



Universidad
Zaragoza

Proyecto Fin de Carrera

Plataforma para la monitorización y análisis de experimentos sensomotores mínimos con EEG

Autor

Sergio Milla Pineda

Directores

Manuel González Bedia

Carlos Alquézar Baeta

Escuela de Ingeniería y Arquitectura

Año 2016

*A mis padres, por permitirme llegar hasta aquí,
a mi hermano, por estar siempre ahí,
a Manuel González Bedía, por su sabiduría teórica,
a Carlos Alquézar Baeta, por su sabiduría práctica,
a todo ISAAC, por su apoyo día a día.*

Resumen

En los ámbitos de simulación computacional y modelado cognitivo se está produciendo un creciente desplazamiento desde una perspectiva en la que se considera que las capacidades mentales se almacenan, procesan, ejecutan y desarrollan exclusivamente en el cerebro (la aproximación de la metáfora del ordenador) hacia una perspectiva en la que el enfoque recae sobre la interacción dinámica agente-entorno, centrando la atención sobre la fluidez del acoplamiento al mundo y la corporización de la acción como fundamentos de las capacidades cognitivas. A esta corriente se le denomina perspectiva sensomotora de la cognición y entiende que los procesos de reconocimiento y comprensión de nuestro entorno deben ser estudiados mediante modelos formales y experimentales que recojan la estructura dinámica del acoplamiento de un agente con su mundo a través de bucles de retroalimentación sensomotores (O'Regan & Noë 2001).

El enfoque sensomotor ha comenzado a desarrollarse en los últimos años desde diferentes disciplinas (a nivel neurocientífico y psicológico), pero no con la misma intensidad a nivel formal y computacional dentro del área de los sistemas inteligentes, donde se necesitan modelos mínimos que reduzcan la complejidad y la dimensionalidad de los sistemas de estudio, para facilitar su análisis y modelización. En este proyecto se ha desarrollado un entorno de experimentación que permite hacer análisis de capacidades sensomotoras simplificadas. Se trata de una plataforma virtual de interacción mínima en la que participan humanos y agentes software, en entornos de muy baja dimensionalidad, donde las interacciones sensomotoras entre sujetos y ambiente pueden registrarse con gran precisión, facilitando el estudio de las dinámicas de interacción.

El entorno virtual está diseñado para registrar, en los experimentos planteados, pautas sensomotoras y registros de la actividad cerebral de los sujetos mediante EEG (Electroencefalograma).

Índice de contenidos

1	Introducción	1
1.1	Motivación	1
1.2	Objetivos y Alcance	1
1.3	Contexto en el que se realiza el proyecto	1
1.4	Herramientas utilizadas	2
1.5	Organización de la memoria	3
2	Estado del arte	5
2.1	Estudio de la percepción: enfoque constructivo del espacio circundante	5
2.2	Leyes sensomotoras como medio para percibir	6
2.3	Cómo estudiar la generación de leyes sensomotoras en entornos experimentales: metodología mínima	8
3	Diseño de la plataforma: SensoMotor Lab.....	9
3.1	Esquema general.....	9
3.2	Entorno.....	10
3.3	Agente virtual: generación de comportamiento.....	11
3.4	Interacción del sujeto humano con el agente.....	15
3.4.1	Percepción.....	15
3.4.2	Acción	19
3.5	Definición de experimento: elementos y estructura temporal	20
3.6	Módulos de la plataforma	21
4	GUI.....	23
4.1	Menú principal	23
4.1.1	Experimento activo	24
4.2	Configuración	25
4.2.1	Navegación entre las pantallas de configuración	25
4.2.2	Configurar experimento	27
4.2.3	Configurar serie	28
4.2.4	Configurar trial	29
4.2.5	Configurar agente.....	31
4.2.6	Visualización y edición de la configuración.....	31
4.2.7	Formato del fichero externo de configuración	32
4.2.8	Estructura de directorios para almacenar los ficheros de configuración	32
4.3	Ejecución	33
4.3.1	Espera previa a ejecutar serie	33

4.3.2 Espera previa a ejecutar trial	33
4.3.3 Algoritmo de la función de actualización del entorno	35
4.4 Reproducción de experimentos	37
4.4.1 Algoritmo de la función de reproducción del entorno	38
5 Registro de datos.....	39
5.1 EEG	39
5.2 Evolución del estado del entorno	40
5.3 Estructura de directorios para el almacenamiento de los ficheros de captura	41
6 Diseño de experimentos con la plataforma	43
7 Conclusiones y trabajo futuro	45
Bibliografía	46
Anexo I: Casos de uso.....	49
Anexo II: Generación de GUI (pantallas y navegación)	51
Anexo III: fichero XML de configuración de experimento.....	52
Anexo IV: Diagrama de Gantt y tablas de tiempos	53

Índice de figuras

Figura 1: vehículo de Braitenberg	6
Figura 2: interacción sensoriomotora de un ser humano con un agente virtual	8
Figura 3: interacción sensoriomotora de un ser humano con un agente virtual dentro de la plataforma.....	9
Figura 4: el entorno contiene al sujeto (bola anaranjada) y al agente virtual (bola verde)	10
Figura 5: entorno lineal acotado	10
Figura 6: entorno toroidal	10
Figura 7: entorno continuo	10
Figura 8: ecuación de comportamiento del agente estático	11
Figura 9: ecuaciones que determinan el comportamiento del agente sinusoidal.....	11
Figura 10: relación entre la fase inicial y la dirección inicial en el agente sinusoidal.	11
Figura 11: ecuaciones que determinan el comportamiento del agente péndulo invertido	12
Figura 12: función [distancia angular]-[velocidad angular] y esquema de la trigonometría inherente al agente péndulo invertido	12
Figura 13: ecuaciones que definen el comportamiento del agente mano loca.....	13
Figura 14: percepción del agente por parte del sujeto humano	15
Figura 15: representación visual del entorno	15
Figura 16: representación visual normal del estado del entorno	16
Figura 17: representación visual invertida del estado del entorno	16
Figura 18: función distancia-volumen en percepción sonora normal	17
Figura 19: función distancia-volumen en percepción sonora invertida.....	17
Figura 20: ecuación para la percepción sonora gaussiana normal	18
Figura 21: ecuación para la percepción sonora gaussiana invertida	18
Figura 22: función distancia-volumen en percepción sonora gaussiana normal e invertida	18
Figura 23: acción del sujeto humano sobre el agente	19
Figura 24: ley física normal del entorno.....	19
Figura 25: ley física invertida del entorno.....	19
Figura 26: estructura temporal de la ejecución de una secuencia de trials	20
Figura 27: estructura temporal de la ejecución de un experimento con dos series.....	20
Figura 28: módulos principales de la plataforma.....	21
Figura 29: detalle de captura del menú principal	23
Figura 30: detalle de captura del menú principal sin el experimento activo seleccionado.....	24

Figura 31: sentido de navegación entre pantallas de configuración de un experimento.....	25
Figura 32: captura de la matriz de botones con los 7 primeros elementos configurados.....	26
Figura 33: captura de los botones inferiores de navegación	26
Figura 34: captura de la pantalla de configuración de experimento	27
Figura 35: captura de la pantalla de configuración de serie	28
Figura 36: captura de pantalla de los elementos que permiten hacer uso del trial plantilla	28
Figura 37: captura de la pantalla de configuración de trial	29
Figura 38: captura de la pantalla de configuración de agente.....	31
Figura 39: estructura de directorios para almacenar el fichero de configuración de un experimento	32
Figura 40: captura de la pantalla de parada mostrada antes de ejecutar cada serie	33
Figura 41: captura de la pantalla de relax mostrada antes de ejecutar cada trial.....	33
Figura 42: diferencia de períodos temporales de captura de datos obtenidos desde cada fuente	34
Figura 43: captura de la pantalla que permite seleccionar el trial a reproducir.....	37
Figura 44: protocolo de comunicación entre la plataforma y la librería que gestiona la captura EEG	39
Figura 45: formato del fichero donde se guarda la captura de la evolución del estado del entorno.....	40
Figura 46: posible visualización estructurada del fichero de captura del estado del entorno desde una aplicación externa.....	40
Figura 47: estructura de directorios para almacenar los ficheros de captura y el fichero de configuración de un experimento	41
Figura 48: sujeto humano ejecutando un experimento de la plataforma mientras su señal cerebral es capturada.....	44

Índice de tablas

Tabla 1: parámetros editables desde la pantalla de configuración de experimento.....	27
Tabla 2: parámetros editables desde la pantalla de configuración de serie.....	28
Tabla 3: parámetros editables desde la pantalla de configuración de trial.....	30
Tabla 4: parámetros editables desde la pantalla de configuración de agente	31
Tabla 5: objetivos cumplidos en el desarrollo de la plataforma	45

1 Introducción

1.1 Motivación

Este proyecto da continuidad a varios estudios cognitivos de carácter teórico realizados durante los últimos años en el contexto del equipo ISAAC (Interdisciplinary Studies in Adaptivity, Autonomy and Cognition) del Departamento de Ingeniería e Informática de Sistemas. En este estudio, el objeto es el análisis de los bucles sensomotores de interacción continua que los sujetos establecen en procesos de exploración del espacio (según la teoría sensomotora desarrollada por [O'Regan, 2001]). En este trabajo nos centraremos en el desarrollo de una plataforma experimental que permita estudiar aspectos de la percepción espacial en mundos diseñados para que sean mínimos (en complejidad y en dimensionalidad) y que habilite la lectura de señal EEG del sujeto participante.

1.2 Objetivos y Alcance

El objetivo principal de este proyecto es el desarrollo de una plataforma software que permita configurar y ejecutar experimentos de interacción entre un sujeto humano y un agente virtual dentro de un entorno unidimensional, obteniendo durante la ejecución información de la evolución del estado del entorno y de la evolución de la señal cerebral del sujeto humano.

1.3 Contexto en el que se realiza el proyecto

El uso de plataformas mínimas de experimentación en Ciencia Cognitiva es una línea de trabajo que busca definir entornos elementales donde se insertan agentes sintéticos físicamente simulados y en los que se hacen evolucionar los parámetros de dichos modelos utilizando como función de optimización la satisfacción de alguna tarea cognitiva mínima. El acceso a la evolución (digitalmente) almacenada y la posibilidad de realizar experimentación exhaustiva y análisis dinámico del conjunto de variables y parámetros que constituyen los modelos simulados y su interacción con ellos, hace de estas herramientas entidades notables de análisis.

Con el diseño de esta plataforma buscamos poder explorar la interacción entre acoplamientos sensomotores y su naturaleza no-lineal que producen los fenómenos cognitivos [Harvey et al, 2005].

Con esta metodología experimental se pretende establecer una nueva alianza entre neurociencia y (tecno) -ciencia a través de simulaciones computacionales que constituyen espacios de fusión conceptual entre hipótesis sobre cognición teórica y conducta cognitiva. Una fusión que permita a su vez explorar la interacción entre evolución, acoplamiento sensoriomotor, mecanismos neuronales y autorregulación en sistemas cognitivos mínimos; replanteando así aspectos fundamentales en hipótesis sobre cognición teórica.

1.4 Herramientas utilizadas

	<p>Doxygen es un parser de texto comentado en ficheros de código fuente que, a partir de un formato establecido que debe reconocer, genera automáticamente el contenido HTML necesario para poder subir a la web la documentación del código parseado con un acabado profesional y un estilo similar a la documentación online de librerías de uso abierto.</p>
	<p>Odoo es un sistema de ERP (<i>enterprise resource planning</i>) integrado de código abierto. En este proyecto se ha utilizado durante un tiempo el módulo de gestión de tareas y proyectos.</p>
	<p>Redmine es una herramienta para la gestión de proyectos que incluye un sistema de seguimiento de incidentes con seguimiento de errores. Se ha utilizado en las fases finales del proyecto.</p>
	<p>Qt Creator es un IDE que permite el desarrollo de aplicaciones con el framework Qt, que a su vez es un conjunto de librerías multiplataforma bajo el paradigma orientado a objetos, especialmente pensado para aplicaciones con interfaz gráfica</p>
	<p>ArgoUML es una aplicación que permite dibujar diagramas UML y que al estar desarrollada en Java está disponible para cualquier sistema operativo que soporte este lenguaje, haciéndola prácticamente de uso universal.</p>

1.5 Organización de la memoria

- Sección 2** Recopilación y exposición de varios trabajos relacionados con la línea de investigación en la que el proyecto está enmarcada.
- Sección 3** Descripción de las funcionalidades de la plataforma siguiendo una estructura bottom-up, comenzando con los parámetros que configuran los experimentos sensomotores, continuando con la estructura temporal de los experimentos, y acabando con un listado de las funcionalidades de alto nivel desarrolladas.
- Sección 4** Documentación del diseño y la implementación de las funcionalidades de alto nivel a través de la interfaz gráfica. Las funcionalidades se separan en Menú principal, configuración, ejecución y reproducción.
- Sección 5** Explicación del formato de los ficheros generados tras la ejecución de los experimentos: captura de la señal cerebral y captura de la evolución del estado del entorno.
- Sección 6** Diseño de experimentos sensomotores configurados especialmente para cada agente virtual desarrollado en la plataforma.
- Sección 7** Tabla con los objetivos planteados antes de comenzar el desarrollo de la plataforma, y su estado de completitud. En este caso, se han completado todos satisfactoriamente.

2 Estado del arte

2.1 Estudio de la percepción: enfoque constructivo del espacio circundante

Uno de los objetos de mayor interés para la Ciencia Cognitiva es el hecho de comprender cómo percibimos el espacio en el que estamos envueltos. El enfoque constructivo del espacio [Piaget, 2012], es decir, la asunción de que la percepción espacial se construye en los primeros meses de vida, es una de las aportaciones más relevantes de los estudios sobre desarrollo cognitivo. Este enfoque, conocido como sensomotor, es contrario a la percepción realista del mundo (que adoptan las ciencias naturales como la física) que entiende que la relación entre el sujeto y su entorno, vía la percepción, se produce en términos de mecanismos de representación.

Esto nos lleva a diferenciar conceptualmente dos formas de entender el espacio en el que estamos inmersos, en función del enfoque que adoptemos: (i) en el enfoque realista, en el que el mundo en que habita un agente es el conjunto de fenómenos físicos que vienen caracterizados por leyes naturales. A este mundo lo denominaremos "entorno" del agente; (ii) en el enfoque constructivista, en el que el mundo del agente estaría construido por él y a través de las interacciones sensomotoras que establece con el medio. A este, lo denominaremos el "espacio sensomotor" que percibe el agente.

El concepto de "espacio sensomotor" fue planteado por primera vez por el biólogo Jakob von Uexküll [Uexküll, 2010], introduciendo la noción de Umwelt (en alemán, "medio ambiente"). El Umwelt es el conjunto de todas las relaciones significativas del ser vivo con su entorno. Según von Uexküll, cada ser vivo tendría un Umwelt, un mundo propio diferente que crearía e incluso modificaría según interaccione con su entorno.

El concepto de Umwelt escapa de la idea de un mundo común para todos los seres vivos regulado por leyes físicas para llegar a la noción de un mundo propio y único que crea cada ser vivo según interacciona con su entorno. Es interesante, por tanto, destacar que la noción de Umwelt puede relacionarse con el enfoque sensomotor, pues es mediante la acción y la percepción como se crea y se modifica el Umwelt propio de cada ser vivo.

Un enfoque formal, sistemático y computacional del enfoque sensomotor sobre la percepción se ha desarrollado en los últimos años [O'Regan, 2001], inspirado en las teorías de Piaget, que considera que, en realidad, "percibimos" mediante leyes sensomotoras. La teoría sensomotora propone que el hecho de "percibir el espacio" se basa en el modo de interacción corporal con nuestro entorno, extrayendo una serie de regularidades, una estructura relacional, de esas interacciones, que percibimos como un espacio que tiene dimensiones y sobre el que ocurren procesos.

2.2 Leyes sensoriales como medio para percibir

En las ciencias tradicionales, el fin de la investigación es la obtención de leyes naturales dirigidas a dar explicaciones causales: estas leyes se conocen como leyes físicas, tienen carácter objetivo y se considera que capturan asépticamente la estructura real de los fenómenos en la naturaleza. Se diferencian de las conocidas como psicofísicas, que asumen un sujeto en interacción con un mundo que supone fuente de estímulos sensoriales, y que se encargan de representar cómo en el sujeto impactan los fenómenos físicos (a qué espectro luminoso son sensibles nuestros órganos de la visión, qué cualidad tiene la sensibilidad acústica de nuestros dispositivos auditivos, etc.). Pretenden representar los patrones que siguen nuestras percepciones del mundo físico.

