t('LAUNCH_TEXT_FILLED_MSG') :
        hintString: name ? requestAttributes.t('LAUNCH_HINT_MSG') :
        textSizeBody: fontSize,
        textSizePrimary: fontSize + 10,
        textSizeSecondary: fontSize + 5,
        textSizeDetails: fontSize,
        textSizeSecondaryHint: fontSize
      },
      transformers: [{
        inputPath: 'hintString',
        transformer: 'textToHint',
      }]
    },
  },
});

```

Figura E.1: Configuración del tamaño de la letra.

```

"description": "Standard font sizes",
"dimensions": {
  "textSizeBody": "${payload.launchData.properties.textSizeBody}",
  "textSizePrimary": "${payload.launchData.properties.textSizePrimary}",
  "textSizeSecondary": "${payload.launchData.properties.textSizeSecondary}",
  "textSizeDetails": "${payload.launchData.properties.textSizeDetails}",
  "textSizeSecondaryHint": "${payload.launchData.properties.textSizeSecondaryHint}"
}

```

Figura E.2: Configuración de tamaño de letra en archivos JSON.

Anexo F

Detalle del cálculo y establecimiento de la periodicidad de los tests

La Figura F.1 presenta el código que configura la periodicidad.

Lo primero que se hace es establecer el día actual. Para ello, se utiliza la biblioteca *moment* de Javascript, que permite trabajar con fechas y calendarios. De esta manera se obtiene la fecha actual, calculando el principio del día del momento actual en la zona horaria obtenida previamente. Tras esto, se compara el valor del dato aportado por el cuidador, que se encuentra almacenado en la variable 'periodicidad' y puede tener el valor 'diariamente', 'semanalmente' o 'mensualmente'.

Dependiendo del valor de esta variable, se establece la nueva fecha de test como el día actual (calculado previamente) más un día, una semana o un mes mediante las funciones *clone* que clona el valor de la variable *today* y *add* que añade tantas unidades de tiempo como se indique en los parámetros de la misma.

A continuación, se almacena la variable en los atributos de sesión, y posteriormente en base de datos para poder recuperar la fecha en otras sesiones posteriores.

```
const today = moment().tz(timezone).startOf('day');
let nextTest;
if (periodicidad === 'diariamente') {
  nextTest = today.clone().add(1, 'day');
}
else if (periodicidad === 'semanalmente') {
  nextTest = today.clone().add(1, 'week');
}
else if (periodicidad === 'mensualmente') {
  nextTest = today.clone().add(1, 'month');
}
```

Figura F.1: Configuración de la periodicidad de las acciones.

Anexo G

Inclusión de nuevos minijuegos en la aplicación

Para añadir minijuegos a la aplicación se debe seguir el siguiente proceso.

En el *frontend* hay que crear un [intent](#) para activar el minijuego y otro para poder interactuar con el dispositivo durante el transcurso del juego. Hay que establecer las [utterances](#) que el usuario podrá decir para activar el juego y mandar los datos necesarios al *backend* almacenándolos en variables.

En el *backend*, se crea un [handler](#) para cada uno de los [intents](#) establecidos y en ellos se incluye la lógica del juego.

Por último, para mostrar el nuevo juego en la lista de minijuegos que se visualiza en la pantalla inicial del usuario, solamente hay que añadir la foto y texto correspondientes del minijuego a la lista de datos que se envían al archivo JSON correspondiente.

Anexo H

Detalle de los minijuegos implementados

Los minijuegos implementados para fomentar el entrenamiento de la memoria del usuario son los siguientes.

- Secuencia de colores: La aplicación dice un color y el usuario debe repetirlo, si este acierta el juego continúa y la aplicación añade un color a la secuencia, que el usuario deberá de repetir. El juego termina cuando el usuario falla. Los posibles colores a aparecer en la secuencia son Rojo, Amarillo, Azul y Verde, y para generar la secuencia, desde el manejador del `intent` correspondiente al juego de la secuencia de colores, se genera una secuencia aleatoria con los *strings* de dicho tipo de dato, que se va almacenando en una variable y se incrementa según el usuario va superando las rondas. Para generar dicha secuencia se utiliza la función especificada en la figura H.1, la cual genera un número aleatorio entre 0 y 1 con la función *random* de la biblioteca *Math*, la multiplica por cuatro y con la función *Math.floor* redondea el número obtenido a la baja, reduciendo los valores posibles a 0, 1, 2, o 3.

```
// Generate a random number between 0 and 3
const colorindex = Math.floor(Math.random() * 4);
```

Figura H.1: Generación de un número aleatorio

Cada número hará referencia a uno de los cuatro colores disponibles y se añadirá el color obtenido aleatoriamente a la secuencia guardada para decírsela al usuario en la siguiente ronda.

- Palabras encadenadas: este minijuego tiene el añadido de ser multijugador; al inicio del mismo se pregunta al usuario cuántos jugadores van a participar,

ajustando así los turnos del mismo al número de participantes que el usuario ha proporcionado. Para ello, en la interacción con el usuario se utiliza un `slot` de tipo `NUMBER` para poder recuperar desde el *backend* el dato del número de jugadores que el usuario diga. De esta manera, en el `handler` del `intent` correspondiente con este juego, se almacena en los `atributos de sesión` del usuario el número de jugadores que participan, y se irán realizando varios turnos hasta alcanzar dicho número. En ese momento, cuando todos los jugadores han participado, el juego termina. El juego comienza cuando la aplicación dice una palabra y cede el turno al primer jugador, y continúa hasta que todos los jugadores han participado. Si un jugador dice una palabra incorrecta, el turno pasa al siguiente jugador. Como idea de posible implementación futura poco costosa, podría añadirse a este minijuego la posibilidad de configurar no solo el número de jugadores sino también el número de rondas, para que sea el usuario quien decida cómo de largo hacer el juego.

- Completar los refranes: este tercer juego consiste en completar los refranes que la aplicación le dice al usuario. La aplicación escoge un refrán y dice la primera parte de este al usuario, que deberá terminar de decir el refrán completo. El juego termina cuando el usuario falla. Para conseguir los refranes, se ha creado una estructura de datos que contiene pares de *strings*, correspondientes con la primera y segunda parte de refranes típicos españoles. De esta manera, utilizando las funciones `Math.random` y `Math.floor` (como ya se ha explicado anteriormente) se consigue seleccionar uno de los pares de manera aleatoria. La aplicación dice la primera parte del refrán al usuario y cuando este responde comprueba que lo que el usuario ha dicho se corresponde con la segunda parte del refrán.