En esta sección nos interesa introducir un tipo de ley diferente a las anteriores y que denominaremos “ley sensorial”. Estar gobernado por una ley sensorial supone asumir que el sistema no es un ente que recibe información de manera pasiva. Al contrario, implica una relación entre la acción del sistema y la percepción que recibe. A diferencia de los otros tipos de leyes (que suponen un observador pasivo) funciona de manera activa, es decir, el sistema extrae una serie de regularidades entre los movimientos que despliega y los estímulos que recibe.

Una ley física (o psicofísica) es una ley fija, proporcionada desde fuera, sin posibilidad de intervención en la configuración de la misma por parte de un agente. Una ley sensorial que regule el comportamiento de un agente, por contra, es una ley que construye el agente en la propia interacción que establece con su entorno, de manera que la acción es necesaria para la percepción y, a la vez, es la percepción la que condiciona la acción del sujeto, estableciendo bucles de causalidad circular.

Para poder diferenciar las nociones introducidas, consideremos un simple dispositivo foto-táctico, por ejemplo, uno de los vehículos de Braitenberg [Braitenberg, 1986]. El vehículo está equipado con dos sensores situados uno al lado del otro enfrente del vehículo y con dos ruedas situadas una al lado de la otra en la parte trasera. El sensor izquierdo está conectado al mecanismo que mueve la rueda derecha y el sensor derecho está conectado al mecanismo que mueve la rueda izquierda. Como resultado de estas conexiones, el vehículo se orienta por hacia las fuentes lumínicas y se mueve hacia ellas (ver Figura). Un mecanismo tan simple puede detectar y perseguir fuentes luminosas.

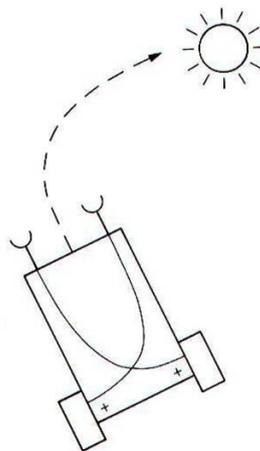


Figura 1: vehículo de Braitenberg

Este sencillo ejemplo, nos permite diferenciar los tres tipos de leyes señalados:

- Una ley física relacionada con el carácter del fenómeno luminoso es la ley de propagación de la luz: la intensidad disminuye en una relación inversamente proporcional al cuadrado de la distancia al centro donde se origina. Respecto al carácter motor del problema, existen leyes mecánicas que explican el desplazamiento de las ruedas por el suelo a partir de la fuerza ejercida por el motor sobre la rueda y la resistencia de rozamiento con el suelo.
- Podemos distinguirlas de leyes psicofísicas sensoras y motoras asociadas. Por ejemplo, la relación entre la intensidad luminosa que le llega al vehículo y la intensidad con la que ésta es percibida por el sensor del mismo (por ejemplo, si la sensibilidad luminosa sigue una función logarítmica con unos umbrales superiores e inferiores a partir de los cuales el sensor deja de detectar la luz). Para el caso de una ley psicofísica motora, podemos seleccionar la relación entre la señal que le llega al motor procedente de los sensores y la acción que éste ejerce sobre la rueda (por ejemplo, la relación que transforma la señal recibida en potencia de giro).
- Por último, en nuestro ejemplo, una ley sensomotora es aquella que relaciona la acción del vehículo con los estímulos que recibe. Este tipo de relación se ve fácilmente en la dinámica de los vehículos de Braitenberg, donde una fuente luminosa a la izquierda del vehículo, hace que el vehículo gire en la dirección de la fuente luminosa. Una vez que el vehículo ha girado, la fuente luminosa ya no estaría a su izquierda, sino a su derecha. Esto causaría que el vehículo girase hacia la derecha hasta que, de nuevo, la luz vuelva a quedar a la izquierda. Esta serie de relaciones estímulo-acción irían moviendo al vehículo hacia la fuente luminosa. Podemos decir que un vehículo así construido se despliega una dinámica gobernada por una ley sensomotora, es decir, un conjunto de reglas de interdependencia entre estímulos y movimiento que le permiten, en este caso tener un comportamiento: perseguir fuentes luminosas.

Para el modo tradicional dominante, el punto de partida para comprender la percepción es abstracto: existen mecanismos de procesamiento de la información capaces de recuperar propiedades del mundo predeterminadas. En contraste, el punto de partida para el enfoque sensomotor es el estudio de cómo en función de los movimientos y acciones se genera la experiencia perceptiva. De este modo, la preocupación de este trabajo en la percepción trata de determinar cómo se producen las vinculaciones entre el sistema sensorial y el motor que explican cómo se nos hace comprensible un espacio mínimo creado en una plataforma experimental.

2.3 Cómo estudiar la generación de leyes sensoriomotoras en entornos experimentales: metodología mínima

La línea de investigación conocida como “cognición mínima” [Beer, 2003; Van Duijn, 2006] busca diseñar experimentos computacionales donde la dimensionalidad sea la mínima compatible con el proceso de estudio, donde el comportamiento de los agentes provenga del control de mínimos recursos o capacidades, y donde se analicen capacidades cognitivas primitivas o los orígenes de capacidades cognitivas más elaboradas. En la última década, la relevancia en la comprensión de fenómenos cognitivos a partir de modelos mínimos ha ido en aumento.

Concentrarse en una versión simplificada, mínima, de un fenómeno de estudio, sirve al investigador como herramienta para pensar sobre las raíces del mismo. Por ejemplo, algunos modelos deliberadamente simples pueden confirmar que existen alternativas explicativas a fenómenos que se suponía que funcionaban de una determinada manera. En palabras de algunos autores [Rohde, 2010], los modelos sirven “como gimnasia mental”, es decir, son entidades equivalentes a los experimentos mentales clásicos, artefactos que nos ayudan a pensar. Estos modelos (computacionales, de simulación, plataformas de experimentación, etc.) son parte de lo que se conoce como “metodología sintética” y que defiende que el estudio ingenieril sobre la mente puede funcionar como un campo científico que permita examinar hipótesis teóricas.

Siguiendo esta filosofía, este proyecto se ha enfocado al diseño de una plataforma experimental con un mundo unidimensional formado por diversos agentes sintéticos con los que el usuario entra en interacción.

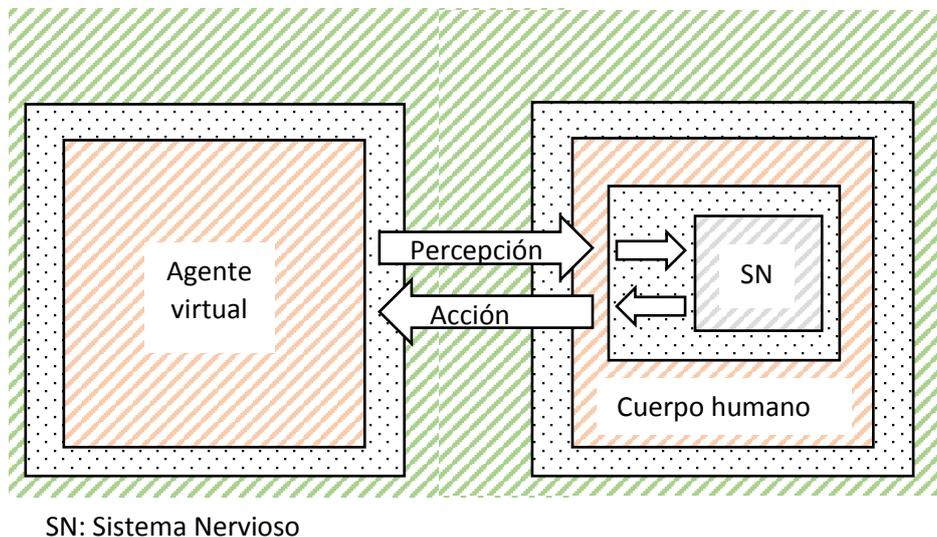


Figura 2: interacción sensoriomotora de un ser humano con un agente virtual

3 Diseño de la plataforma: SensoMotor Lab

3.1 Esquema general

En la interacción humano – agente planteada, el agente no tiene problema para ‘percibir el entorno’, ya que tanto él como el entorno son virtuales, y la percepción del primero se basa en consultar las variables que definen el estado del segundo. Lo mismo ocurre con la acción, siendo ésta para el agente una modificación de las variables que definen su estado dentro del entorno.

Sin embargo, el sujeto humano no tiene acceso directo a las variables internas del ordenador, por lo que hay que diseñar una interacción mediante periféricos de entrada y salida que le permita percibir el estado del entorno y ser capaz de actuar sobre él.

Para poder obtener la captura de la señal cerebral del sujeto durante la ejecución de los experimentos sensomotores, se emplea la tecnología EEG (electroencefalografía), por ser la menos invasiva al medir la señal en la periferia del cerebro, desde el cuero cabelludo.

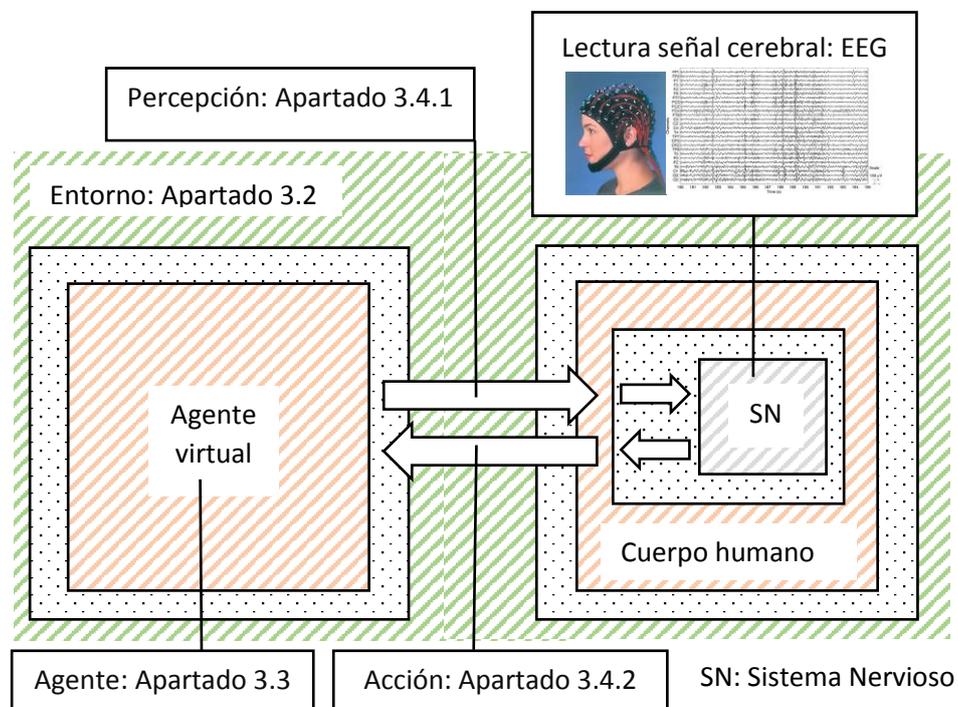


Figura 3: interacción sensomotora de un ser humano con un agente virtual dentro de la plataforma

Las secciones que siguen describen las características de los entornos desarrollados, los distintos comportamientos otorgados al agente virtual con el que va a interactuar el sujeto humano durante los experimentos, y las posibles interacciones implementadas para el sujeto, tanto a nivel de percepción como a nivel de acción.

3.2 Entorno

Para reducir el número de parámetros, se ha considerado un entorno unidimensional. En su interior sólo hay ubicados 2 elementos: el sujeto humano y el agente virtual. Ambos tienen capacidad de desplazarse por la dimensión del entorno obedeciendo a las restricciones dadas por éste.



Figura 4: el entorno contiene al sujeto (bola anaranjada) y al agente virtual (bola verde)

Según el tipo de agente y el comportamiento que se desee estudiar en el sujeto que interactúa con él se plantean 3 tipos de entorno con distinto grado de restricción al movimiento:

Entorno lineal acotado: es el más restrictivo. Acota el desplazamiento a un rango $[0.0, 1.0]$ y los límites del rango son infranqueables, como si hubiera una pared. Está pensado para experimentos donde el sujeto debe reconocer el patrón de movimiento libre de un agente que siempre está dentro de los límites.

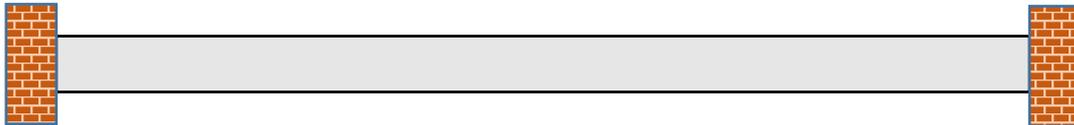


Figura 5: entorno lineal acotado

Entorno toroidal: Mantiene el rango $[0.0, 1.0]$ de desplazamiento, pero los límites están unidos entre sí, de modo que franquear el 1.0 implica aparecer en el 0.0 y viceversa, como si el entorno fuera el interior un toroide. Está pensado para experimentos con agentes cuyo movimiento está influido por los movimientos del sujeto, en los que la tarea deja de ser factible si el sujeto choca con paredes, pero donde no interesa el recorrido del sujeto por el entorno, sino la distancia a la que se encuentre del agente en cada momento.

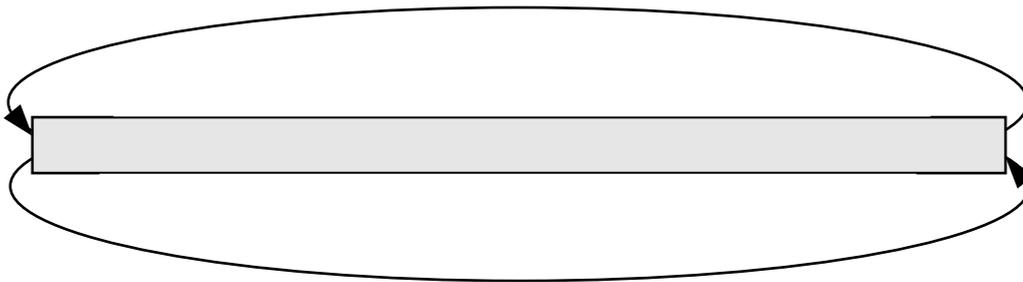


Figura 6: entorno toroidal

Entorno continuo: es el menos restrictivo. Permite desplazamiento por toda la dimensión, sin límites. Está pensado para experimentos en los que sí interese obtener una traza continua de las coordenadas por las que se va moviendo el sujeto durante la ejecución.



Figura 7: entorno continuo

3.3 Agente virtual: generación de comportamiento

La existencia del agente virtual se reduce a su posición dentro del entorno, y a una función de comportamiento que modifica esa posición a partir del estado actual del entorno, definido a partir de la posición actual del agente y la posición actual del sujeto humano.

Para la plataforma se han desarrollado 4 agentes con distinto comportamiento:

- **Agente estático:** el comportamiento más simple de todos. Permanece siempre en la misma posición. La ecuación empleada es la siguiente:

$$\text{nextPos}(\text{ageX}) = \text{ageX}$$

Figura 8: ecuación de comportamiento del agente estático

- **Agente sinusoidal:** este agente se limita a desplazarse de un lado al otro del entorno en el rango [0,0, 1,0] de forma continua, siguiendo una función sinusoidal con una frecuencia configurable, partiendo de una posición y sentido de movimiento también configurables. Las ecuaciones que permiten este comportamiento son las siguientes:

$$\text{nextPos}(\text{ageX}) = (\cos(\omega t + \varphi_0) + 1)/2$$

$$\omega = 2\pi f$$

$$t = t_{\text{actual}} - t_{\text{inicial}}$$

$$\varphi_0 = \begin{cases} (1 - \text{ageX}_{\text{init}}) * \pi & \text{si } \text{dirInit} = \text{izquierda} \\ 2\pi - (1 - \text{ageX}_{\text{init}}) * \pi & \text{si } \text{dirInit} = \text{derecha} \end{cases}$$

Figura 9: ecuaciones que determinan el comportamiento del agente sinusoidal. En la primera, la adición de uno al coseno y la división entre 2 posterior se emplean para pasar del rango [-1,1] propio del coseno al rango [0,1] propio del entorno.

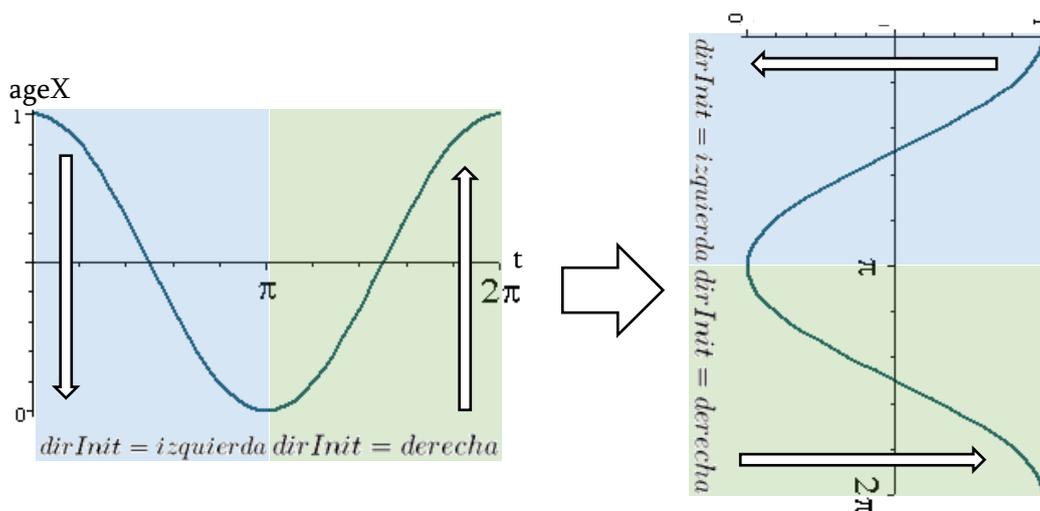


Figura 10: relación entre la fase inicial y la dirección inicial en el agente sinusoidal: entre 0 y π , el coseno en rango [0,1] se acerca a 0, hacia la izquierda. Entre π y 2π , el coseno se acerca a 1, hacia la derecha.

- **Agente péndulo invertido:** simula el desplazamiento del extremo superior de un péndulo invertido. El sujeto se encuentra en el otro extremo del péndulo, y al desplazarse modifica el ángulo de inclinación del péndulo con respecto a la vertical. Si no se mueve, el péndulo también modifica su grado de inclinación por efecto de un tipo especial de gravedad. Es especial porque, al contrario que la gravedad real, que otorga al péndulo mayor velocidad de caída cuanto más se aleja de la vertical, esta gravedad otorga al péndulo la mayor velocidad en la vertical, pero va ralentizando la caída del péndulo conforme éste se va alejando de la vertical. Este cambio en la física se ha visto necesario después de comprobar la gran dificultad de tratar de mantener el péndulo invertido sin caerse cuando sólo se percibe el estado del entorno mediante el canal auditivo. Las ecuaciones que permiten simular este comportamiento son las siguientes:

$$nextPos(ageX, sujX) = sujX - \sin(next\varphi) * poleLength$$

$$next\varphi = \varphi + \frac{d\varphi}{dt} * t_{ciclo}$$

$$\varphi = \arcsin \frac{sujX - ageX}{poleLength}$$

$$\frac{d\varphi}{dt} = \begin{cases} -g - \frac{g\varphi}{\pi/2} & \text{si } 1^{\circ} \text{ cuadrante} \\ g - \frac{g\varphi}{\pi/2} & \text{si } 2^{\circ} \text{ cuadrante} \end{cases}$$

Figura 11: ecuaciones que determinan el comportamiento del agente péndulo invertido

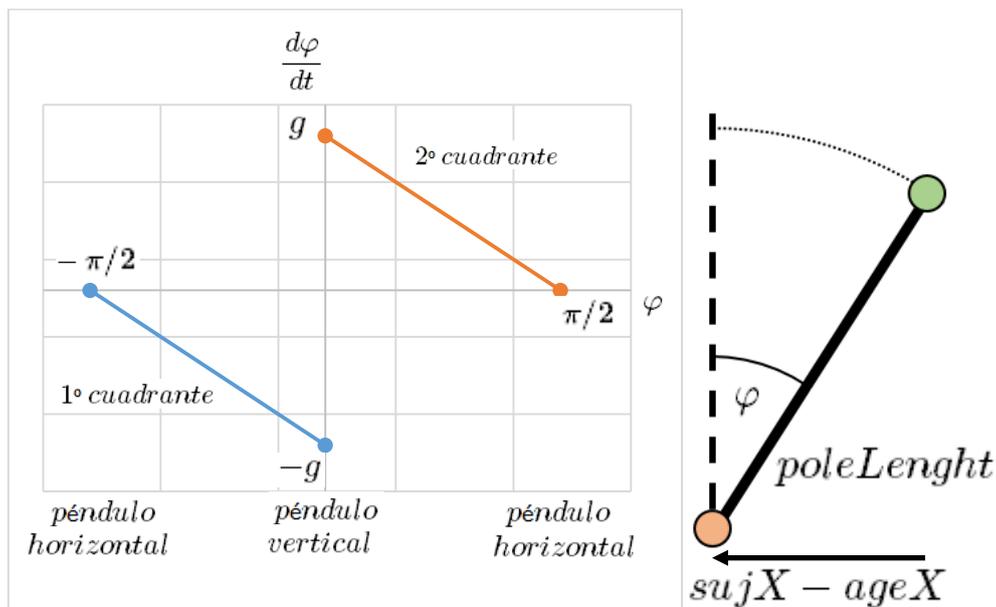


Figura 12: función [distancia angular]-[velocidad angular] y esquema de la trigonometría inherente al agente péndulo invertido

- **Agente mano loca:** simula la dinámica del juguete infantil conocido en España como mano loca, y en EEUU como sticky hand, muy popular en los 80 y los 90. Consiste en un material pegajoso unido con un nexo elástico a la mano del usuario. Si el material se adhiere a una superficie y el usuario comienza a estirar de él, alejando del trozo de material pegado el extremo que él sujeta en dirección paralela a la superficie, podrá alejarse una cierta distancia sin que el material se desprenda, pero a una distancia determinada, la tensión aplicada desde el nexo elástico vencerá a la fuerza de rozamiento que impide a la superficie pegada moverse, y ésta se moverá hacia el usuario siguiendo un movimiento rectilíneo uniformemente acelerado, con una velocidad inicial alta y una aceleración negativa, hasta que el material se pegue en otra superficie o la tensión del nexo elástico lo haga volver nuevamente hacia el usuario.



Las ecuaciones que permiten simular este comportamiento son las siguientes:

$$\begin{aligned}
 & \text{nextPos}(sujX, ageX) = \\
 & \begin{cases} ageX & \text{si } (|sujX - ageX| < shootDistance) \text{ AND } (v_{actual} = 0) \\ \dots & \text{si } (|sujX - ageX| \geq shootDistance) \text{ OR } (v_{actual} > 0) \end{cases} \\
 & \dots = \begin{cases} ageX + v_{actual} * t_{ciclo} + \frac{a * t_{ciclo}^2}{2} & \text{si } (sujX - ageX) \geq 0 \\ ageX - v_{actual} * t_{ciclo} + \frac{a * t_{ciclo}^2}{2} & \text{si } (sujX - ageX) < 0 \end{cases} \\
 & v_{actual} = v_{max} + a * (t_{actual} - t_{shoot})
 \end{aligned}$$

Figura 13: ecuaciones que definen el comportamiento del agente mano loca

3.4 Interacción del sujeto humano con el agente

Como se ha comentado, el sujeto humano no tiene acceso directo a la memoria de la aplicación, por lo que para hacer posible que perciba el estado del entorno es necesario generar con el ordenador una representación del mismo a través de alguno de sus periféricos de salida, de manera que esa representación sea percibida a través del correspondiente canal sensorial.

La representación del estado del entorno mencionada se debe entender como una traducción desde estado lógico (1s y 0s) guardado dentro del ordenador hasta un formato perceptible por el sujeto, y NO con la idea de representación de la información percibida dentro del cerebro del sujeto, idea que la teoría sensomotora invalida.

3.4.1 Percepción

La filosofía de los experimentos realizados con esta plataforma busca que el sujeto no sea capaz de acoplarse con el entorno de forma rápida y sencilla, como lo haría mediante el canal visual, sino que deba explorar el entorno mediante el movimiento para poder entender su funcionamiento. Para conseguir esto, se ha desarrollado la capacidad de representar el estado del entorno a través del canal auditivo. Igualmente, se puede representar por el canal visual para entrenamientos, comprobaciones o reproducciones posteriores.

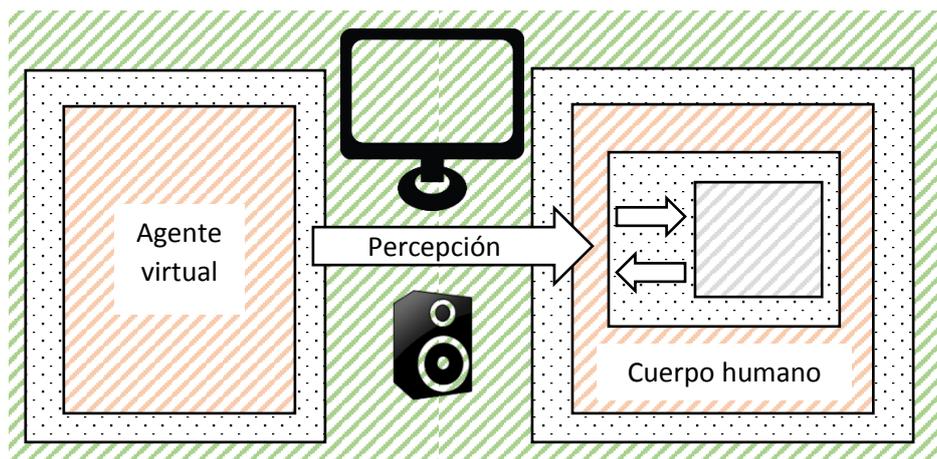


Figura 14: percepción del agente por parte del sujeto humano

3.4.1.1 Percepción visual

Para la interacción mediante el canal visual se emplea como periférico la pantalla del ordenador. En ella se representa el entorno a pantalla completa:

- Una línea gruesa, horizontal, de color blanco, que se extiende de un lado a otro de la pantalla, indica visualmente la dirección de desplazamiento por el entorno.
- A lo largo de esa línea blanca se representa el sujeto mediante un círculo de color azul cian y el agente mediante un círculo de color gris claro.



Figura 15: representación visual del entorno

Se han desarrollado dos formas de representar el estado del entorno por el canal visual: aquella que genera una representación a la que estamos habituados en el mundo real, y una diferente, que invierte la representación habitual respecto al eje vertical, como si se estuviera observando el estado del entorno reflejado en un espejo.

Representación visual no invertida: el mapeo de las coordenadas del entorno a los pixels de pantalla se aplica directamente.

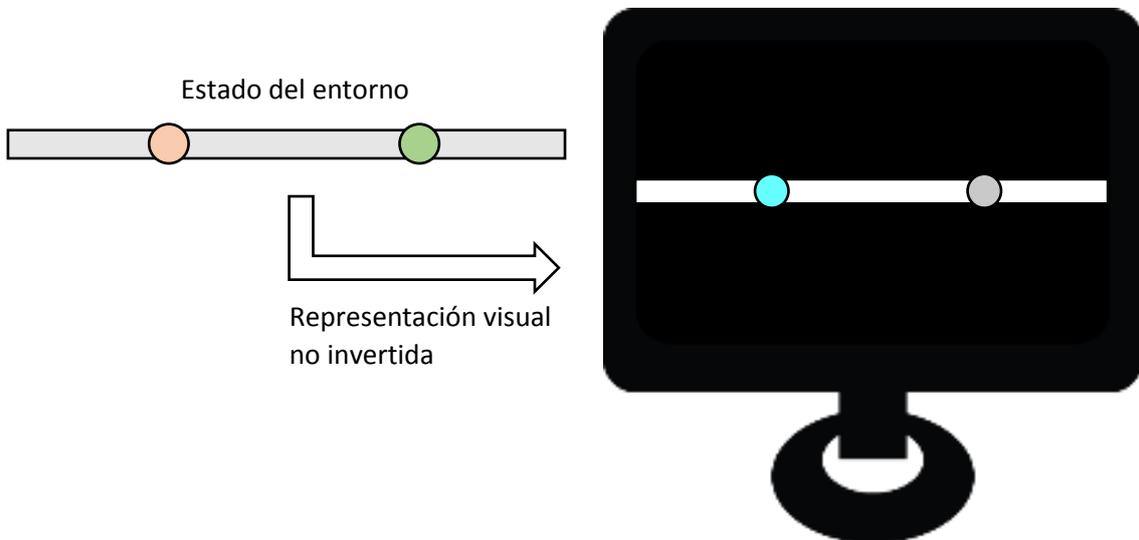


Figura 16: representación visual normal del estado del entorno

Representación visual invertida: Antes de aplicar el mapeo directo de coordenadas, estas se transforman en su equivalente especular, considerando el eje vertical del centro de la pantalla como un eje de giro de 180°.

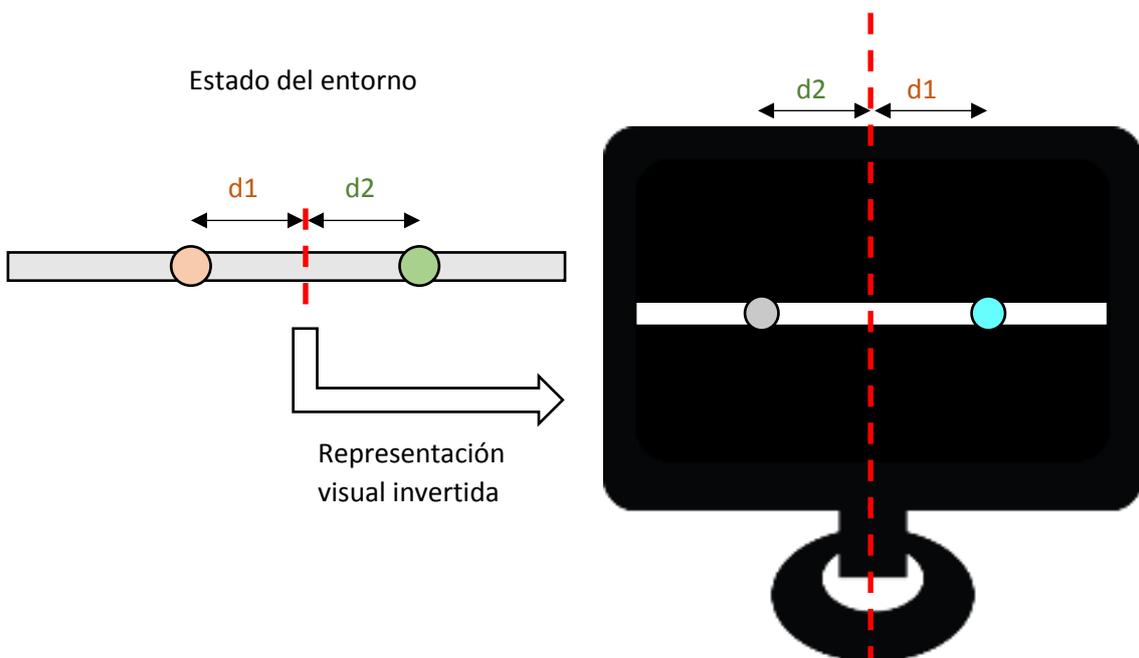


Figura 17: representación visual invertida del estado del entorno

3.4.1.2 Percepción auditiva

Para la interacción mediante el canal auditivo, el periférico elegido es el altavoz del sistema. En ella se representa el estado del entorno a través de una onda sonora de frecuencia, timbre y duración determinados, reproducida en bucle durante toda la ejecución. A partir de la distancia entre el sujeto y el agente dentro del entorno, se calcula un valor de volumen y se asigna este volumen a la reproducción de la onda sonora.

Al igual que en el caso de la percepción visual, se ha implementado percepción auditiva del modo en que el sujeto está habituado a percibirla en el mundo real, donde cuanto más cerca se encuentra de un foco sonoro con más volumen lo percibe, y su inversa. Además se han planteado dos aproximaciones a la variación del volumen con la distancia: por un lado variación lineal, y por otro variación no lineal, donde a poca distancia el cambio de volumen es grande pero conforme la distancia aumenta el sonido varía más lentamente.

Percepción auditiva lineal: se define el volumen a distancias mínima y máxima, y la transición intermedia se interpola linealmente. Se considera distancia máxima 1.0 por ser lo más lejos que se pueden encontrar sujeto y agente en los entornos acotado y toroidal. En el entorno continuo, si la distancia es mayor que 1.0, se mantiene el mismo volumen que para la distancia 1.0.

- Percepción auditiva lineal no invertida: a distancia 0.0 el volumen es máximo. A distancia 1.0 o superior, el volumen es nulo.

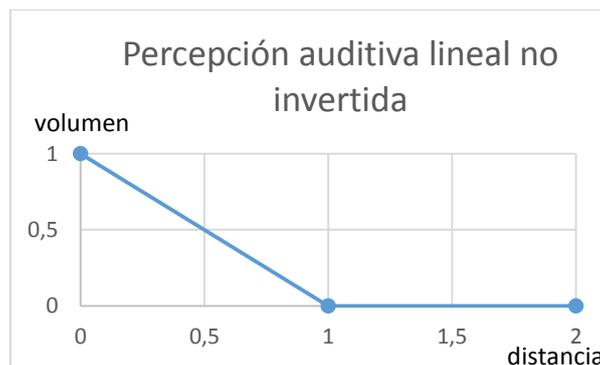


Figura 18: función distancia-volumen en percepción sonora normal

- Percepción auditiva lineal invertida: a distancia 0.0 el volumen es nulo. A distancia 1.0 o superior, el volumen es máximo.

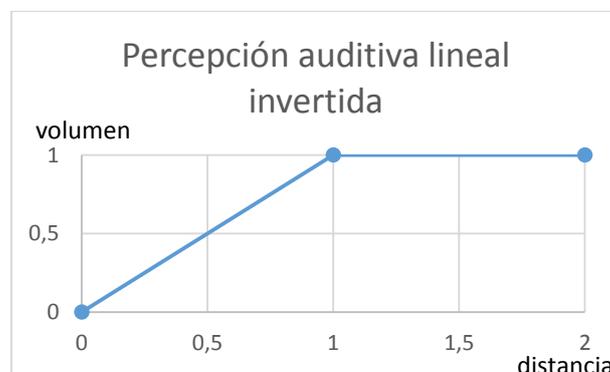


Figura 19: función distancia-volumen en percepción sonora invertida

Percepción auditiva gaussiana: se define el volumen a distancia mínima, y ese volumen corresponde al único extremo local de la función de densidad de probabilidad de una distribución normal, escalada para que su amplitud sea 1.0 en el eje Y.

- Gaussiana no invertida: a distancia 0.0 el volumen es máximo. A poca distancia (aprox. 0.2 en adelante), el volumen es prácticamente nulo. Se aplica la siguiente ecuación, con $\mu = 0.0$ y $\sigma = 0.05$:

$$\left\langle \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \right\rangle * \sigma\sqrt{2\pi}$$

Función de densidad de probabilidad original

Normalización entre 0.0 y 1.0 de la función de densidad

Figura 20: ecuación para la percepción sonora gaussiana normal

- Gaussiana invertida: a distancia 0.0 el volumen es nulo. A distancia mayor, se aplica la siguiente ecuación, con $\mu = 0.0$ y $\sigma = 0.05$:

$$1 - \left\langle \left\langle \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \right\rangle * \sigma\sqrt{2\pi} \right\rangle$$

Normalización entre 0.0 y 1.0 de la función de densidad

Inversión de la función de densidad

Figura 21: ecuación para la percepción sonora gaussiana invertida

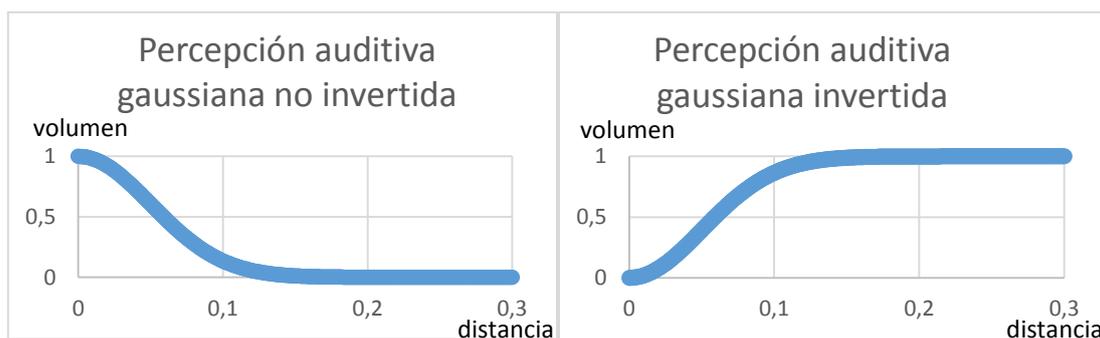


Figura 22: función distancia-volumen en percepción sonora gaussiana normal e invertida

3.4.2 Acción

La otra parte de la interacción consiste en definir un periférico de entrada del ordenador que transforme movimientos realizados por el sujeto desde el mundo real en cambios de su posición dentro del entorno. El periférico elegido es, por facilidad de uso, el ratón. El sujeto desplaza el ratón horizontalmente para modificar su posición dentro del entorno. En cualquier caso, la misma funcionalidad es otorgada por el touchpad, por lo que cualquiera de los dos periféricos de entrada se puede emplear durante los experimentos para desplazarse por el entorno.

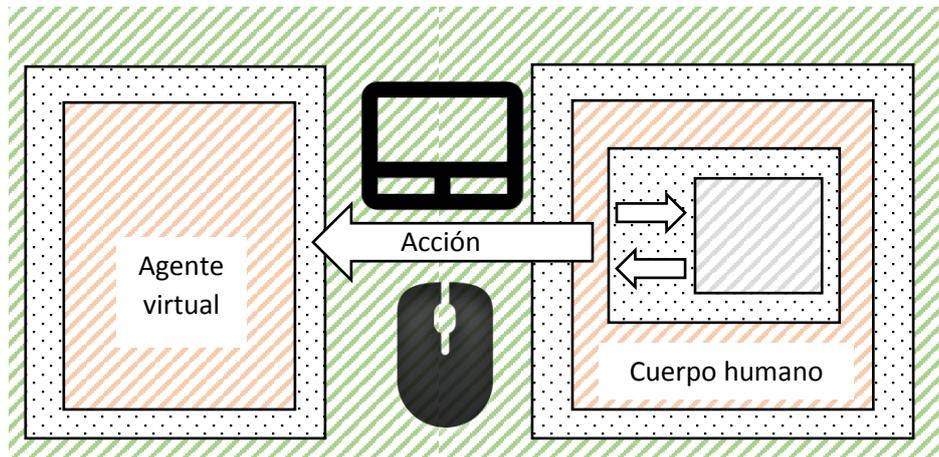
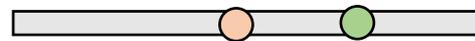


Figura 23: acción del sujeto humano sobre el agente

Se han desarrollado dos leyes físicas distintas de desplazamiento por el entorno:

- No invertida

Acción: desplazamiento hacia la derecha



Sujeto se desplaza hacia la derecha

Ley física del entorno: física no invertida

Figura 24: ley física normal del entorno

- Invertida

Acción: desplazamiento hacia la derecha



Sujeto se desplaza hacia la izquierda

Ley física del entorno: física invertida

Figura 25: ley física invertida del entorno

3.5 Definición de experimento: elementos y estructura temporal

El tipo de experimentos habituales realizados a un sujeto al que se le está monitorizando su actividad cerebral consisten en la ejecución de una secuencia de trials individuales, en cada uno de los cuales al sujeto se le solicita que intente resolver una tarea.

Para capturar la señal cerebral correctamente, el sujeto debe permanecer toda la duración de cada trial con la cabeza inmóvil, ya que los movimientos musculares generan intensidad eléctrica en la zona que es capturada por los sensores cercanos. Por ejemplo, el pestañeo del sujeto durante la captura del EEG genera un artefacto que distorsiona la señal capturada.

Para que el sujeto no tenga que aguantar sin pestañear una secuencia de trials completa, se precede cada nuevo trial con un período de relax.

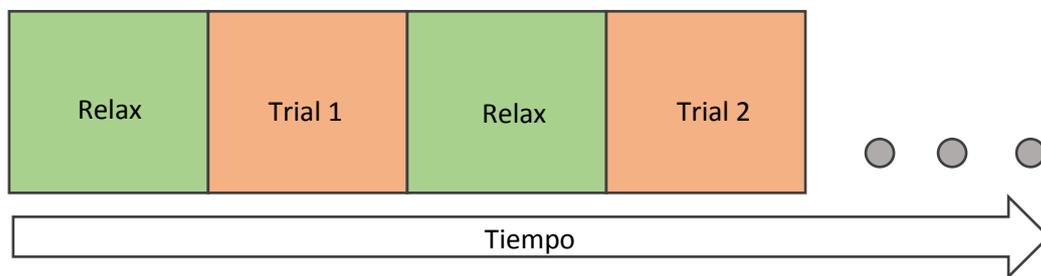


Figura 26: estructura temporal de la ejecución de una secuencia de trials

Dependiendo del número de trials y el tiempo de ejecución de cada uno, la ejecución de un experimento se puede alargar mucho en el tiempo, llevando a la pérdida progresiva de la atención del sujeto hacia la tarea que se le solicita realizar. En estos casos, lo más adecuado es detener el experimento un rato, que el sujeto cambie de contexto y descansa hasta que se encuentre con ánimo de continuar. Para permitir este descanso, la secuencia completa se divide en grupos de un número razonable de trials, con un descanso indefinido antes de comenzar su ejecución. Cada conjunto reducido de trials se denomina serie. Así pues, se define un experimento como una secuencia de series.

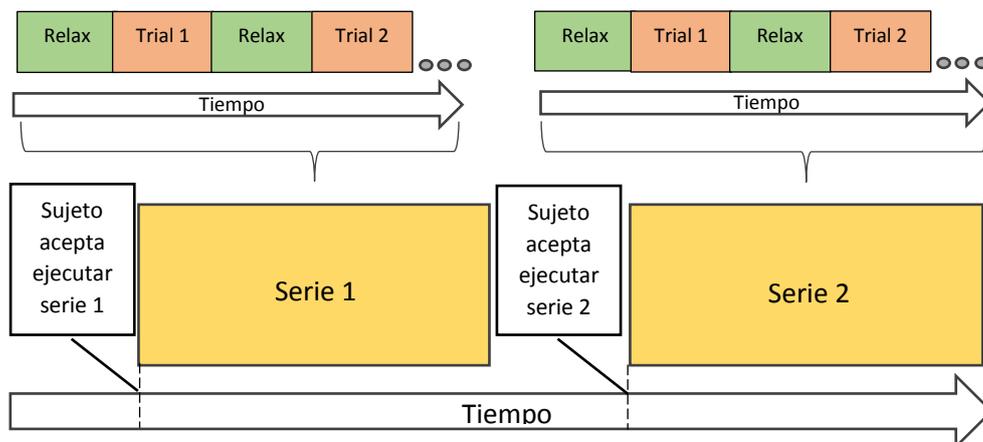


Figura 27: estructura temporal de la ejecución de un experimento con dos series

3.6 Módulos de la plataforma

Como se ha indicado, el objetivo principal de este proyecto es desarrollar una plataforma que permita configurar experimentos sensomotores, después ejecutarlos, y extraer, durante la ejecución de los mismos, información de dos fuentes distintas con la que al final del experimento se generarán distintos ficheros.

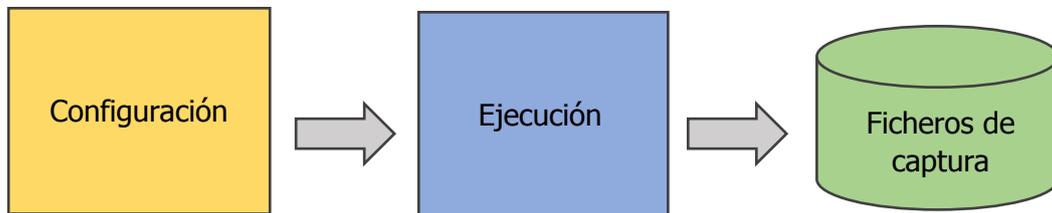


Figura 28: módulos principales de la plataforma

Sin embargo, se plantean otros objetivos adicionales para mejorar notablemente la usabilidad de la plataforma:

- Si sólo se guarda la configuración de un experimento en la memoria interna de la aplicación, ésta se pierde al cerrar el programa, y habrá que volver a configurar el mismo experimento la próxima vez que se abra si se desea ejecutar de nuevo. Para evitar esta situación, se añaden 2 funcionalidades:
 - almacenamiento de la información de configuración de un experimento en un fichero externo a la aplicación.
 - carga de la configuración de un experimento a partir de su fichero externo.
- Para poder consultar los parámetros de un experimento más allá del momento de su configuración, se habilita la visualización de todos sus parámetros.
- Si durante la visualización se desea cambiar algún parámetro, o aprovechar una configuración ya hecha para crear un nuevo experimento muy similar, se habilita la posibilidad de editar la configuración de un experimento durante su visualización.
- Por último, se habilita la posibilidad de reproducir la ejecución de un experimento ya realizado, pudiendo observarse la evolución que ha seguido el estado del entorno en el tiempo.

El anexo I muestra los diagramas de caso de uso para cada funcionalidad desarrollada.

4 GUI

Para el desarrollo de la interfaz gráfica de la plataforma ha sido de mucha utilidad el uso de la librería Qt, que por un lado ya tiene implementados muchos widgets de uso general y facilita la animación de los mismos mediante asignación de funciones a eventos generados en la interacción con ellos, y por el otro permite desarrollar widgets personalizados sin mucho esfuerzo, lo cual ha sido necesaria en este proyecto, materializado en la matriz de botones, para permitir una navegación más eficiente y una realimentación del estado de la configuración más cómoda y completa para el usuario. Qt también permite personalizar el estilo visual de la aplicación a partir de una plantilla que puede estar escrita en un fichero externo y leerse al comienzo de la ejecución, posibilitando cambiar de estilo visual sin tener que compilar de nuevo toda la aplicación.

El anexo II describe la secuencia de código empleada para generar la interfaz gráfica, tanto a nivel de pantallas individuales como de navegación entre ellas.

Seguidamente se describen las funcionalidades de la plataforma a partir de las pantallas que las hacen posibles.

4.1 Menú principal

Es la pantalla principal de la plataforma. En ella hay 5 botones que permiten acceder a todas las funcionalidades que ofrece.



Figura 29: detalle de captura del menú principal

La funcionalidad de guardar un fichero externo de configuración se realiza automáticamente al aceptar la configuración de un experimento, ya sea experimento nuevo o experimento ya existente al que se le han modificado parámetros.

La generación de los ficheros de captura se realiza automáticamente durante la ejecución del experimento correspondiente y al final de la misma los ficheros están disponibles para su uso.

Al pulsar el botón de carga de un fichero de configuración, se abre un explorador de ventanas que permite navegar por el árbol de directorios del sistema para buscar el fichero deseado. Tras seleccionarlo, la plataforma lee su configuración y la guarda, volviendo al menú principal.

4.1.1 Experimento activo

Al iniciar la ejecución de la plataforma no hay ninguna configuración de experimento guardada, y por lo tanto las funcionalidades de ver/editar, ejecutar y reproducir un experimento están deshabilitadas. Para guardar la configuración de un experimento en la plataforma y habilitar esas funcionalidades, se pueden hacer dos cosas: o bien configurar un nuevo experimento, ya que en el momento de aceptar la configuración ésta se guardará tanto en la plataforma como en el fichero externo correspondiente; o bien cargar una configuración ya creada desde su fichero externo correspondiente. El menú principal indica si hay o no un experimento activo en cada momento, mostrando en caso afirmativo el path del fichero externo que guarda su configuración.



Figura 30: detalle de captura del menú principal sin el experimento activo seleccionado

4.2 Configuración

Para configurar un experimento, hay que configurar todas sus series, y para configurar una serie hay que configurar todos sus trials. Pero además, dado que hay 4 tipos de agentes disponibles en la plataforma, que tienen conjuntos de parámetros diferentes, se considera su configuración de forma separada a la del trial. Así, para cada elemento mencionado se habilita una pantalla de configuración diferente:

- Configurar experimento
- Configurar serie
- Configurar trial
- Configurar agente

4.2.1 Navegación entre las pantallas de configuración

Al comenzar la configuración se accede a la pantalla del elemento más alto en la jerarquía (configuración del experimento), que en ese momento no tiene creada ninguna serie. El investigador introduce el nº de series que desea crear, y accede a cada una de ellas individualmente para configurarlas. Al acceder a la configuración de una serie nueva, ésta no tiene creado ningún trial, por lo que el investigador debe introducir el nº de trials que desea crear y acceder a configurarlos individualmente. Por último, desde la pantalla de configuración de un trial se da acceso a la pantalla de configuración de su agente correspondiente.

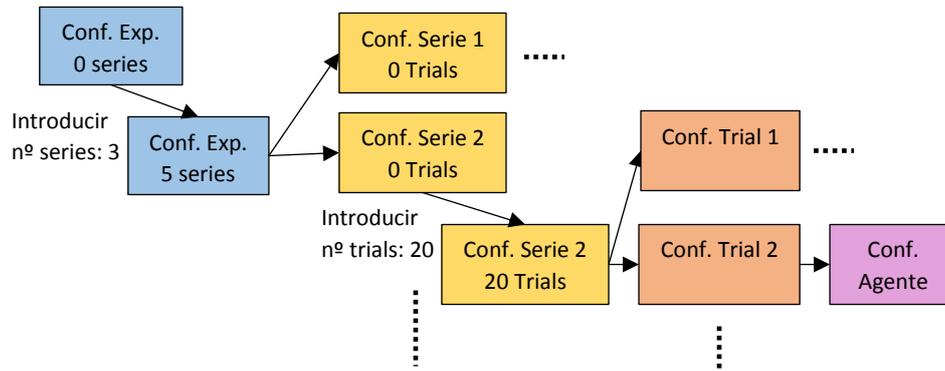


Figura 31: sentido de navegación entre pantallas de configuración de un experimento

El acceso de una pantalla a otra se realiza siempre por medio de botones. Éstos pueden pertenecer a la matriz de botones o al conjunto de botones inferiores de la pantalla:

- Matriz de botones: la pantalla de configuración de experimento muestra una matriz de botones que al iniciar la configuración aparece vacía. En ese estado, se muestra un texto que solicita al investigador que introduzca el nº de series del experimento. Tras introducirlo y pulsar un botón para aceptar, desaparece el texto y en la matriz aparecen tantos botones como nº series introducido, con un texto que asigna un índice distinto a cada botón. Pulsando esos botones se accede a la pantalla de configuración de la serie correspondiente a su índice. En la pantalla de configuración de series aparece otra matriz de botones con el mismo funcionamiento, salvo que los botones de esta matriz dan acceso a los trials de la serie que se esté configurando en ese momento.
 - o Estado 'Elemento configurado': al aceptar la configuración de una serie, se regresa a la pantalla de configuración del experimento. En ella se indica que la serie está configurada mostrando un borde de color verde alrededor del botón correspondiente de la matriz. Se aplica el mismo funcionamiento en la pantalla de configuración de series con respecto a sus trials configurados.



Figura 32: captura de la matriz de botones con los 7 primeros elementos configurados

- Botones inferiores: aceptar, cancelar, volver
 - Aceptar: En la pantalla de configuración del experimento, al pulsar el botón aceptar se comprueba que los parámetros introducidos son correctos y que no queda ningún parámetro por configurar, y si es así se guarda la configuración en la plataforma, se escribe el fichero externo con la configuración del experimento y se muestra el menú principal con el experimento recién creado como experimento activo. Si hay algún error, se mantiene la misma pantalla de configuración de experimento y se indica la fuente del error. Para el resto de pantallas de configuración el funcionamiento es similar: también se comprueba que no haya errores, y si los hay se indican sobre la misma pantalla. La diferencia es que si no hay errores, no se vuelve al menú principal ni se crea ningún fichero externo, sino que se regresa a la pantalla inmediatamente superior de la jerarquía después de haber guardado los parámetros en la estructura interna que usa la plataforma para el experimento activo.
Hay que tener en cuenta para las pantallas que contengan matrices de botones, que hasta que todos los botones estén en estado 'Elemento configurado', se mostrará error al aceptar por haber elementos aún sin configurar.
 - Cancelar: está presente en todas las pantallas de configuración, y permite volver al menú principal en cualquier momento desechando la configuración actual.
 - Volver: sólo está presente en las pantallas de configuración de serie, trial y agente, y permiten regresar a la pantalla de configuración del elemento inmediatamente superior (experimento, serie y trial, respectivamente).



Figura 33: captura de los botones inferiores de navegación

4.2.2 Configurar experimento

Al momento de guardar la configuración del experimento en un fichero externo es necesario saber en qué directorio va a estar localizado y qué nombre va a tener. Ambos son parámetros que debe definir el investigador en el momento de la configuración. Por otro lado, se debe indicar el número de series que se desean configurar para el experimento, y configurar cada una por separado.

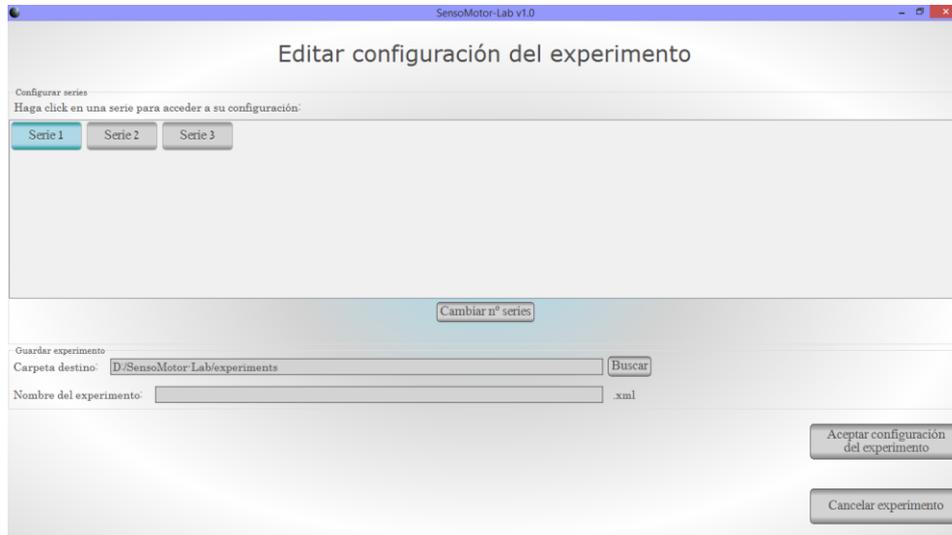


Figura 34: captura de la pantalla de configuración de experimento

Parámetro	Método de configuración	Comentarios
Path experimento	Buscar en sistema de ficheros	
Nombre experimento	Introducir vía texto	Solo letras, números, '-' y '_'
Nº series experimento	Introducir vía texto y aceptar	Máximo nº series: 100 Configuración una a una

Tabla 1: parámetros editables desde la pantalla de configuración de experimento

4.2.3 Configurar serie

Para configurar una serie sólo es necesario indicar cuántos trials se van a configurar en ella y configurar cada trial por separado. En caso de que varios o todos los trials de una serie deban tener una misma configuración, el acceso uno por uno para hacer configuraciones idénticas puede resultar tedioso. Para acelerar el proceso en estos casos, se habilita una funcionalidad que permite acceder a la configuración de un trial especial, denominado trial plantilla, tal que una vez configurado puede ser utilizado para copiar su configuración sobre varios trials a la vez.

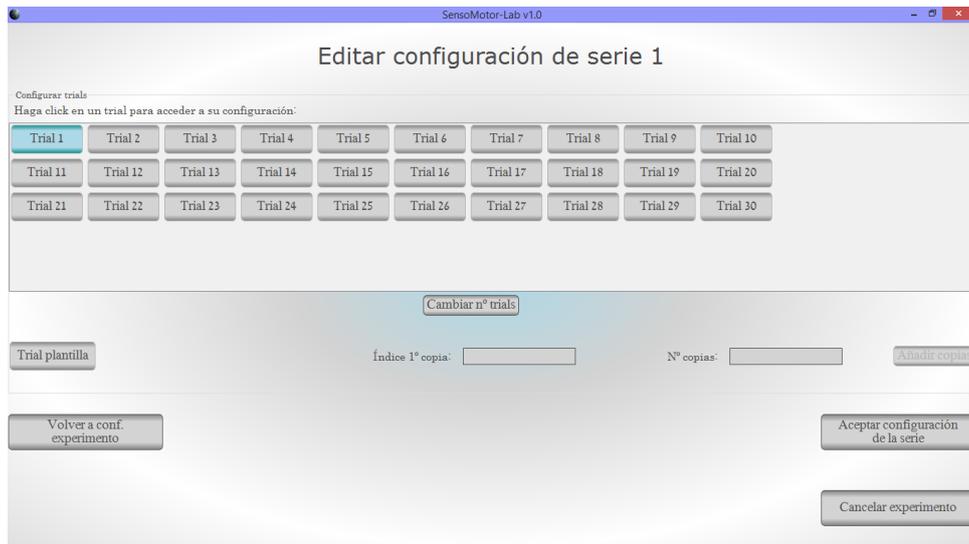


Figura 35: captura de la pantalla de configuración de serie

4.2.3.1 Trial plantilla

Una vez indicado el nº de trials de la serie, cuando se muestran los botones de acceso a su configuración en la matriz, aparece también un botón con el texto 'Trial plantilla' debajo de la matriz. Al pulsar éste se accede a la pantalla de configuración de trial habitual. Una vez realizada y aceptada la configuración de este trial, se regresa a la pantalla de configuración de serie y el botón del trial plantilla aparece con un borde de color verde, indicando que está correctamente configurado. A su derecha se habilitan 2 campos de texto a rellenar. El primero es para indicar el índice en la matriz de botones del 1º trial al que se desea aplicar la copia de la configuración del trial plantilla. El segundo campo es para indicar a cuántos trials, incluyendo al del índice dado por el primer campo y siguiendo en orden creciente de índice, se quiere copiar dicha configuración. Por ejemplo, para una serie con 5 trials (índices 1, 2, 3, 4 y 5, respectivamente), un valor de 2 en el 1º campo y un valor de 3 en el 2º campo copia la configuración del trial plantilla a los trials con índice 2, 3 y 4, es decir, a 3 trials con índices sucesivos, comenzando desde el trial 2. En este caso, los trials 1 y 5 permanecen como estaban.



Figura 36: captura de pantalla de los elementos que permiten hacer uso del trial plantilla

Parámetro	Método de configuración	Comentarios
Nº trials serie	Introducir vía texto y aceptar	Máximo nº trials: 100 Configuración uno a uno o mediante trial plantilla

Tabla 2: parámetros editables desde la pantalla de configuración de serie

4.2.4 Configurar trial

Los parámetros a asignar en la pantalla de configuración de trial se pueden clasificar en parámetros relacionados con el entorno, parámetros relacionados con el tiempo, parámetros relacionados con la interacción sujeto - entorno, y parámetros relacionados con el tipo de captura de información realizada durante la ejecución.

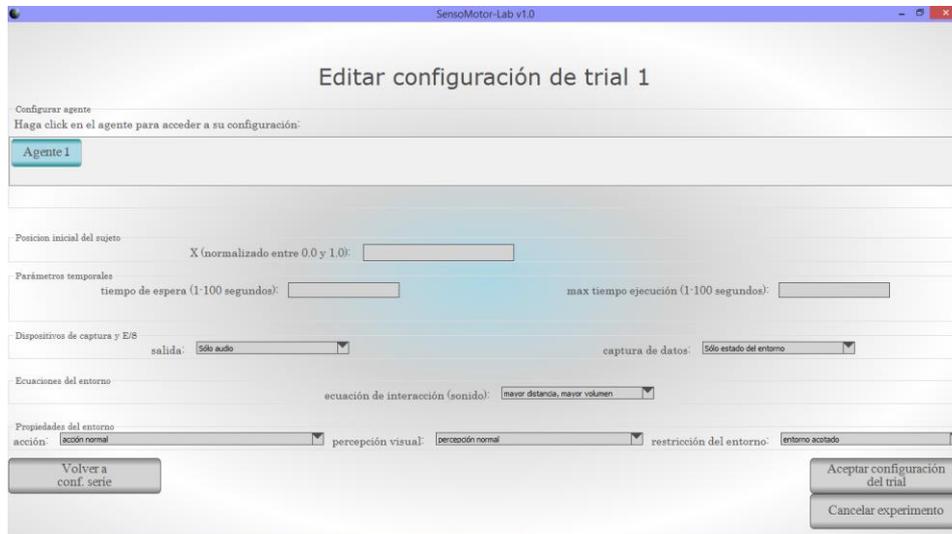


Figura 37: captura de la pantalla de configuración de trial

4.2.4.1 Parámetros relacionados con el entorno

Para definir el entorno, hay que dar valor al estado inicial, que se define por la coordenada del sujeto y la coordenada del agente. Entre los cuatro agentes planteados en la plataforma, hay dos que siguen un comportamiento independiente del sujeto, y dos cuyo comportamiento es dependiente de la coordenada del sujeto. Para simplificar el número de parámetros de configuración, se ha decidido que los agentes con comportamiento dependiente del sujeto partan de una coordenada inicial calculada internamente a partir de la coordenada inicial introducida para el sujeto.

Desde la pantalla de configuración de trial no se puede saber el comportamiento que va a tener el agente hasta que el investigador acceda a su pantalla de configuración y lo indique, y según sea el comportamiento elegido, la coordenada inicial del agente es o no es un parámetro a configurar. Por tanto, la pantalla de configuración de trial sólo permite asignar valor inicial a la coordenada del sujeto.

Además del estado inicial, Se habilita un selector que permite indicar el grado de restricción de movimiento que plantea el entorno.

4.2.4.2 Parámetros temporales

El primer parámetro temporal del trial es el tiempo de relax que se desea esperar antes de la ejecución. El segundo parámetro es el tiempo máximo de ejecución permitido para el trial. En tareas con agentes en los que se decida que la ejecución termine cuando se alcance un estado determinado del entorno, puede que dicho estado no se alcance en mucho tiempo. Para evitar que la ejecución de un trial pueda alargarse indefinidamente, se introduce un tiempo tras el cual la ejecución terminará tanto si se ha resuelto la tarea como si no.

4.2.4.3 Parámetros de interacción

Hay que seleccionar si el sujeto va a percibir el estado del entorno por el canal visual, por el canal auditivo, o por ambos.

Se habilita un selector que permite elegir el tipo de percepción visual, y otro que permite seleccionar el tipo de percepción auditiva.

En la parte de la interacción correspondiente a la acción, no hay que seleccionar el periférico de entrada porque siempre es el ratón (o el touchpad), pero sí hay que decidir qué tipo de ley física del entorno, si normal o invertida, se pretende aplicar.

4.2.4.4 Parámetros de captura

Por último, dado que tal vez no todos los investigadores que quieran hacer uso de esta plataforma tengan acceso al lector de señal cerebral con el que se conecta la aplicación, se ha decidido permitir en la configuración la opción de que la ejecución del trial genere sólo el fichero de evolución del estado del entorno, y por tanto no sea necesario que el sujeto se coloque el casco con los sensores para poder ejecutar el experimento. Así, se habilita un selector que permite elegir si se desean capturar los dos tipos de fuentes de información (opción 'Trazas sujeto-agente y EEG', o sólo se desea capturar la información de la evolución del estado del entorno (opción 'Sólo trazas sujeto-agente').

Parámetro	Método de configuración	Comentarios
Coord. Inicial sujeto	Introducir vía texto y aceptar	Rango válido: [0.0,1.0]
Restricción de movimiento del entorno	Seleccionar opción en lista desplegable	{acotado, toroidal, continuo}
Tiempo de relax	Introducir vía texto	Rango válido: [1,100]
Tiempo máx. de ejecución	Introducir vía texto	Rango válido: [1,100]
Canal de percepción	Seleccionar opción en lista desplegable	{audio, vídeo, audio/vídeo}
Tipo de percepción visual	Seleccionar opción en lista desplegable	{normal, invertida}
Tipo de percepción auditiva	Seleccionar opción en lista desplegable	{lineal normal, lineal invertida, gaussiana normal, gaussiana invertida}
Tipo de acción	Seleccionar opción en lista desplegable	{normal, invertida}
Tipo de captura de datos	Seleccionar opción en lista desplegable	{EEG y estado entorno, sólo estado entorno}

Tabla 3: parámetros editables desde la pantalla de configuración de trial

4.2.5 Configurar agente

Como ya se ha comentado en la pantalla de configuración de trial, los parámetros de configuración del agente varían según sea su comportamiento, hasta el punto en el que para dos tipos de agentes el comportamiento es el único parámetro configurable, gestionando la plataforma internamente la asignación de los demás parámetros.

Se habilita un selector para elegir uno entre los 4 comportamientos posibles: estático, sinusoidal, péndulo invertido y mano loca. Una vez se ha seleccionado el tipo de agente, se muestra el resto de parámetros que ese tipo de agente en particular necesita.

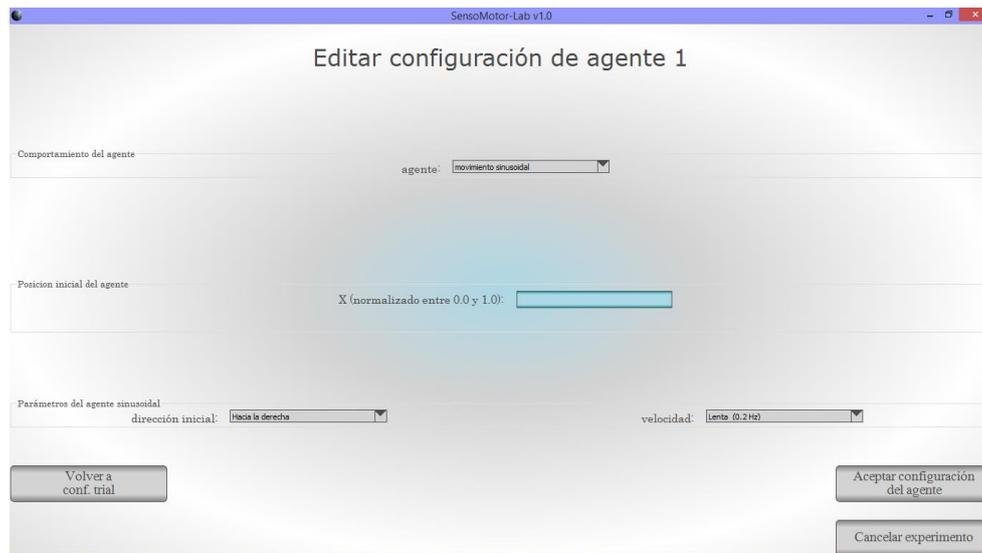


Figura 38: captura de la pantalla de configuración de agente

Parámetro	Método de configuración	Comentarios
Comportamiento	Seleccionar opción en lista desplegable	{estático, sinusoidal, péndulo invertido, mano loca}
Coord. Inicial agente	Introducir vía texto	Rango válido: {0.0, 1.0} Sólo agentes estático y sinusoidal
Dirección inicial	Seleccionar opción en lista desplegable	{hacia la derecha, hacia la izquierda} Sólo agente sinusoidal
Velocidad inicial	Seleccionar opción en lista desplegable	{lenta (0.2 Hz), media (0.4 Hz), rápida (0.6 Hz)} Sólo agente sinusoidal

Tabla 4: parámetros editables desde la pantalla de configuración de agente

4.2.6 Visualización y edición de la configuración

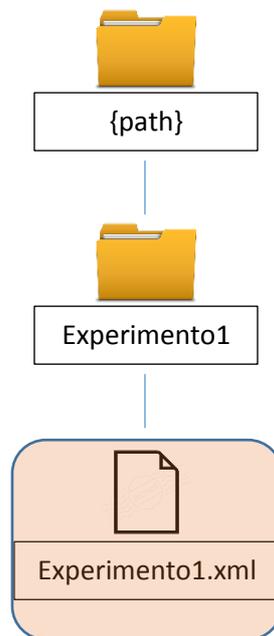
La visualización de la configuración de un experimento existente muestra las mismas pantallas creadas para la configuración de un nuevo experimento, con la diferencia de que en la visualización se deshabilita la modificación de los distintos parámetros mostrados, y el botón 'aceptar configuración' de cada pantalla se sustituye por 'Editar configuración', el cual, al ser pulsado, vuelve a habilitar la modificación de parámetros en todas las pantallas.

4.2.7 Formato del fichero externo de configuración

Al elegir el formato del fichero externo de configuración de un experimento se han considerado 3 requisitos: que sea un formato de uso extendido para evitar problemas de compatibilidad con otras aplicaciones, que sea un formato que permita generar estructuras anidadas que representen adecuadamente la jerarquía de elementos de configuración de un experimento, y que sea un formato con funciones ya desarrolladas dentro del framework QT (empleado para la implementación) que faciliten su escritura y lectura en fichero. Con estos requisitos, el formato elegido ha sido XML. El anexo III muestra la estructura del fichero de configuración de un experimento en formato XML, relacionando cada etiqueta con su parámetro de configuración correspondiente.

4.2.8 Estructura de directorios para almacenar los ficheros de configuración

Al aceptar la configuración de un experimento, se crea una carpeta con el nombre indicado dentro del path indicado. Dentro de la carpeta con el nombre del experimento, se guarda el fichero XML con su configuración.



Configuración del experimento

Figura 39: estructura de directorios para almacenar el fichero de configuración de un experimento

4.3 Ejecución

La ejecución de cada experimento completo va asociada al sujeto que la ha realizado, ya que los ficheros generados en la ejecución se almacenan dentro de una carpeta con el nombre dado al sujeto. Este nombre es solicitado en el momento de la ejecución. Así, cuando se accede desde el menú principal a la funcionalidad de ejecución del experimento activo, aparece una pantalla en la que se debe introducir un nombre para el sujeto.

4.3.1 Espera previa a ejecutar serie

Al describir la estructura temporal de un experimento, se ha mencionado que hay un tiempo indefinido de espera al comienzo de cada serie. Esto se traduce en una pantalla con un botón que espera ser pulsado por el sujeto para comenzar a ejecutar la serie. Para introducir el nombre del sujeto antes de empezar la primera serie, se aprovecha esta pantalla añadiendo un campo de texto en el que introducirlo. De la segunda serie en adelante, se reutiliza esta pantalla simplemente deshabilitando la modificación del campo de texto con el nombre del sujeto.



Figura 40: captura de la pantalla de parada mostrada antes de ejecutar cada serie

4.3.2 Espera previa a ejecutar trial

La pantalla que se muestra durante el tiempo de relax, tiene un fondo de color claro, encima del cual aparece la palabra RELAX en color verde, y en la zona inferior hay una cuenta atrás que va mostrando los segundos que quedan para comenzar la ejecución del trial.



Figura 41: captura de la pantalla de relax mostrada antes de ejecutar cada trial

Antes de pasar a explicar el funcionamiento de la ejecución de un trial, hay que aclarar que los ficheros de captura no están capturando información durante los mismos períodos temporales. En los experimentos que analizan la dinámica cerebral se captura también el período de relax previo a cada trial. Por tanto, la captura de señal cerebral se realiza durante toda la extensión de la serie, generando un fichero de captura por serie, a diferencia de las capturas de la evolución del estado del entorno, que generan un fichero de captura por trial.

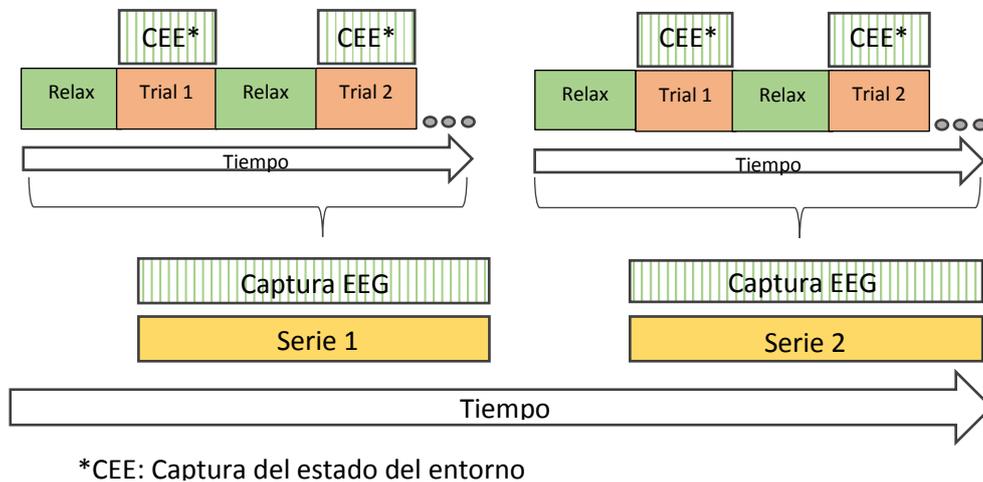


Figura 42: diferencia de períodos temporales de captura de datos obtenidos desde cada fuente

La ejecución de un trial consiste en generar una animación por canal visual y/o auditivo que parta del estado inicial configurado, y a partir de ahí comenzar un bucle de espera-captura-representación hasta que el trial termine por haber excedido el tiempo máximo o por haber llegado a un estado de terminación.

La espera mencionada en el bucle se implementa con un timer configurado a 25 ms de cuenta atrás, que cuando llega a 0 llama a la función de actualización del entorno. Hay que considerar que la ejecución simultánea de otros programas en el sistema hace que el tiempo entre llamadas sucesivas sea realmente mayor a 25 ms.

Durante el tiempo que pasa entre representaciones sucesivas, tanto el agente como el sujeto humano van actualizando su estado. El sujeto humano se desplaza con el ratón libremente en ese tiempo, pero el agente no: el agente simula que se está desplazando durante el tiempo de espera mediante la llamada a su función de comportamiento al final de la función de actualización, calculando la posición que tendrá en el estado siguiente.

Por último, antes de mostrar el algoritmo de ejecución de un trial, aclarar que, debido al problema comentado de que el tiempo de espera entre sucesivas llamadas a la función de actualización no coincide exactamente con el parámetro asignado, el fichero de captura de la evolución del estado del entorno, además de capturar las coordenadas del sujeto y el agente, captura el timestamp (tiempo en ms transcurrido desde el 1 de enero de 1970), para, de ese modo, al restar los timestamps de dos estados sucesivos del fichero, obtener el valor real del tiempo en ms transcurrido entre ambos estados. Así pues, la información guardada en el fichero de captura de la evolución del estado del entorno en cada actualización será el estado del entorno precedido del timestamp del instante en que se realizó la captura.

4.3.3 Algoritmo de la función de actualización del entorno

- 1º llamada:

- Guardar timestamp (instante inicial de la ejecución)
- Inicializar variables que permiten representar y modificar el estado.
- Inicializar estructura de captura del estado
- Inicializar el timer a 25 ms y empezar la cuenta atrás
- Representar el estado inicial
- Llamar a la función de comportamiento del agente para calcular el valor que tendrá ageX en la siguiente llamada a la función de actualización.
- Para entorno acotado, si el movimiento del agente se sale de los límites, asignar a ageX el límite más cercano a su valor actual.

- Llamadas posteriores:

- Guardar timestamp (instante actual de la ejecución).
- Comparar timestamp actual con timestamp inicial para comprobar si se ha alcanzado el tiempo máximo de ejecución.
- Obtener posición del cursor (en pixels)
- Mapear posición en pixels del cursor a coordenadas del entorno
 - Si entorno acotado -> Mapeo en entorno acotado
 - Si entorno toroidal -> Mapeo en entorno toroidal
 - Si entorno libre -> Mapeo en entorno libre
- Aplicar física invertida en su caso
- Añadir estado actual: {Timestamp actual, sujX, ageX} a la estructura de captura del estado del entorno
- Si se ha alcanzado el tiempo máximo de ejecución o el estado actual indica que debe terminar la ejecución:
 - Parar el timer que llama a la función cada 25 ms
 - Generar fichero de captura del estado a partir de la estructura de captura del estado.
 - Si fin de experimento: parar captura datos EEG y volver a menú principal
 - Sino, si fin de serie: parar captura datos EEG, preparar siguiente serie para ejecutar y mostrar pantalla de espera.
 - Sino, Si fin de trial: preparar siguiente trial a ejecutar y mostrar pantalla de relax con el tiempo de espera configurado.
- Si no se ha terminado la ejecución
 - Aplicar mapeos de percepción visual y/o auditiva invertida (o gaussiana en el segundo caso).
 - Representar el estado actual por los canales correspondientes.
 - Llamar a la función de comportamiento del agente para calcular el valor que tendrá ageX en el estado capturado durante la siguiente llamada a la función de actualización.
 - Para entorno acotado, si el movimiento del agente se sale de los límites, asignar a ageX el límite más cercano a su valor actual.

4.4 Reproducción de experimentos

A partir de la información del fichero de captura de la evolución del entorno se puede volver a representar la secuencia de estados capturados para un trial determinado de una serie determinada en un experimento determinado.

Para aprovechar el concepto de experimento activo como ‘el experimento con el que se está trabajando actualmente’, y no tener que crear otro diálogo para seleccionar el experimento que contiene el trial que se desea reproducir, sólo se permiten reproducir trials correspondientes al experimento activo. Por tanto, si el trial que se desea reproducir no pertenece al experimento activo actualmente, debe emplearse la funcionalidad de carga de experimento externo para localizar y cargar el fichero de configuración del experimento que contenga el trial buscado.

Una vez cargado el experimento correcto, al acceder a la funcionalidad de reproducción se cambia a una pantalla que solicita introducir un nombre de sujeto, un índice de serie y otro de trial. Si la serie con el primer índice introducido existe en el experimento activo y el trial con el segundo índice introducido existe dentro de la serie anterior, y además existe una carpeta con el nombre del sujeto introducido, entonces se busca el fichero de captura del estado del entorno para el trial y la serie, que sólo puede ser uno, sólo puede tener un nombre y sólo puede estar ubicado en una carpeta, y si está se carga su contenido y se reproduce. Si no hay ninguna carpeta con ese nombre del sujeto, o si la serie o el trial se salen del rango configurado para el experimento activo, se muestra un mensaje de error en el campo correspondiente. Si se desea volver al menú principal desde esa pantalla, se ha habilitado un botón para hacerlo.

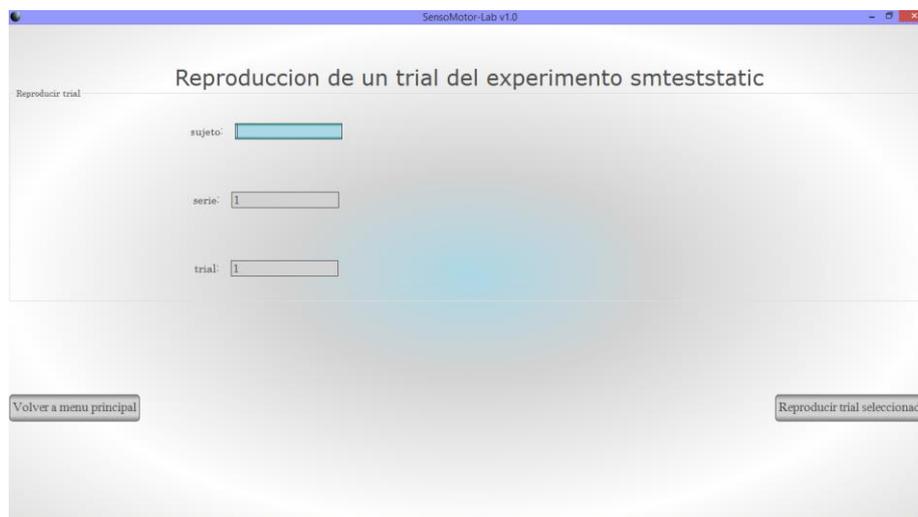


Figura 43: captura de la pantalla que permite seleccionar el trial a reproducir

La reproducción se realiza mediante un algoritmo similar en estructura al de ejecución, pero mucho más sencillo al no tener en cuenta mapeos de pixel a coordenada (aunque sí hay que realizar mapeo de coordenada a pixel), ni capturas de datos. A continuación se muestra el pseudo-código del algoritmo.

4.4.1 Algoritmo de la función de reproducción del entorno

- 1º llamada:

- Cargar los datos del fichero externo en una estructura interna. Como se calcula el tiempo en ms entre un estado y otro a partir de la línea actual y la siguiente, se escribe 2 veces la última línea del fichero externo en la estructura interna para no salirse de rango al representar el último estado.
- Inicializar variables que permiten representar el estado por los canales visual y auditivo.
- Inicializar a 0 una variable que indica el nº de estados representados durante la reproducción.
- Inicializar el timer a 0 ms y empezar la cuenta atrás (realizar la 2º llamada directamente)

- Llamadas posteriores:

- Comparar el nº de estados representados con el nº de líneas del fichero de captura para ver si se debe terminar la reproducción.
- Si se debe terminar la reproducción
 - Parar timer
 - Mostrar menú principal
- Si no debe terminar la reproducción
 - Calcular tiempo entre esta llamada a la función y la llamada a la función siguiente a partir del timestamp del estado actual y el timestamp del estado siguiente
 - Inicializar el timer con ese valor y empezar la cuenta atrás
 - Mapear las coordenadas del estado actual a píxeles en pantalla
 - Representar el estado por pantalla y por los altavoces
 - Sumar 1 a la variable que indica el nº de estados representados

5 Registro de datos

5.1 EEG

Los datos capturados por el lector de señal cerebral son secuencias temporales del estado de intensidad eléctrica medido en distintas zonas de la periferia del cerebro del sujeto. Como estos datos se capturan de forma continua durante la ejecución de cada serie, es necesario indicar, en un fichero externo distinto al que contiene las intensidades, en qué instante de tiempo comienza la ejecución de un trial y en qué instante de tiempo comienza el período de relax previo a la ejecución de un trial. El fichero donde se indica esta información temporal se denomina fichero de marcadores.

La plataforma hace uso del lector de señal cerebral a través de las funciones de una librería implementada por el grupo de investigación ISAAC, y que facilita su uso casi transparente. Esta librería emplea varios métodos para comunicarse con el lector de señal cerebral:

- Conexión del ordenador con el lector a través de sockets, escuchando en un puerto determinado.
- Comienzo de la lectura continua de los datos recibidos a través de ese socket y almacenamiento de los mismos separándolos en varios ficheros de captura. Esta lectura/almacenamiento se ejecuta de forma transparente a la aplicación, desde un thread distinto al principal.
- Inserción de marcador 'Comienza tiempo de relax' en el fichero de marcadores.
- Inserción de marcador 'Comienza ejecución de trial' en el fichero de marcadores.
- Detención del bucle lectura/almacenamiento para cuando termine la ejecución de la serie completa correspondiente.

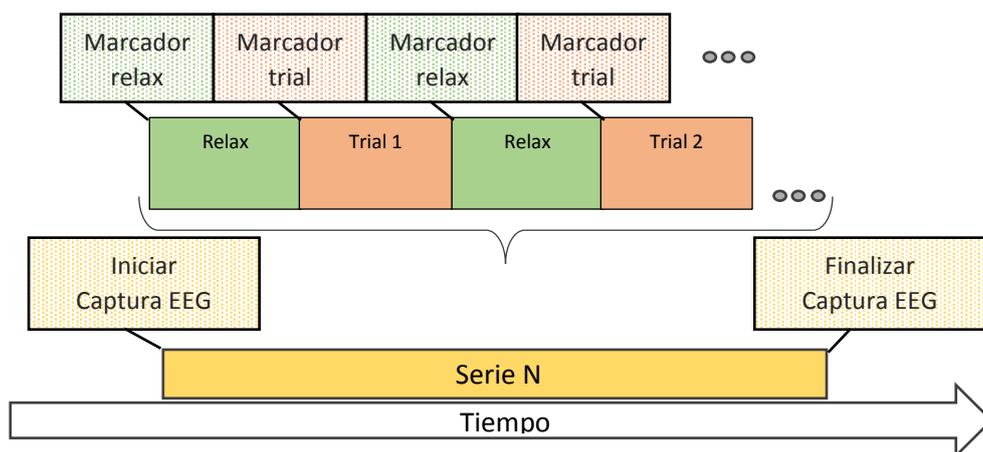


Figura 44: protocolo de comunicación entre la plataforma y la librería que gestiona la captura EEG

5.2 Evolución del estado del entorno

El estado del entorno capturado se almacena externamente en un fichero CSV, que permite escribir los datos de manera sencilla y leerlos de manera estructurada con muchos programas comerciales. Así, generando un fichero con este contenido:

```
sep=,
windowWidth,1366
windowHeight,768
decimal digits,6
"TIMESTAMP","SUBJECT_X","SUBJECT_Y","AGENT1_X","AGENT1_Y"
001472055296513,0000500000,0000500000,0000603457,0000052421,
001472055296544,0000500000,0000500000,0000606831,0000056342,
001472055296575,0000500000,0000500000,0000610122,0000060332,
001472055296606,0000500000,0000500000,0000613331,0000064384,
001472055296638,0000500000,0000500000,0000616460,0000068492,
```

Figura 45: formato del fichero donde se guarda la captura de la evolución del estado del entorno

Se puede abrir con otros programas, obteniendo una visión estructurada similar a esta:

windowWidth	1366			
windowHeight	768			
decimal digits	6			
TIMESTAMP	SUBJECT_X	SUBJECT_Y	AGENT1_X	AGENT1_Y
1472055296513	500000	500000	603457	52421
1472055296544	500000	500000	606831	56342
1472055296575	500000	500000	610122	60332
1472055296606	500000	500000	613331	64384
1472055296638	500000	500000	616460	68492

Figura 46: posible visualización estructurada del fichero de captura del estado del entorno desde una aplicación externa

- La 1ª línea, "sep=," , indica al programa que considere el carácter 'coma' como separador de elementos.
- La segunda línea y la tercera indican el tamaño en pixels (ancho y alto) de la pantalla sobre la que se ha realizado la representación por el canal visual del estado del entorno.
- La cuarta línea indica la potencia de 10 por la que hay que dividir los valores de coordenada de sujeto y agente que aparecen en el fichero para obtener las coordenadas reales, ya que antes de pasar esos datos de la estructura interna al fichero se multiplican por esa potencia y se guarda en el fichero sólo la parte entera del resultado. Eso hace que todos los valores de coordenadas ocupen el mismo tamaño en el fichero CSV, a pesar de ser valores reales.
- La quinta línea informa de los elementos que corresponden a cada columna. Facilita visualizar las columnas desde aplicaciones que lo vayan a cargar fuera de la plataforma.
- Desde la sexta línea hasta la última línea del fichero están contenidos los distintos estados capturados del entorno, precedidos del instante de tiempo en que se realizó la captura.

5.3 Estructura de directorios para el almacenamiento de los ficheros de captura

Aquí se muestra el ejemplo de un experimento con 3 series, tal que la 3ª serie tiene 3 trials. Un sujeto llamado Ana ha ejecutado este experimento, y en dicha ejecución se ha generado una carpeta con su nombre, y dentro de ella una carpeta por cada serie del experimento. Dentro de la carpeta de cada serie se guarda un fichero distinto por cada trial que tenga configurado, y los ficheros generados por la captura de señal cerebral realizada durante toda la serie.

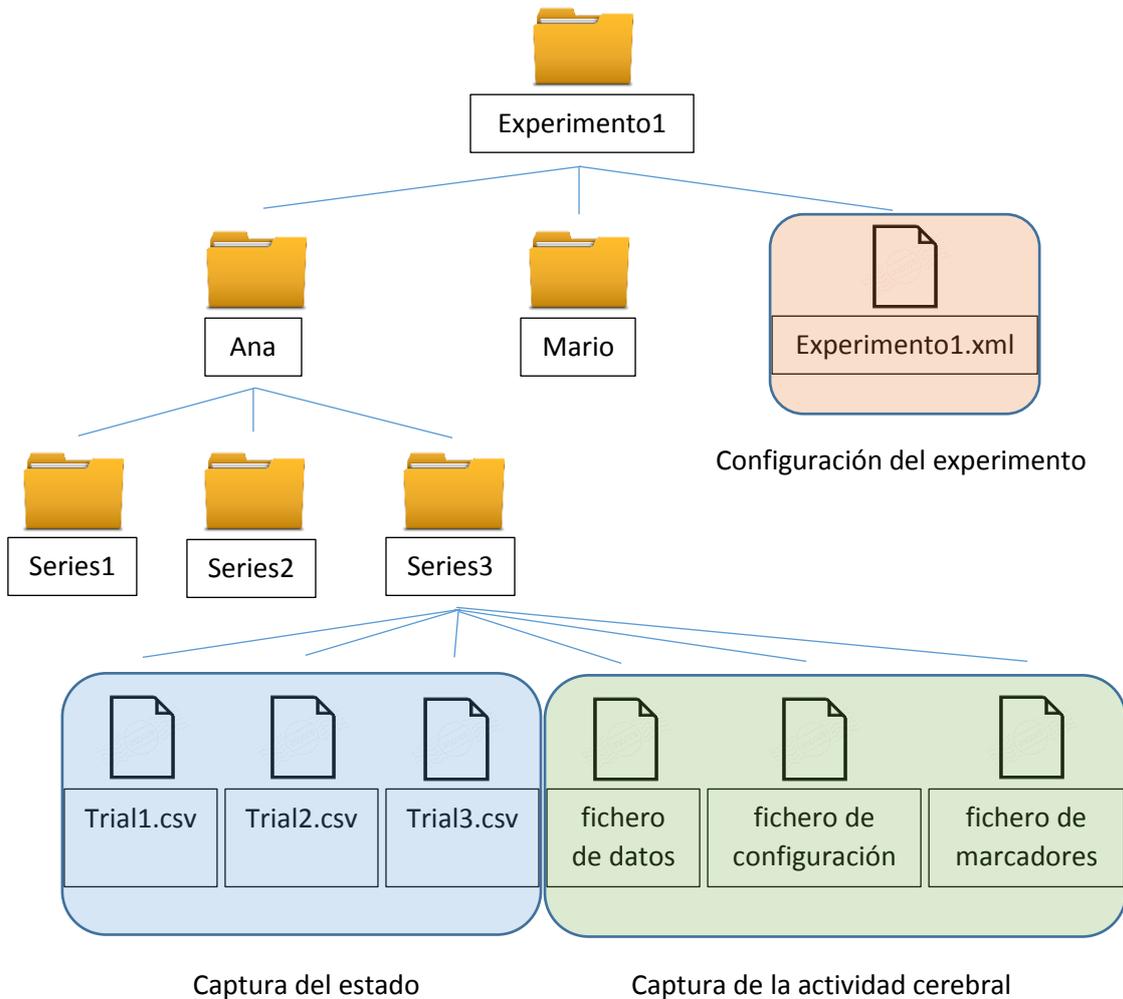


Figura 47: estructura de directorios para almacenar los ficheros de captura y el fichero de configuración de un experimento

6 Diseño de experimentos con la plataforma

Para cada tipo de agente se ha diseñado un determinado experimento, sin perjuicio de que más allá del período en que se redactó esta memoria se puedan plantear otros diseños con estos agentes, o se puedan implementar otros agentes que den lugar a otros experimentos diseñados. Para los 4 experimentos diseñados se ha considerado sólo percepción auditiva no invertida. Además, en todos los experimentos diseñados se ha considerado que 10 trials es suficiente para capturar todo el proceso de aprendizaje ante una única tarea y configuración de entorno, sobre todo si el tiempo máximo de ejecución de cada trial es grande, pero a criterio del investigador se puede modificar este número de trials por serie si es insuficiente o excesivo:

- **Agente estático:** se plantea analizar las diferencias de adaptación del sujeto a los 3 entornos configurados (entorno acotado, entorno toroidal, entorno continuo) durante la tarea de acercamiento a este agente. Para ello se diseña un experimento con 3 series de 10 trials cada una, de forma que cada serie esté configurada con un entorno distinto, y en cada trial la posición inicial de sujeto y agente estático sean diferentes.
- **Agente sinusoidal:** se plantea estudiar la dinámica de aprendizaje del sujeto en la tarea de acoplar su desplazamiento al movimiento oscilatorio del agente sinusoidal a distintas velocidades, tanto cuando el entorno es acotado como cuando es continuo. Para ello se diseña un experimento con 6 series de 10 trials cada una. Cada pareja de series tiene configurada una velocidad distinta para el agente sinusoidal, y en cada pareja la primera serie se desarrolla en entorno acotado y la segunda en entorno continuo.
- **Agente péndulo invertido:** se plantea comprobar cuánto tiempo se tarda en dominar la tarea de mantener el péndulo sin que la parte superior (el agente) 'caiga al suelo' (el ángulo del péndulo respecto a la vertical sobrepase un umbral dado). Esta tarea puede conllevar mayor dificultad al aplicar la física invertida en la acción del sujeto. Se plantea el diseño de un experimento de 4 series de 10 trials cada una. En todas las series se configura la tarea de mantener el péndulo vertical un tiempo máximo fijo y se configura el entorno toroidal, pero en las series 1 y 3 la acción será normal y en las series 2 y 4 la acción será invertida.
- **Agente mano loca:** se plantea averiguar qué capacidad adaptativa tiene el sujeto para entender las características únicas del movimiento que realiza este agente. Para ello se diseña un experimento de una serie con 10 trials en entorno continuo.

Para ejecutar un experimento en la plataforma hay que seguir los siguientes pasos:

- El investigador configura el experimento con los parámetros que desee.
- Se solicita al comité ético de la universidad que apruebe la ejecución de tal experimento.
- Se solicita al sujeto que quiera ejecutar el experimento la firma de un consentimiento informado de que sus datos de señal cerebral serán utilizados para estudios científicos, y si lo acepta pueden ser insertados en bases de datos de uso internacional para ser utilizados en otros estudios por otros investigadores.
- En caso de aceptar las condiciones, se le sienta frente al ordenador, se le coloca el casco de EEG con sensores en las áreas de la periferia cerebral cuya actividad se desee analizar, y se conecta el cable USB de transmisión de datos EEG al ordenador para que sea accesible desde la plataforma.
- El investigador ofrece al sujeto las explicaciones oportunas sobre el experimento. Según el objetivo de cada experimento puede ser interesante que el sujeto reciba poca información en cuanto al tipo de entorno y el tipo de interacción con el agente. Esto potenciará su necesidad de aprender a desenvolverse por el entorno mediante una exploración activa (desplazamiento), en lugar de asociar cada estado y cada cambio de estado percibido en el entorno a un modelo mental formado a partir de las explicaciones que haya podido recibir.
- Una vez investigador y sujeto estén de acuerdo en empezar, el sujeto ejecutará el experimento configurado y al final de éste se generarán los ficheros de captura correspondientes.



Figura 48: sujeto humano ejecutando un experimento de la plataforma mientras su señal cerebral es capturada

7 Conclusiones y trabajo futuro

El anexo IV muestra el calendario del proyecto y la distribución de horas trabajadas, separadas por funcionalidades y por fases de desarrollo.

De los objetivos planteados para esta plataforma, todos se han cumplido satisfactoriamente:

Desarrollo de una plataforma de experimentación con el framework Qt/C++.	✓
Implementación de un panel de configuración que permita a los investigadores modificar y diseñar diferentes experimentos mediante un interfaz gráfico.	✓
Integración en la plataforma de un sistema de adquisición de señal EEG (equipo ActiCHamp).	✓
Implementación de un mínimo de 3 agentes software con distinto comportamiento.	✓
Registro de tres tipos de señal durante cada experimento: movimiento del sujeto, su señal EEG y movimiento de los diferentes agentes software.	✓

Tabla 5: objetivos cumplidos en el desarrollo de la plataforma

Durante la realización de este proyecto se ha aprendido de forma práctica la complejidad que entraña el diseño de una interfaz gráfica, no tanto a nivel conceptual, sino a nivel del número de atributos a configurar para cada elemento individual que se muestra por pantalla, para obtener un resultado a la vez funcional y visualmente atractivo. Por ello, a la hora de planificar el tiempo en trabajos futuros, se analizará detalladamente la complejidad que engloba la interfaz gráfica dentro del proyecto antes de dar una estimación de tiempos con la que trabajar como base.

Además, para la interacción con algunos agentes ha sido necesario generar un motor físico simplificado (sin gestión de colisiones) que simulara un determinado desplazamiento. Este trabajo ha servido al proyectando para profundizar y comprender con mayor facilidad determinadas magnitudes físicas y su rol en ecuaciones matemáticas. Por ello, se apuesta por impulsar el uso de animaciones por ordenador como herramienta para la docencia de materias que contengan ecuaciones dinámicas, en lugar de imágenes o diagramas estáticos, para facilitar la comprensión de las ecuaciones a los alumnos de dichas materias.

Trabajo futuro

Se plantea la posibilidad de realizar un cambio de distribución de elementos en la pantalla de configuración de trial para dividir los parámetros entre aquellos relacionados con el entorno, con el tipo de percepción, con el tipo de acción y con el tipo de captura, de un modo estructurado.

En otro orden de cosas, esta plataforma nace con la pretensión de expandirse poco a poco desde la base generada en este proyecto, según se vaya planteando la necesidad de nuevos agentes o nuevos entornos para simular nuevas interacciones sensomotoras que analizar. Una de las expansiones que puede tener interés más inmediato es la de añadir un módulo de análisis de los datos capturados para no tener que emplear otra herramienta de análisis como MatLab o similares.

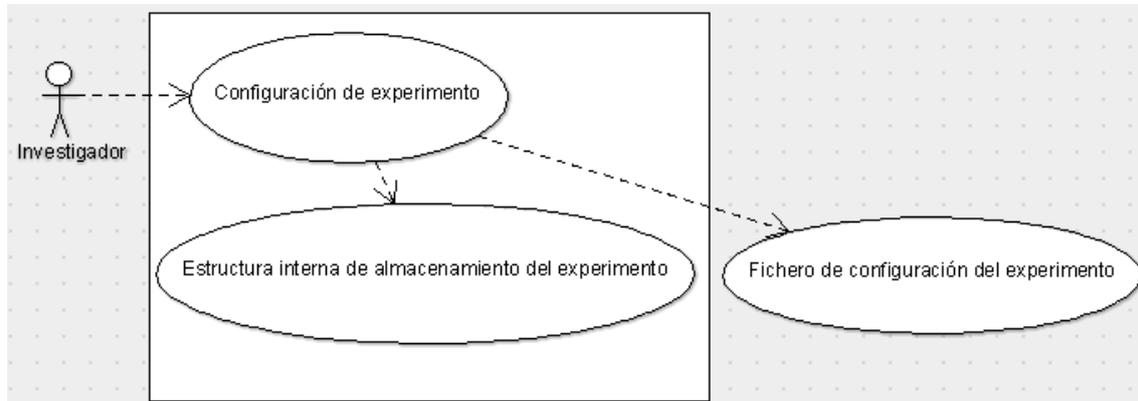
Bibliografía

- Harvey, I., Di Paolo, E. A., Tuci, E., Wood, R., Quinn, M., (2005). Evolutionary robotics: A new scientific tool for studying cognition. *Artificial Life*, 11(1/2), pp. 79 – 98.
- Piaget, J. (2012). La equilibración de las estructuras cognitivas: problema central del desarrollo (primera edición 1978). Siglo XXI editores, ISBN 978-84-323-1625-8.
- Von Uexküll, J. (2010). A Foray Into the Worlds of Animals and Humans: With a Theory of Meaning. University of Minnesota Press, 2010, ISBN-10: 0816659001
- O'Regan, J. K., & Nöe, A. (2001). A Sensorimotor Theory of Perceptual Experience. *Synthese*, 129, 79-103.
- Braitenberg, V. (1986). *Vehicles: Experiments in Synthetic Psychology*. Bradford Book; First Edition edition (February 7, 1986) ISBN-10: 0262521121
- Beer, R. D. (2003). The dynamics of active categorical perception in an evolved model agent. *Adaptive Behavior* 11(4):209-243.
- Van Duijn, M., Keijzer, F., & Franken, D. (2006). Principles of Minimal Cognition: Casting Cognition as Sensorimotor Coordination. *Adaptive Behavior* 2006; 14; 157
- Rohde, M. (2010) *Enaction, Embodiment, Evolutionary Robotics. Simulation Models in the Study of Human Cognition*. Atlantis Press, Amsterdam, Series: Thinking Machines.

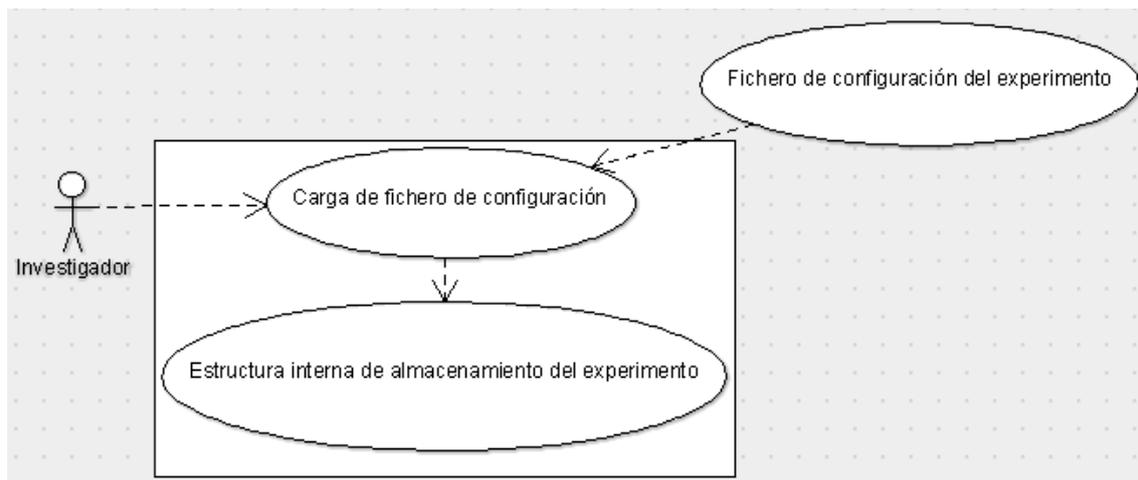
Anexos

Anexo I: Casos de uso

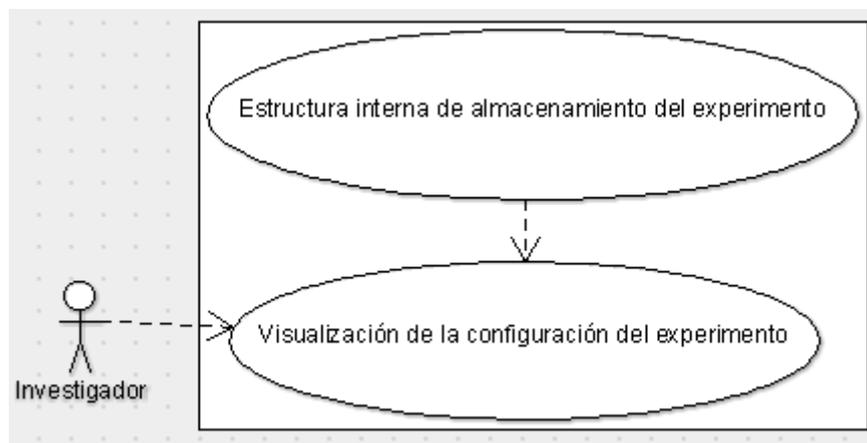
Caso de uso 1: configurar experimento



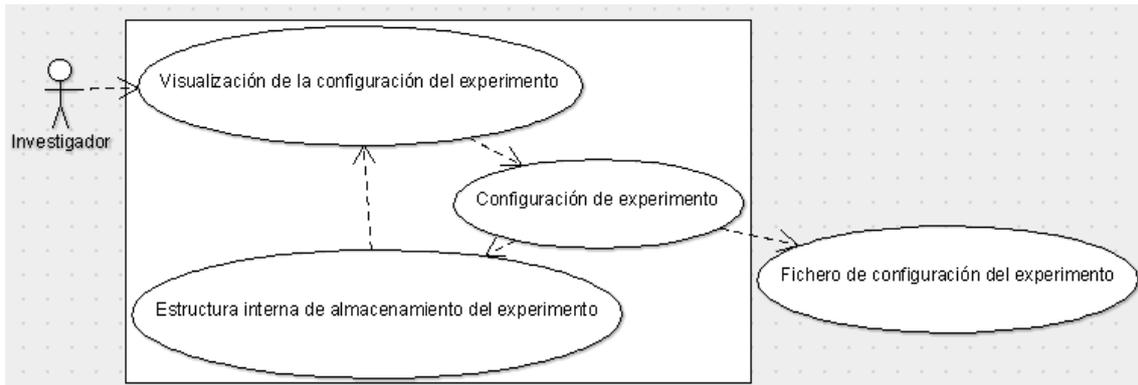
Caso de uso 2: cargar experimento desde fichero



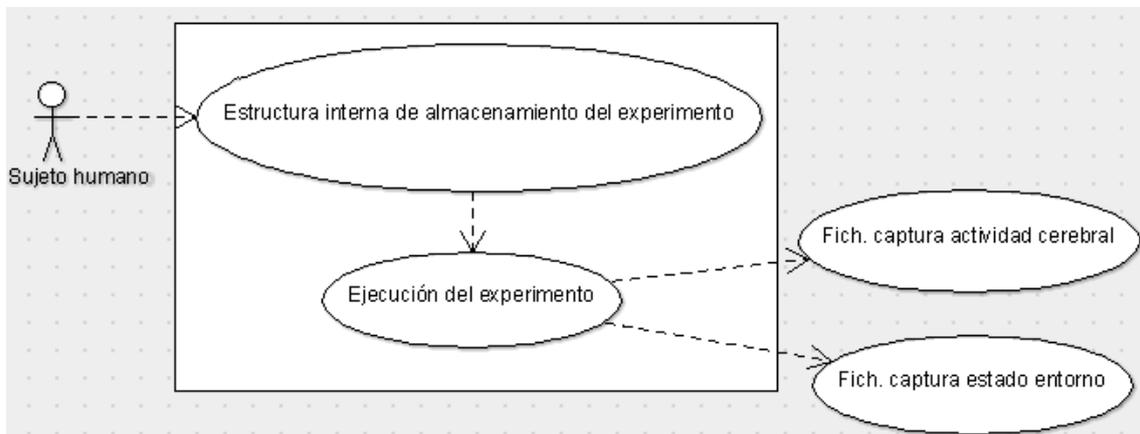
Caso de uso 3: visualizar configuración de experimento activo



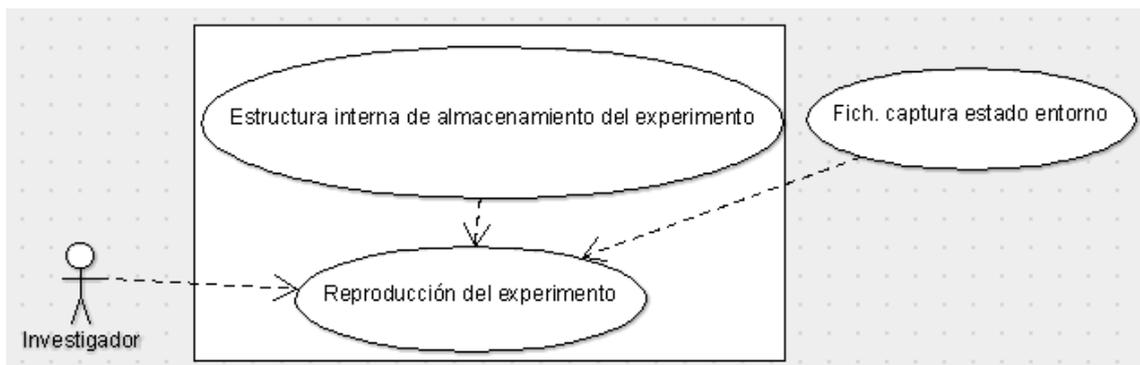
Caso de uso 4: edición, a través de la visualización, de la configuración del experimento activo



Caso de uso 5: ejecución del experimento activo



Caso de uso 6: reproducción de un trial del experimento activo



Anexo II: Generación de GUI (pantallas y navegación)

Proceso general seguido en el código de la aplicación para conseguir generar la GUI (pantallas y navegación):

- El punto de entrada al programa (main.cpp) llama al constructor de la clase MainWindow. Después muestra en pantalla el contenido de MainWindow y comienza la ejecución real de la aplicación, permitiendo la generación y escucha de eventos en las distintas clases que los empleen. Un evento es una señal asíncrona generada por el cambio de estado de un elemento gráfico (widget), el cual puede ser asignado a un método de clase, de manera que dicho método se ejecute cada vez que dicho evento se genere en el widget asignado.

- El constructor de MainWindow:

- llama al constructor de la clase Experimento, que crea un experimento vacío.
- Cada pantalla mostrada en la aplicación se gestiona desde una clase diferente, y MainWindow llama al constructor de cada clase pantalla por separado. Este constructor crea los widgets de la pantalla, los distribuye espacialmente siguiendo un layout determinado e inicializa cada widget con valores por defecto.
- Tras llamar a los constructores de clases pantalla, asigna a cada pantalla los punteros a las clases accesibles desde ella misma. Por ejemplo, desde la pantalla menú principal se puede acceder a 4 pantallas distintas, por lo que MainWindow asignará a la clase pantalla del menú principal 4 punteros a cada una de las 4 clases pantalla accesibles.
- Después, MainWindow crea un QStackedWidget, que es una pila que permite apilar varios widgets permitiendo que sólo se muestre visible el que se encuentra en la cima.
- Seguidamente, MainWindow añade cada clase pantalla a la pila de widgets.
- Finalmente, conecta los eventos de la clase pantalla del menú principal, y la coloca en la cima de la pila de widgets.

Con esto termina la ejecución del constructor de MainWindow. A partir de aquí, el conjunto de clases pantalla funciona como una máquina de estados donde cada estado es una clase pantalla y cada transición o cambio de pantalla es resultado de la generación de un evento de botón pulsado. En cada transición se aplica el siguiente algoritmo:

- Desconectar eventos de la clase pantalla actual
- Modificar clase pantalla siguiente con los valores adecuados
- Conectar eventos de clase pantalla siguiente
- Situar clase pantalla siguiente en la cima de la pila



Anexo III: fichero XML de configuración de experimento

Estructura fichero XML

tag	tipo	parámetro
<numseries>	entero	nº series del experimento
<series>	estructura	-
<series> -> <id>	entero	identificador de una serie
<numtrials>	entero	nº trials de una serie
<trial>	estructura	-
<trial> -> <id>	entero	identificador de un trial
<subjectpos>	estructura	-
<subjectpos> -> <x>	float	coordenada del sujeto en el entorno
<subjectpos> -> <y>	float	no utilizada en esta primera versión
<waittime>	entero	tiempo de relax previo al trial
<maxexectime>	entero	tiempo máximo de ejecución del trial
<interacteqinx>	entero	tipo de percepción auditiva
<outputtypeinx>	entero	canal o canales de percepción
<capturetypeinx>	entero	tipo de captura
<actionlnx>	entero	tipo de acción
<visualperceptioninx>	entero	tipo de percepción visual
<environRestrains>	entero	tipo de entorno
<numAgents>	entero	nº agentes del trial (siempre 1)
<agent>	estructura	-
<agent> -> <id>	entero	identificador de un agente (siempre 1)
<behavioureqinx>	entero	tipo de comportamiento del agente
<agentpos>	estructura	-
<agentpos> -> <x>	float	coordenada del agente en el entorno
<toright>	boolean	dirección inicial del agente sinusoidal
<freqinx>	float	velocidad del agente sinusoidal

Anexo IV: Diagrama de Gantt y tablas de tiempos

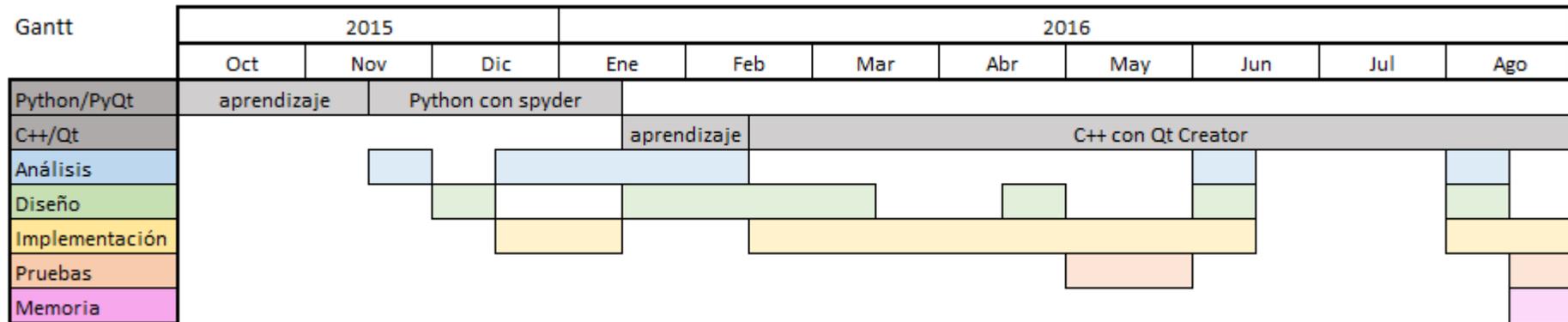


tabla de tiempos

Por fases	Configuración	Ejecución	Otras funcionalidades	Nº horas por fase
Análisis	75	30	15	120
Diseño	100	70	35	205
Implementación	200	130	50	380
Pruebas	30	30	10	70
Nº horas por funcionalidad	405	260	110	775
			Memoria	90
			Total	865

